Towards Generalized Machine Learning Models for Dislocation Image Analysis: A Parametric Based Synthetic Data Approach

Von der Fakultät für Georessourcen und Materialtechnik der
Rheinisch-Westfälischen Technischen Hochschule Aachen

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigte Dissertation

vorgelegt von

Kishan Govind, M. Sc.

Berichter: Herr Univ.-Prof. Dr. Stefan Sandfeld
Herr Univ.-Prof. Dr. rer. nat. Joachim Mayer

Tag der mündlichen Prüfung: 07.05.2025

Acknowledgements

Embarking on the journey of a PhD is a similar experience as setting sail into the vast unknown—a path filled with challenges, yet brimming with opportunities for growth and discovery. Though it is an individual endeavor, it is by no means a solitary one. The completion of this thesis is a testament to the collective support, guidance, and encouragement I've been fortunate to receive from many remarkable individuals, to whom I owe my deepest gratitude.

First and foremost, I would like to express my heartfelt appreciation to my supervisor, Professor Dr. Stefan Sandfeld. His expertise and vision have been invaluable in shaping the direction of my research. Sandfeld's unwavering support, belief in my abilities, and willingness to push me further than I thought possible fostered an environment where learning and innovation flourished. His guidance was not just about my work, but about my growth as a researcher and thinker. His mentorship has been central to every step of my academic journey, always encouraging me to reach for new heights and produce high-quality research.

I am equally grateful to Professor Marc Legros and his group, whose collaboration significantly enriched my research. His generosity in sharing his profound knowledge of transmission electron microscopy (TEM) image data of dislocations provided a crucial foundation for my work. His insights opened new doors for exploration and offered a broader perspective on the field, for which I am deeply thankful.

A PhD journey is marked not only by academic challenges but also by moments of personal triumph and trials. In these moments, the support of family and friends has been indispensable. To my friends—your encouragement, humor, and companionship have been my source of strength during the highs and the lows. I am so grateful for the times you've lifted my spirits and reminded me to take a break when needed.

I would also like to extend my sincere thanks to the European Research Council (ERC) for providing essential funding through the MuDiLingo project, Grant agreement ID: 759419, "A Multiscale Dislocation Language for Data-Driven Materials Science." The financial support from the ERC played a crucial role in advancing this research, enabling me to delve deeper into the complexities of my work and explore new methodologies. This funding not only provided the resources necessary to pursue my research but also fostered collaboration and innovation, which were integral to the success of this project.

To my parents, your love and sacrifices have laid the foundation for everything I've achieved. Your unwavering belief in me, even from afar, has been a constant source of motivation. I cannot thank you enough for the strength you've given me to pursue this path. To my wife, Pooja, words fall short in expressing how grateful I am for your support. Your patience, love, and understanding have been my steady anchor throughout this journey. You've stood by me through moments of

doubt and frustration, always ready to offer encouragement and a smile, even when the days were long and the nights felt endless. Your faith in me has made all the difference.

Finally, I would like to offer my deepest thanks to God for the strength, guidance, and blessings that have seen me through this journey. I recognize that this achievement is not mine alone but is shared with all those who have supported and believed in me. To all of you—family, friends, mentors, and loved ones—this milestone would not have been possible without you. Thank you for being part of this journey with me.

Abstract

Since the first observation of dislocations in the mid 1950s, when electron microscopy was used to visualize these defects, there have been significant advancements in microscopy techniques, allowing for the acquisition of high-quality, high-resolution dislocation image data. Today, it is even possible to perform in-situ mechanical testing, enabling the observation of dislocation microstructure evolution during the plastic deformation of materials. The dislocation image data generated in such experiments need to be studied quantitatively to facilitate meaningful calculations and to understand the underlying mechanisms. Deep learning methods, particularly image segmentation based on convolutional neural networks like U-Net, offer a powerful tool for segmenting dislocation lines which can provide us a way to represent the dislocations as splines to perform quantitative studies. However, these methods require substantial amounts of labeled training data, requiring us to perform many more experiments and labor-intensive manual labeling of dislocation lines. Lack of high quality, large quantity training data presents a significant challenge to applying state-of-the-art deep learning models to dislocation image data. This work addresses that challenge.

In this work, we introduce a novel parametric-based synthetic data generation model, which enables the creation of synthetic training datasets for deep learning-based training of Transmission Electron Microscopy (TEM) images of dislocation microstructures. The synthetic data generation model proposed in this work is designed to generate training data in a way that not only replicates the background of TEM images but also renders complex dislocation microstructures—an essential aspect of materials science research.

Two distinct methods are used for generating synthetic image backgrounds. The first method leverages Perlin noise, combined with random white noise, to create a purely synthetic background, offering a controlled environment for dislocation rendering. The second method, which is much more realistic, uses patches of backgrounds from real TEM images, reassembling them to form realistic-looking backgrounds. This approach mirrors the complexity and variability present in real TEM images, providing a more accurate context for the synthetic dislocation structures.

The core innovation of this work lies in the modeling of dislocation microstructures for synthetic training data. We start with dislocation line and model it as a spline by providing support points for the spline. By representing dislocations as splines, the model achieves high fidelity in simulating dislocation patterns, such as dislocation pileups. These support points can be obtained through two methods: polynomial approximation of dislocation lines or manual selection of key points using image annotation tools like Labelme[1] on dislocations in real TEM images. This flexibility allows for the creation of diverse range of dislocation microstructures consisting of a wide range of configurations, such as dislocation pileups, with varying slip widths, directions, and dislocation counts. Additionally, two more structures—slip trace lines and grain boundaries—are incorporated into the microstructure which are modelled as a line, further aiding

machine learning models in learning the characteristics of dislocations and improving predictive accuracy. The ability to generate complex dislocation structures, some of which are challenging or even impossible to observe in actual TEM images, is particularly significant.

After generating the synthetic training data, the next step involves training machine learning models. In this work, we explore three different machine learning approaches. The first two approaches, multi-label segmentation and instance segmentation, predict individual dislocations as binary masks, which need to be post-processed to represent dislocations as splines and obtain digital representation of the image. Third approach is a more direct approach which estimates the spline support points on the dislocations to represent the dislocation splines directly. We conduct extensive studies to demonstrate the use of the synthetic data and show how it can be used as an alternate to real experimental data or along with real data.

This research represents an important step toward developing generalized machine learning models for dislocation analysis by leveraging synthetic data. The development of a novel parametric-based synthetic data generation model addresses the need of obtaining high-quality training data for machine learning models, particularly for TEM image analysis. The synthetic data generation model enables the creation of synthetic images that closely resemble real TEM images while capturing complex dislocation structures. By generating diverse and realistic training datasets, this research opens up new possibilities for applying advanced deep learning methods, such as U-Net and Mask R-CNN, to the segmentation and analysis of dislocations enabling high throughout studies. Furthermore, the study demonstrates the effectiveness of using machine learning models trained on synthetic data to perform quantitative analysis on real experimental data, reinforcing the practical applicability of these methods in material science research and offers valuable insights into the mechanisms of plastic deformation, further contributing to our understanding of material behavior.

Zusammenfassung

Seit der ersten Beobachtung von Versetzungen Mitte der 1950er Jahre, als die Elektronenmikroskopie zur Sichtbarmachung dieser Defekte eingesetzt wurde, gab es erhebliche Fortschritte bei den Mikroskopietechniken, die die Erfassung hochwertiger, hochauflösender Versetzungsbilddaten ermöglichten. Heute ist es sogar möglich, mechanische Prüfungen vor Ort durchzuführen, um die Entwicklung der Mikrostruktur von Versetzungen während der plastischen Verformung von Werkstoffen zu beobachten. Die bei solchen Versuchen erzeugten Versetzungsbilddaten müssen quantitativ untersucht werden, um sinnvolle Berechnungen zu ermöglichen und die zugrunde liegenden Mechanismen zu verstehen. Deep-Learning-Methoden, insbesondere die Bildsegmentierung auf der Grundlage von neuronalen Faltungsnetzen, sind ein leistungsstarkes Werkzeug zur Identifizierung von Versetzungslinien. Diese können dann verwendet werden, um Versetzungen als Splines zu rekonstruieren, was quantitative Studien ermöglicht. Für diese Methoden sind jedoch große Mengen an beschrifteten Trainingsdaten erforderlich, so dass eine große Anzahl von Experimenten und eine arbeitsintensive manuelle Beschriftung der Versetzungslinien durchgeführt werden müssen. Der Mangel an qualitativ hochwertigen Trainingsdaten in großen Mengen stellt eine große Herausforderung für die Anwendung moderner Deep-Learning-Modelle auf Versetzungsbilddaten dar. Diese Arbeit befasst sich mit dieser Herausforderung. In dieser Arbeit stellen wir ein neuartiges parametrisches Modell zur Erzeugung synthetischer Daten vor, das die Erstellung synthetischer Datensätze für die Deep-Learning-basierte Analyse von TEM-Bildern von Versetzungsmikrostrukturen ermöglicht. Das in dieser Arbeit vorgeschlagene Modell zur Erzeugung synthetischer Daten wurde entwickelt, um Trainingsdaten so zu erzeugen, dass nicht nur der Hintergrund von TEM-Bildern nachgebildet wird, sondern auch komplexe Versetzungsmikrostrukturen wiedergegeben werden - ein wesentlicher Aspekt für die Analyse komplexer Mikroskopiebilder in der Materialwissenschaft.

Für die Erzeugung synthetischer Bildhintergründe werden zwei verschiedene Methoden verwendet. Die erste Methode nutzt das so genannte Perlin-Rauschen in Kombination mit weißem Rauschen, um einen rein synthetischen Hintergrund zu erzeugen, der eine kontrollierte Umgebung für die Wiedergabe von Versetzungen bietet. Die zweite Methode, die wesentlich realistischer ist, verwendet segmentierte Bereiche von echten TEM-Hintergrundbildern und setzt sie zu realistisch wirkenden Hintergründen zusammen. Dieser Ansatz spiegelt die Komplexität und Variabilität echter TEM-Bilder wider und bietet einen genaueren Kontext für die synthetischen Versetzungsstrukturen.

Eine weitere Innovation dieser Arbeit liegt in der Modellierung von Versetzungsmikrostrukturen für synthetische Trainingsdaten. Wir beginnen mit einer Versetzungslinie und modellieren sie als kubischen Spline, indem wir Stützpunkte für den Spline bereitstellen. Durch die Darstellung von Versetzungen als glatte, mathematische Kurve erreicht das Modell eine hohe Wiedergabetreue bei der Simulation realistisch aussehender Versetzungsmuster, wie z. B. Versetzungsanhäufungen. Diese Stützpunkte können durch zwei Methoden ermittelt werden:

polynomiale Annäherung von Versetzungslinien und manuelle Auswahl von Schlüsselpunkten mit Hilfe von Bildbeschriftungstools wie Labelme[1] auf Versetzungen in echten TEM-Bildern. Diese Flexibilität ermöglicht die Erstellung einer Vielzahl von Versetzungsmikrostrukturen, die aus einer breiten Palette von Konfigurationen mit unterschiedlichen Positionen, Längen, Ausrichtungen und „Krümmungen“ bestehen. Darüber hinaus werden zwei weitere Strukturen - Schlupfspuren und Korngrenzen - in die Mikrostruktur integriert, die als Linie modelliert werden, was maschinelle Lernmodelle beim Erlernen der Eigenschaften von TEM-Bildern weiter unterstützt und die Vorhersagegenauigkeit verbessert. Die Fähigkeit, komplexe Versetzungsstrukturen zu erzeugen, von denen einige in tatsächlichen TEM-Bildern nur schwer oder gar nicht zu beobachten sind, ist besonders wichtig.

Nach der Generierung der synthetischen Trainingsdaten geht es im nächsten Schritt darum, Modelle für maschinelles Lernen zu trainieren. In dieser Arbeit untersuchen wir drei verschiedene Ansätze des maschinellen Lernens. Die ersten beiden Ansätze, Multi-Label-Segmentierung und Instanzsegmentierung, sagen einzelne Versetzungen als binäre Masken voraus, die nachbearbeitet werden müssen, um die Versetzungen als Splines darzustellen. Der dritte Ansatz ist ein direkterer Ansatz, der die Spline-Stützpunkte auf den Versetzungen schätzt, um die Versetzungssplines direkt darzustellen. Wir führen umfangreiche Studien durch, um die Verwendung der synthetischen Daten zu demonstrieren und zu zeigen, wie sie als Alternative zu echten experimentellen Daten oder zusammen mit echten Daten verwendet werden können.

Diese Forschungsarbeit stellt einen wichtigen Schritt zur Entwicklung verallgemeinerter maschineller Lernmodelle für die Versetzungsanalyse dar, indem synthetische Daten genutzt werden. Die Entwicklung eines neuartigen, parametrisch basierten Modells zur Erzeugung synthetischer Daten geht auf die Notwendigkeit ein, qualitativ hochwertige Trainingsdaten für maschinelle Lernmodelle zu erhalten, insbesondere für die TEM-Bildanalyse. Dieses Modell ermöglicht die Erstellung synthetischer Bilder, die realen TEM-Bildern sehr ähnlich sind und gleichzeitig komplexe Versetzungsstrukturen erfassen. Durch die Erzeugung vielfältiger und realistischer Trainingsdatensätze eröffnet diese Forschung neue Möglichkeiten für die Anwendung fortschrittlicher Deep-Learning-Methoden, wie U-Net und Mask R-CNN, auf die Segmentierung und Analyse von Versetzungen. Darüber hinaus demonstriert die Studie die Effektivität der Verwendung von Modellen des maschinellen Lernens, die auf synthetischen Daten trainiert wurden, um quantitative Analysen an realen experimentellen Daten durchzuführen. Dies unterstreicht die praktische Anwendbarkeit dieser Methoden in der materialwissenschaftlichen Forschung und bietet wertvolle Einblicke in die Mechanismen der plastischen Verformung, was zu unserem Verständnis des Materialverhaltens beiträgt.

Contents

Chapter 1 Introduction

1.1 Background: Dislocations

This section presents a historical overview leading to the experimental discovery of dislocations and their significance in material science. It begins with the discovery of X-ray diffraction and Bragg's Law, foundational developments that enabled the detailed study of atomic structures. The mechanisms of plastic deformation in metals are discussed where we focus on material responses to applied stress and the factors governing deformation behavior. A comparison of theoretical versus experimental estimate of elastic limit of perfect crystals follows, addressing key discrepancies between the two and exploring theories based on imperfections in the crystals proposed to explain these differences. Finally, the concept of dislocations is provided as a fundamental aspect of material plasticity, essential for understanding the underlying mechanisms of plastic deformation in metals.

1.1.1 The Discovery of X-ray Diffraction and Bragg's Law

The early 20th century was a period of remarkable progress in the field of material science. Researchers were beginning to uncover the intricate details of crystal structures and their properties, thanks to advancements in X-ray diffraction and other experimental techniques. These breakthroughs allowed scientists to visualize and model the atomic arrangements within crystalline materials with unprecedented accuracy. Max von Laue discovered X-ray diffraction in crystals in April 1912 at the Institute of Theoretical Physics at the University of Munich [2]. Experiments conducted by Max von Laue provided compelling evidence for the wave nature of X-rays and simultaneously provided images of the atomic structure of crystals [3]. This breakthrough occurred when Laue demonstrated that X-rays, when directed at a crystal, produced a distinct pattern of diffraction spots on a photographic plate. These spots resulted from the interaction between the X-rays and the crystal's orderly arrangement of atoms. Within a year of Laue's groundbreaking discovery, William Lawrence Bragg, along with his father William Henry Bragg, made use of this phenomenon to estimate the atomic-level arrangement within the crystal lattice from the experimental diffraction data [3, 4]. William Lawrence Bragg developed a theoretical framework to interpret the diffraction patterns produced by crystals, which became

known as Bragg's Law. Bragg's Law provides a simple yet powerful equation to describe the relationship between the wavelength of the X-rays, the angle at which they are diffracted, and the distance between the atomic planes in the crystal.

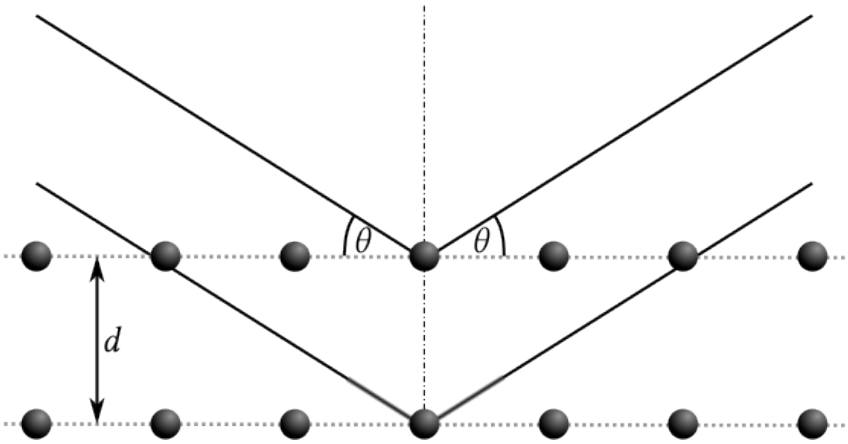


Figure 1.1: Schematic representation of Bragg's law applied to X-ray diffraction, illustrating the constructive interference of X-rays scattered by atomic planes in a crystalline lattice. A beam of X-rays is incident on the crystal lattice with an inter-planar spacing d at an angle of incidence θ. Constructive interference occurs when the condition $2d \sin\theta = n\lambda$ is satisfied, where n is an integer representing the order of diffraction and λ is the wavelength of the X-rays.

Crystals can be thought of as consisting of parallel planes of atoms, separated by a distance d, as illustrated in Figure 1.1. When a beam of X-rays strikes the crystal, each atom within the crystal acts as a scattering center, reflecting the incident X-rays. Due to the periodic arrangement of atoms in the crystal lattice, the scattered X-rays can interfere constructively or destructively depending on the angle of incidence, θ and the wavelength, λ of the X-rays. The path difference between X-rays reflected from adjacent planes must be an integer multiple of the wavelength to produce constructive interference. The path difference is $2d \sin(\theta)$ which accounts for the extra distance traveled by the reflected beam due to the geometric arrangement of the crystal planes. The Bragg's condition can be written as:

$$2d \sin\theta = n\lambda \tag{1.1}$$

where n is an integer representing the order of the diffraction peak (first-order, second-order, etc.). This relationship is fundamental for determining the crystal structure, as it allows material scientists to calculate the inter-planar spacing and infer the arrangement of atoms within the crystal by analyzing the angles and intensities of the diffracted beams.

1.1.2 Plastic Deformation in Metals

With the advent of X-ray diffraction and the insights provided by Bragg's Law, researchers gained the unprecedented ability to visualize crystal structures and determine the atomic arrangements that underpin the macroscopic mechanical properties of materials. One key area where this structural information proved invaluable is in understanding the phenomena of plastic deformation in metals.

Metals, like many other crystalline solids, are arranged in regular arrays of atoms, forming distinctive crystal structures such as face-centered cubic (FCC), body-centered cubic (BCC), and hexagonal close-packed (HCP) lattices. When a metal is subjected to an external stress that exceeds its elastic limit, it undergoes plastic deformation. Unlike elastic deformation, which is fully reversible, plastic deformation involves permanent changes in the material's structure and shape.

Prior to the insights gained from crystallography, the microscopic mechanisms behind plastic deformation remained elusive. Early theories struggled to explain why metals, seemingly rigid and tightly packed with atoms, could deform so readily without fracturing.

After years of dedicated work by brilliant scientists and researchers, the following became well-established facts within the materials science research community regarding plastic deformation in metals.

1. The plastic deformation of metals occurs while preserving their crystalline structure [5]. During deformation, the crystal lattice rearranges to accommodate strain without losing the material's overall crystalline nature, a fundamental aspect of metal plasticity.

2. Early microscopic examination methods, pioneered by researchers like Sorby and others (1904), revealed the granular structure of metals [6, 7]. Etched and polished metal surfaces display irregularly shaped grains. Crystal facets within a single grain reflect light uniformly, while different grains reflect light variably, depending on the viewing angle.

3. Deformation processes such as cold working, rolling, and metal forming elongate grains in the direction of applied strain. Annealing restores the grain structure, erasing deformation marks, and the microstructure before and after annealing appears nearly identical.

4. Experimental evidence shows that metals strained beyond their yield point develop slip bands within each grain, oriented approximately parallel to one another as shown in Figure 1.2. This phenomenon is universal across metals and occurs under various loading conditions, such as tension, shear, bending, and compression. Slip bands form once the elastic limit is surpassed.

5. The appearance of slip bands is strain-dependent. Initially, a single system of bands forms

within a grain, but with increasing strain, multiple slip systems (up to four) may emerge, oriented at different angles.

Figure 1.2: Formation of slip bands in iron. Plastic deformation in metals occurs through a series of internal slips happening at intervals in each grain. Adapted from Ewing and Rosenhain [8]

These observations from early experiments provided a novel perspective on the phenomenon of plastic deformation in metals. These studies offered experimental evidence supporting the occurrence of plastic deformation, often referred to as "plastic flow", which arises from sliding of crystal portions within each grain along well-defined gliding surfaces. Under significant plastic deformation, the metal's crystalline structure remains largely intact, with each grain maintaining its inherent crystallographic alignment, despite macroscopic changes in the shape or orientation of the grain. This implies that the crystalline structure of the metal is preserved even under severe strain, as the deformation predominantly occurs via slip within individual grains.

In particular, it has been observed that heavily strained metals retain their crystalline structure, with the fundamental periodicity and orientation of the crystal lattice preserved at the micro-scale. The structure of these strained metals closely resembles that of the unstrained material, indicating that plastic deformation via slip does not result in a loss of crystallinity. This observation

challenges the earlier notions of a complete breakdown of the crystal lattice under strain and reinforces the concept that plastic deformation in metals occurs through localized rearrangements of the crystal lattice rather than a wholesale destruction of the structure [9–11].

This finding raised a fundamental question in materials science: how can metals undergo significant plastic deformation while preserving their crystal structure? This question has become central to the study of dislocation dynamics, crystal defects, and the mechanisms governing plasticity in crystalline materials.

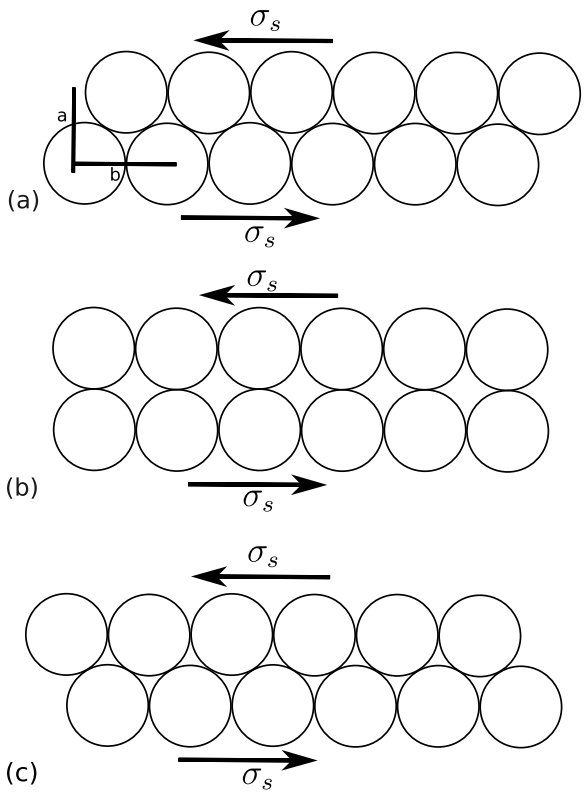


Figure 1.3: Perfect crystal configuration based on Frenkel's model of crystal structure where (a) Initial position of the atoms in a crystal lattice before, (b) during the slip of the crystal place and (c) after the slip of the lattice by one atomic distance. *a* is lattice spacing and *b* is one atomic distance to slip the layer of items to occupy same energy configuration.

1.1.3 Theoretical vs. Experimental Critical Shear Stress

Formation of slip bands is one of the most important feature of crystalline structure of metals but slip bands formation starts only at a critical value. Considering a perfect crystalline structure of metals, Frenkel (1926) [12] attempted to calculate this critical value.

Let us consider a perfect crystal structure arranged in closed packed manner as shown in Figure 1.3a. The structure is in equilibrium and no external stress is required to maintain this arrangement of lattice planes. When the layer of atoms is slipped by one atomic distance denoted as b, the lattice again occupies equilibrium position as shown in Figure 1.3c. We can assume that the shear stress σ_s is a sinusoidal function of displacement, x of the atomic planes with periodicity of $\frac{2\pi}{b}$ and is given by

$$\sigma_s = K \sin(\frac{2\pi x}{b}) \tag{1.2}$$

The constant K can be determined using Hooke's law for small displacements which gives $\sigma_s \approx K\frac{2\pi x}{b} \approx \mu\frac{x}{a}$ where a is lattice spacing. This gives us the value of constant K as $\frac{\mu b}{2\pi a}$. The shear stress can be rewritten as

$$\sigma_s = \frac{\mu b}{2\pi a} \sin(\frac{2\pi x}{b}) \tag{1.3}$$

When the lattice is deformed by a shear stress of $\sigma_s^{critical}$, the two atomic layers of the crystal reach a critical state as shown in Figure 1.3b. The critical stress required to reach this state is $\sigma_s^{critical} = \frac{\mu b}{2\pi a}$. An approximate value of critical stress is obtained as $\sigma_s^{critical} \approx \frac{\mu}{2\pi}$ assuming $a \approx b$.

This simple model proposed by Frenkel approximates the theoretical critical shear stress required for "plastic slip" as $\frac{\mu}{2\pi}$, which when compared against experimentally observed value was found to be several orders of magnitude higher. The discrepancy arises because Frenkel's model assumes that plastic deformation occurs through the simultaneous displacement of two atomic planes by a fixed amount, requiring a high shear stress to overcome the periodic potential barriers between atoms. It motivated the material science community to look for answer to the question of why is the plastic deformation initiates at such low stresses?

1.1.4 Dislocations: The Key to Understanding Material Plasticity

The discrepancy between the theoretically calculated shear stress and the lower stress observed in experiments led to significant advancements in understanding plastic deformation of materials. This "anomaly" suggested that there must be mechanisms at play within the crystalline structure that facilitate easier deformation than predicted by simple atomic displacement models. This puzzle was the driving force behind the development of the concept of dislocations.

The theory of elastic fields due to defects in a continuum body, first introduced by Vito Volterra in 1907 [13] profoundly impacted the understanding of material science, particularly in how imperfections within a crystal lattice. Volterra's work focused on the mathematical descriptions of distortions in elastic bodies, including the concept of dislocations as line defects within crystals. Volterra developed what are now known as "Volterra dislocations", which he described through a series of cut, slip, and weld operations on a material. In his model, a cut is made in the material, the two sides of the cut are displaced relative to each other, and then the material is welded back together, introducing a mismatch, or dislocation, within the lattice. This model was crucial because it provided a geometric and mechanical framework to understand how materials could deform under much lower stresses than those predicted by models assuming perfect lattices.

Although Volterra's contributions initially focused more on the theoretical aspects of elasticity and continuum mechanics, they laid the groundwork for later scientists to explore the specific implications of dislocations in crystals. It was not until the 1930s that the existence of dislocations was postulated independently by Taylor [14], Orowan [15], and Polanyi [16] in order to explain the discrepancy between the theoretical and observed shear stress.

Geoffrey Ingram Taylor was a British physicist and engineer whose work primarily focused on the mechanical properties of metals. Taylor was motivated by the discrepancy between the theoretical and observed strengths of materials and sought to understand the underlying mechanisms of plastic deformation. Taylor proposed that plastic deformation occurs due to the movement of line defects, which he termed "dislocations" within the crystal lattice. He suggested that these dislocations allowed layer of atoms to slip past each other incrementally in a locallised manner, reducing the stress required for plastic deformation compared to the simultaneous movement of entire planes of atoms as shown in Figure 1.4. It starts with a dislocation line within the crystal, marked by a row where there is a discontinuity in the lattice alignment as shown in Figure 1.4b. Under shear stress, the dislocation begins to shift. One part of the lattice above the dislocation line moves relative to the other, creating a distortion in the lattice. As the dislocation nearing the edge of the crystal lattice as shown in Figure 1.4e and is about to exit the lattice on the right side, the dislocation will leave behind a slightly shifted crystal structure. Each movement of the dislocation involves breaking and forming of only a few atomic bonds and requires less energy compared to the movement of the entire plane of atoms simultaneously, which is why dislocations can deform under lower stresses. Taylor introduced the concept of

dislocation density, which refers to the number of dislocations within a given volume of the crystal. He proposed that the density and distribution of dislocations significantly influence the mechanical properties of materials, including their yield strength and work-hardening behavior.

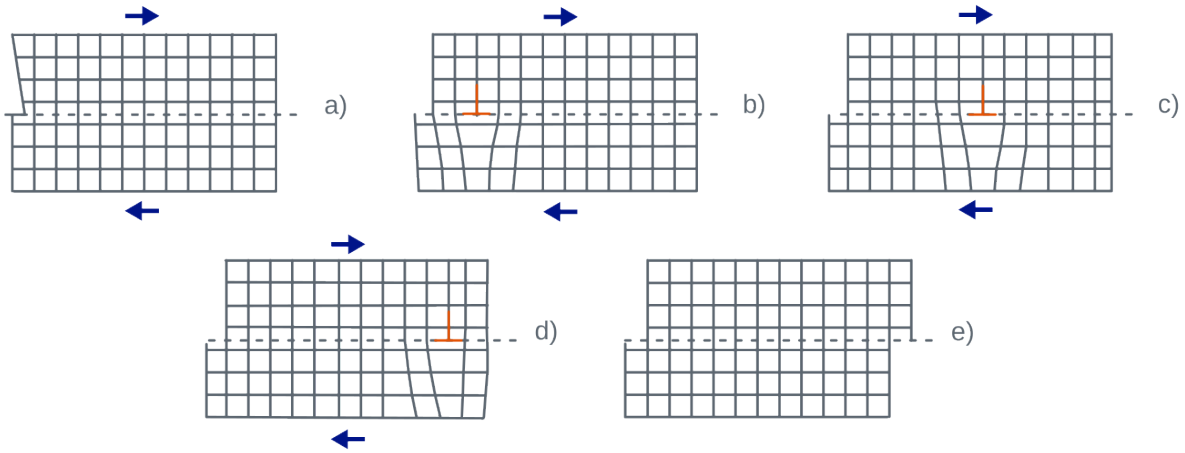


Figure 1.4: Slip in a crystal by motion of dislocation starting the (a) where the dislocation (shown in red) is nucleated at the left and starts to move incrementally breaking atomic bonds and finally exit the crystal in (e).

Egon Orowan focused on the mechanics of dislocation movement within the crystal lattice. He emphasized that the movement of dislocations could explain the low yield strength of crystals observed experimentally. Orowan highlighted the energy aspects of the dislocation motion, demonstrating that the energy required to move a dislocation through the lattice is significantly lower than the energy needed to move an entire plane of atoms. This insight provided a theoretical basis for understanding why plastic deformation occurs at lower stresses.

Michael Polanyi showed how dislocations weaken shear resistance and enable deformation under low stresses. Dislocations typically occur along densely packed planes, moving through simple atomic shifts that allow for plastic deformation. The study also highlighted that dislocations are more likely to form at the crystal surface, where stress is concentrated due to atomic surface roughness. Energy for dislocation formation may come from thermal fluctuations, even at very low temperatures, contributing to the material's persistent plasticity.

1.2 Experimental Evidence of Dislocations

Before the direct experimental verification of dislocations in 1947 and the subsequent visualization of their movement through advanced electron microscopy techniques in the late 1950s, several

lines of theoretical reasoning and indirect evidence strongly suggested the existence of dislocations [17]. A solid theoretical background of dislocation theory was already developed as described in details in Cottrell et al. [18]. There was a need to obtain direct experimental evidence of dislocations and which is where TEM came into use. In this section, we discuss the role of TEM and how the first dislocation was observed experimentally.

1.2.1 The Role of TEM in Overcoming Optical Limitation

Figure 1.5: The first prototype electron microscope, built by Ernst Ruska and Max Knoll in 1931 at the Technical University of Berlin. The instrument employed two magnetic lenses in series to form an "electron-optical" imaging system, demonstrating the foundational principle of electron microscopy. Although its resolution was initially comparable to that of a light microscope, this device paved the way for later designs that exceeded optical limits. Adapted from Williams et al. [19].

It is also important to understand why it was not possible to see the dislocations using optical microscopy methods. Optical microscopy, which uses visible light to magnify objects, was the primary tool for observing the grain microstructure of materials before the advent of electron microscopy. However, optical microscopy has several inherent limitations that make it ineligible for observing dislocations in crystals. The resolution of optical microscopy is fundamentally limited by the wavelength of visible light, which ranges from approximately 400 to 700 nanometers. According to the Abbe diffraction limit [20] we can estimate the resolution,

d for a given wavelength of light using $d \approx \frac{\lambda}{2.8}$. The best possible resolution using visible light is about 200 nanometers. Dislocation can be very long but their core radius is of a few angstroms. Therefore, the size of dislocations are several orders of magnitude smaller than the resolution limit of optical microscopes.

French physicist Louis de Broglie proposed (1924) the idea that electrons are not only material particles but also have wave-like properties and the wavelength of an electron can be calculated using

$$\lambda = \frac{h}{mv} \tag{1.4}$$

where h is Planck constant, m is mass of electron and v is its speed. For an electron, which is accelerated through 200kV voltage would have a wavelength of about 0.025 angstroms which can allow us to probe the structure of the materials down to sub Angstrom level [19].

1.2.2 From Theory to Experiment: The First Direct Observation of Dislocations with TEM

The first electron microscope was developed by Ruska and Knoll in 1932 [19]. With the advancement in science and technology, they where able to resolve to the nano-scale range which allowed experimentalists to get high resolution images of the microstructure using thin samples of materials. The thin films of samples allowed transmission of electrons and reveal important characteristics of microstructure. During those days, one of the main challenge was to prepare very thin samples for the TEM experiment. Bob Thorne, a PhD student of Hirsch prepared a thin sample of aluminum using etching which was a great success. He took pictures of the aluminum sample under the TEM which consisted of short lines in linear arrays at the boundaries between neighboring sub-grains.

At initial guess, it was postulated that the lines could either be Moire pattern fringes[1] due to overlapping crystals or dislocations based on the periodicity of the lines. It was difficult to identify the structure as dislocation accurately. After many efforts, the TEM was configured with double condenser which increased the brightness and then the lines started to move as shown in Figure 1.6. From there, it became clear that the lines observed were indeed dislocations. Hirsh said back then "So the lines started running around, and from there on everything became clear", which opened a whole new area of research and motivated researchers all over the world to use TEM to look deeper into the material. Before the experimental verification of the dislocations, they were postulated without experimental evidence. During the experiments, it was possible

[1] A moiré pattern is an interference pattern that emerges when two grids or patterns with similar but not identical spacing are superimposed. This superposition creates a new pattern characterized by large-scale fringes or bands, which can be visually striking. In microscopy, moiré fringes can occur when two crystalline structures overlap, leading to periodic variations in intensity that may resemble linear arrays.

to quickly verify some of the most important features of the dislocations. They were able to observe slip traces left by dislocations along their paths. The slip trace width and direction along with orientation of the crystal from diffraction allowed to calculate the thickness of the sample. Bowing effects of the dislocations was observed where dislocations were pinned at the surface and bowed out before they could move. These observations further helped researchers to improve the understanding of dislocations and obtain experimental verification of the theory of dislocations.

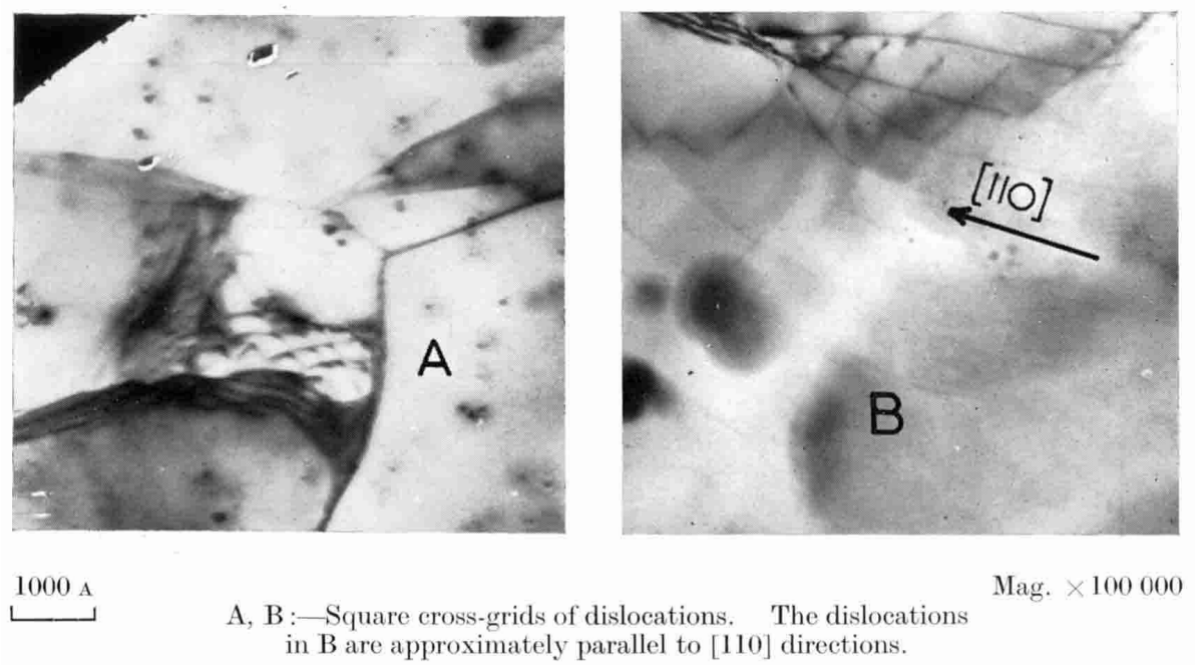


Figure 1.6: First direct experimental observation of dislocations by TEM, as reported by Hirsch, Horne, and Whelan (1956). Shown is a TEM image of aluminium in which dislocation lines—created during annealing and visible via diffraction contrast—are resolved and their motion recorded under suitable imaging conditions. Adapted from Hirsch et al. [21].

1.3 Advancements in TEM: A Deeper Understanding of Dislocations

An electron microscope produces high-resolution images by directing a focused beam of electrons on a specimen. An electron gun generates an electron stream, which is focused into a coherent beam by condenser lenses and narrowed by a condenser aperture. The focused beam interacts

with the sample, and transmitted electrons are captured by the objective lens to form an initial image. The objective aperture enhances contrast by blocking high-angle diffracted electrons, while the selected area aperture provides crystallographic information by analyzing periodic diffraction. Intermediate and projector lenses further magnify the image, which is displayed on a phosphor screen. Thicker, denser regions of the sample appear darker due to reduced electron transmission, while thinner, less dense areas appear brighter.

When a specimen is placed under an electron microscope, it can interact with the material of specimen in two ways: inelastic and elastic scattering. In inelastic scattering, the incident electrons loose energy to the sample through various mechanisms, such as exciting electrons within atoms, causing vibrations in the crystal lattice (phonons), or loosing other forms of energy. This energy loss occurs because the incident electrons interact with the electric field of atoms in the material, transferring part of their energy to the material's electrons or lattice. Inelastic scattering is often considered undesirable for high-resolution imaging because the electrons that have lost energy do not contribute to the image's formation in a manner that can be easily related to the precise atomic structure or defects in the material. This is because the energy loss is not directly tied to specific features of the crystal lattice but rather to broader material properties. Elastic scattering, on the other hand, involves the deflection of incident electrons by the electric fields of atoms in the sample without any loss of energy. The electrons are scattered by the nuclei and electron clouds of the atoms, changing direction but retaining their kinetic energy. The scattered electrons retain their original energy, maintaining the coherence and phase relationships necessary for high-resolution imaging and diffraction analysis. Only elastically scattered electrons are used to form images in TEM. These electrons can be diffracted by the crystal lattice, where defects like dislocations have an influence on the contrast in the resulting images.

1.3.1 Diffraction Contrast in TEM Images for Dislocation Visualization

The TEM contrast condition refers to the specific set of circumstances under which features within a sample, such as atomic planes, defects, or dislocations, become visible or exhibit contrast in a TEM image [22]. Contrast in TEM arises primarily due to the interaction of electrons with the sample, leading to differences in electron intensity across the image. These differences can be attributed to various factors, including mass-thickness contrast, diffraction contrast, and phase contrast. Among these, diffraction contrast is most relevant to imaging dislocations which is based on the diffraction of electrons by the crystal lattice and how this diffraction is affected by defects that distort the lattice. Dislocations introduce local distortions in the crystal lattice, which can alter the path of electrons traveling through the sample.

These distortions affect the conditions under which electrons are diffracted by the crystal planes, leading to variations in image intensity that correspond to the presence and nature of

dislocations. The visibility of dislocations in TEM images under diffraction contrast is governed by several factors, including:

1. Bragg Condition: For electrons to be diffracted by a set of crystal planes, they must satisfy the Bragg condition eq. (1.1). The presence of a dislocation can locally change the orientation of these planes, affecting the diffraction condition. By tilting the specimen to satisfy the Bragg condition for specific planes near the dislocation, the dislocation can be made visible due to the contrast generated by the disturbed diffraction.

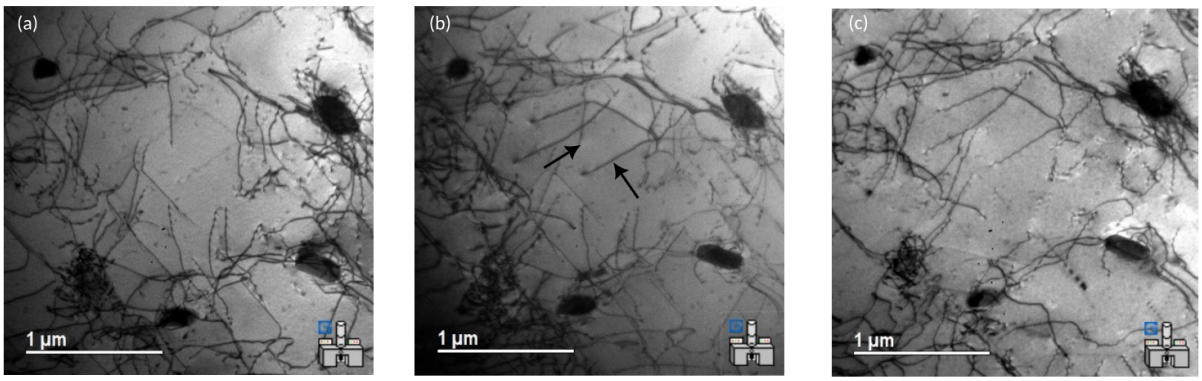


Figure 1.7: Illustration of the two-beam diffraction contrast condition in TEM: by tilting the specimen so that one set of lattice planes satisfies the Bragg (or near-Bragg) condition, specific dislocations become visible. Dislocations indicated by arrows are resolved under this two-beam condition. Adapted from Jenkins and Kirk [23]

2. Visibility Condition: The visibility of dislocations is also related to the so-called $\vec{g}.\vec{b}$ condition, where $\vec{g}$ is the diffraction vector (related to the set of planes being used for imaging) and $\vec{b}$ is the Burgers vector of the dislocation which defines the displacement produced by dislocations. The dot product of $\vec{g}$ and $\vec{b}$ determines whether a dislocation will be visible. For a dislocation to be visible under diffraction contrast, the dot product $\vec{g}.\vec{b}$ should NOT be zero. This condition implies that the diffraction vector must have a component along the Burgers vector for the dislocation to produce contrast. This method is often used to determine the Burgers vector by tilting the specimen while focusing on the same dislocation and observing the diffraction pattern.

3. Two-Beam Condition: The two-beam condition is a simplified scenario where only one set of lattice planes (aside from the direct beam) significantly contributes to the diffraction pattern. This condition is often used for imaging dislocations because it simplifies the interpretation of contrast. By carefully selecting the orientation of the crystal so that it meets the two-beam condition with respect to the electron beam, specific dislocations can be highlighted based on their orientation relative to the crystal lattice as shown in Figure 1.7.

1.3.2 Bright-Field and Dark-Field TEM Imaging

Once the electron beam interacts with the specimen, there are two different ways to obtain the final image under the TEM, commonly known as bright field and dark field as shown in Figure 1.8 which shows the two views of the microstructure of the same region where precipitates are more visible in dark field view. In bright-field TEM imaging, the image is formed by electrons that are transmitted through the sample without being diffracted, or those that are only diffracted once (kinematic scattering), and then collected by the objective lens. Under conditions close to, but not exactly meeting, the Bragg condition, dislocations appear as dark lines on a bright background. The observed dark line width in the TEM image corresponds to the region adjacent to one side of the dislocation where the crystal lattice is slightly tilted or distorted, causing the Bragg condition to be locally satisfied and leading to electron diffraction away from the bright-field detector. This region's width, typically several nanometers, reflects the extent of lattice distortion caused by the dislocation.

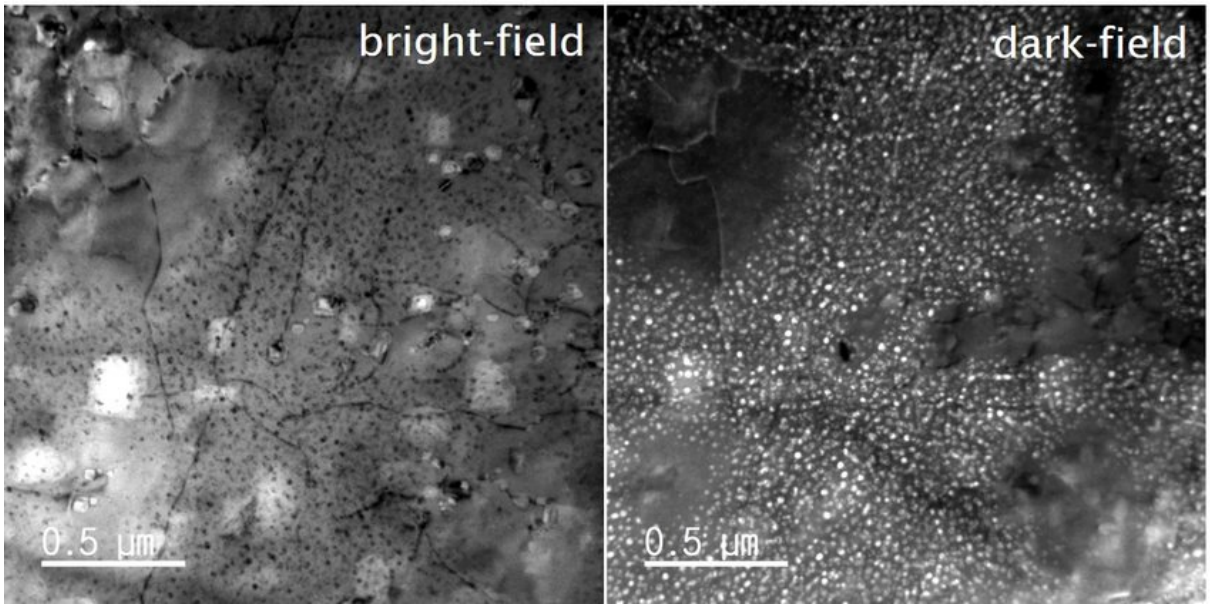


Figure 1.8: Comparison of bright-field and dark-field TEM images of the same specimen region. In the bright-field view, unscattered (or weakly scattered) electrons are used to form the image, so features which scatter electrons strongly appear darker. In the dark-field view, scattered electrons meeting a selected diffraction condition are used instead, enhancing contrast from lattice defects, strain fields, or specific crystallographic features that may be less visible in bright-field mode. Adapted from Park et al. [24].

Weak-beam dark-field imaging mode, as described by Hirsch et al. [25], is a specialized form of dark-field microscopy designed to enhance resolution. It uses a large excitation error, meaning

the Bragg condition is satisfied only very close to the dislocation core. This results in a very thin, bright line (the dislocation) on a dark background. The "weak-beam" condition improves resolution by minimizing contributions from regions away from the dislocation core, focusing on the core's immediate vicinity. However, the weak-beam technique is challenging to use due to its low image contrast, which makes the screen almost entirely dark except for the thin bright lines representing the dislocations. This requires highly stable microscope conditions and longer exposure times to capture the necessary detail, making it less practical for routine use but invaluable for high-resolution studies of dislocation structures.

In this work, the TEM image data of dislocations will be based on bright-field imaging conditions. Bright-field imaging is selected because it effectively balances image contrast and ease of use, making it well-suited for our study of dislocation structures. Unlike weak-beam dark-field imaging, which, although it offers higher resolution, requires highly stable microscope conditions and longer exposure times, bright-field imaging provides sufficient contrast to observe dislocations as dark lines against a bright background without the need for complex adjustments. This makes it more practical for routine analysis while still allowing for the detailed examination of lattice distortions caused by dislocations. This is one of the main reason why bright field imaging is used for quantitative TEM studies of dislocations.

1.4 Quantitative TEM Analysis of Dislocation Microstructures

Quantitative *in-situ* TEM combines dynamic imaging of microstructural changes with simultaneous measurements of stress, strain, and dislocation parameters. By integrating advanced imaging techniques with micro-electro-mechanical systems (MEMS)-based testing platforms, this approach bridges the gap between nanoscale observations and macroscopic mechanical behavior.

One of the key advantages of quantitative TEM is its ability to capture transient and localized dislocation behaviors that are challenging to resolve through postmortem analyses. For instance, Zhou et al. [26] observed the dynamic activation and deactivation of Frank-Read sources during deformation of Cu-brass films, revealing mechanisms that govern dislocation motion under stress. Similarly, Kim et al. [27] demonstrated that long-range internal stress fields from dislocation pile-ups, rather than short-range interactions such as Lomer-Cottrell locks, dominate strain hardening in Fe-Mn-Al-C steels. These insights highlight the value of *in-situ* TEM in understanding dislocation-mediated phenomena. Quantitative TEM data can be integrated with computational techniques to further enhance understanding of localized phenomena. Steinberger et al. [28] manually tracked dislocation positions in TEM images to inform finite element simulations, demonstrating stress distributions within nano-scale regions. This integration enables detailed

analysis of dislocation behavior, such as pinning and de-pinning events, which are critical for understanding material responses to deformation.

Quantitative TEM enables precise characterization of dislocation interactions with microstructural obstacles. Nogiwa et al. [29] quantified obstacle strength in thermally aged Fe–Cu alloys by analyzing dislocation bow-out angles. Advances in MEMS-based TEM systems have further enhanced this capability, as demonstrated by Samaee et al. [30], combined nano-scale stress-strain measurements with direct observations of dislocation behavior. Such studies provide a deeper understanding of dislocation-driven plasticity and its role in material deformation.

Dislocations play a critical role in strain hardening mechanisms. Kim et al. [27] highlighted that long-range stress fields from dislocation pile-ups are key contributors to the superior mechanical properties of Fe-Mn-Al-C steels. Similarly, Legros [31] emphasized the importance of dislocation interactions with microstructural features, such as precipitates and grain boundaries, in determining macroscopic strength. Quantitative TEM facilitates direct measurement of these interactions, providing insights into activation stresses, energies, and strain rates.

High-entropy alloys (HEAs) such as the FeCoCrMnNi Cantor alloy are multi-principal-element alloys containing at least five components. They have emerged as ideal candidates for developing strong materials due to their exceptional strength, ductility, and fracture toughness. The high configurational entropy in Cantor alloys stabilizes a single-phase face-centered cubic (FCC) structure, despite significant lattice distortions. The alloy has atoms with different radii which occupy random lattice positions in the crystal lattice and hence the crystal lattice is strongly distorted. This makes estimation of the behavior of alloy very difficult since this distortion created local elastic fields which may act as pinning points for dislocations. Lee et al. [32] used quantitative TEM to estimate shear stress from dislocation curvature, revealing the role of local dislocation geometry in governing material strength of Cantor alloy. Additionally, Utt et al. [33] identified localized pinning and de-pinning events as the cause of jerky dislocation motion, a phenomenon influenced by the high lattice friction stresses and complex local stress fields inherent to Cantor alloys. Temperature-dependent deformation mechanisms in Cantor alloys have also been investigated. Chen et al. [34] reported a transition from dislocation-mediated plasticity at room temperature to twinning-dominated deformation at cryogenic temperatures, driven by changes in stacking fault energy. These findings emphasize the versatility of quantitative TEM in capturing the effects of external conditions on dislocation behavior.

1.5 Towards High-Throughput TEM Dislocation Analysis

High-throughput TEM dislocation analysis involves tracking dislocations across multiple frames and automatically extracting their shapes to conduct spatio-temporal analyses. Only a few such

studies have been performed since it requires a lot of effort and time to extract dislocations from TEM images. Zhang et al. [35] focused on high-throughput in-situ TEM experiments to analyze the dynamics of dislocations in the Cantor alloy by manually labeling and indexing each dislocation as polygon in 330 frames. A novel data-mining approach, incorporating spatio-temporal coarse-graining, was developed to quantify the evolution of dislocation structures and the strength of local pinning points. The study revealed that pinning points vary in strength and spatial locations, influenced by dislocation passage and interactions, with some points strengthened while others weaken. This approach provides new insights into the friction mechanisms and strengthening behavior in Cantor alloys, demonstrating the potential of dislocations as tools for probing complex energy landscapes.

In the study by Song et al. [36], a groundbreaking approach was developed to enable quantitative analysis of dislocation avalanches using in situ TEM experiments. This research used a high-throughput, deep learning-based method to automate analysis of TEM data, reconstructing the spatio-temporal evolution of dislocation microstructures with precision, making use of thousands of frames from experiment video. The analysis revealed the jerky, stick–slip motion characteristic of dislocations in Cantor alloy. By integrating a digital twin of TEM experiments with discrete dislocation dynamics (DDD) simulations, the study quantified the avalanche-like dynamics of dislocations, observing scale-free distributions in plastic strain increments and stress drops. Unlike conventional FCC metals, the study of avalanche-like behavior was possible in Cantor alloy where the high lattice friction slows down the dynamics of dislocations considerably and dislocations could be tracked. This work established a connection between experimental findings and statistical physics frameworks, such as depinning transitions, highlighting the universality of dislocation avalanches. The novel methodology not only demonstrated the critical role of dislocations in influencing material behavior but also showcased the potential of data-driven approaches to transform TEM microscopy into a truly quantitative and reproducible tool for materials science.

The increasing availability of high-throughput TEM data presents an unprecedented opportunity to explore dislocation dynamics with greater depth and precision. However, there are a number of challenges to realize the full potential of such datasets. These challenges are discussed in the following section starting with reconstruction of three-dimensional dislocation microstructure from two-dimensional TEM images obtained from experiments.

1.5.1 Three-Dimensional Reconstruction of Dislocation Microstructures

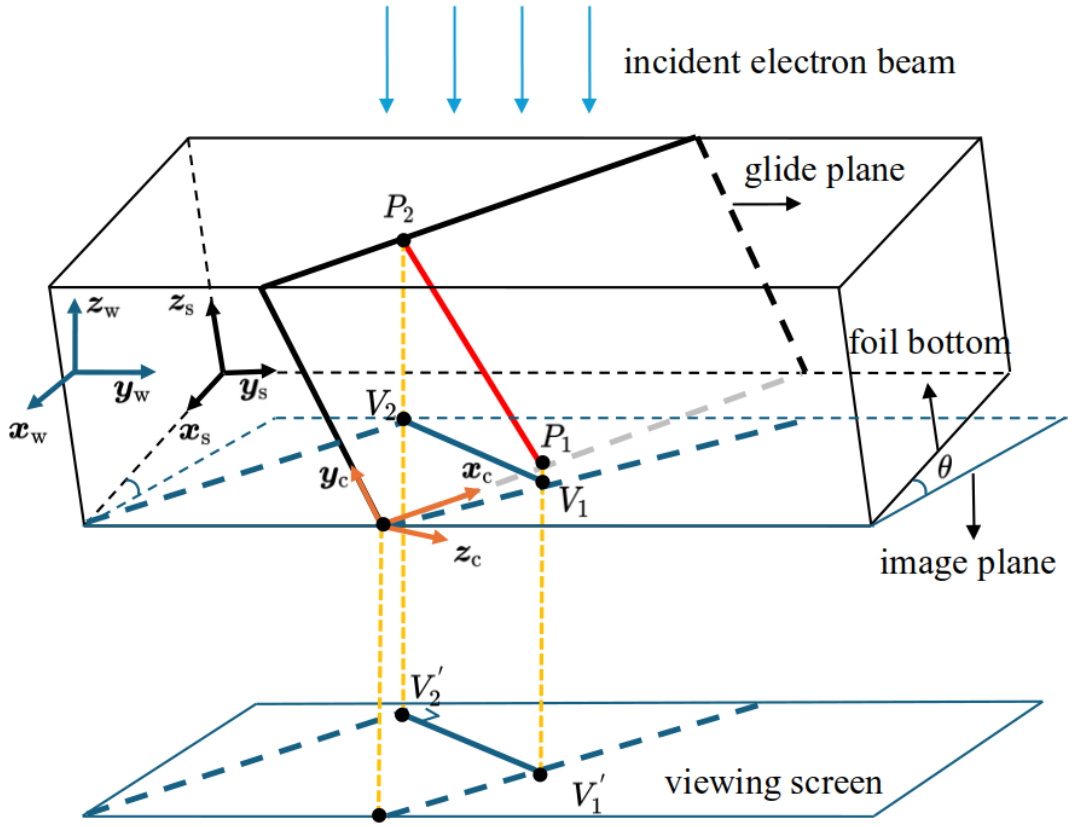


Figure 1.9: Schematic of projection geometry in TEM. A three-dimensional dislocation line, P_1P_2, in the crystal is projected onto the image plane as $V_1^{'}V_2^{'}$. To reconstruct the true 3D configuration, coordinate transformations between the world frame (x_w, y_w, z_w), the sample frame (x_s, y_s, z_s), and the crystal frame (x_c, y_c, z_c) are required. Adapted from Zhang et al. [35].

One of the key bottlenecks in high-throughput TEM dislocation analysis is the accurate interpretation of the inherently three-dimensional (3D) dislocation structures from the two-dimensional (2D) images captured by TEM. TEM images are two-dimensional projections of inherently three-dimensional dislocation structures as shown in Figure 1.9. The electron beam interacts with the dislocation, P_1P_2 of the material and the projected 2D dislocation, $V_1^{'}V_2^{'}$ is viewed in the TEM image. We are working with 3 coordinate systems here, the world coordinate frame x_w, y_w, z_w, sample frame x_s, y_s, z_s and crystal frame x_c, y_c, z_c and would need to perform coordinate transformation between the frames. Accurate 3D reconstruction is crucial for understanding the true spatial configurations and interactions of dislocations. Please refer to Zhang et al. [35]

where authors performed a detailed study to obtain representation of the dislocations from the image plane to the crystal lattice plane.

The 3D reconstruction of TEM images [37, 38] can be achieved through two main approaches namely direct methods and indirect methods. Direct methods rely on deep learning techniques to reconstruct the 3D structure directly from the images. In contrast, indirect methods involve post-processing steps, where the 3D structure is inferred from the analysis and interpretation of 2D images obtained during imaging. For example:

- Sills and Medlin [39] introduced a semi-automated, object-based tomographic approach for dislocation structure analysis, aimed at overcoming limitations of intensity-based reconstruction methods in TEM. Their method involves three key steps. First, dislocation lines are extracted from TEM images through segmentation, which includes noise removal, adaptive thresholding, and skeletonization to reduce lines to a single-pixel width, followed by conversion into an object-based representation of line segments connected by nodes. These nodes are classified as either physical (e.g., intersections or endpoints) or non-physical (arbitrarily defined along the line). Second, a manual matching process aligns and associates dislocation lines across multiple tilted images, leveraging a graph-based representation to identify shared physical nodes. Finally, tomographic reconstruction determines the three-dimensional configuration of the dislocation lines, using a mapping technique based on arc length along the tilt axis. Despite its advantages, the approach remains semi-automated, requiring manual corrections during line extraction and object matching. It also struggles with distinguishing overlapping dislocations, resolving ambiguities in physical node identification, and handling uncertainties introduced by line orientation and extraction errors. One of the critical challenge faced by the authors is accurate segmentation of dislocations which is also the focus of this work.

- Altingövde et al. [40] proposed a deep learning-based pipeline for the automated 3D reconstruction of dislocations using stereo pairs of TEM images, bypassing the traditional, labor-intensive tomography approaches. Their method involves a Siamese network configuration for segmentation and feature extraction using a U-Net, followed by a 3D CNN to estimate disparities for constructing a 3D structure. This end-to-end approach combines detection and matching, eliminating manual intervention while delivering comparable accuracy to traditional methods. However, the performance heavily relies on high-quality and diverse training data, which is challenging to acquire and annotate. The model's reliance on stereo pairs with a limited tilt angle limits its generalization, as higher angles degrade visual similarities, complicating the matching process. Furthermore, while the system automates many steps, the limited dataset diversity hinder its robustness across varying imaging conditions and materials, and the disparity refinement still requires assumptions about the geometry and matching continuity of dislocations.

A critical step in these approaches is the accurate segmentation of dislocations in 2D images

and their representation as lines or splines. Deep learning could be very useful in automating the dislocation representation but such studies need high quality training data.

1.5.2 Variability in Dislocation Imaging due to Material Properties and TEM Conditions

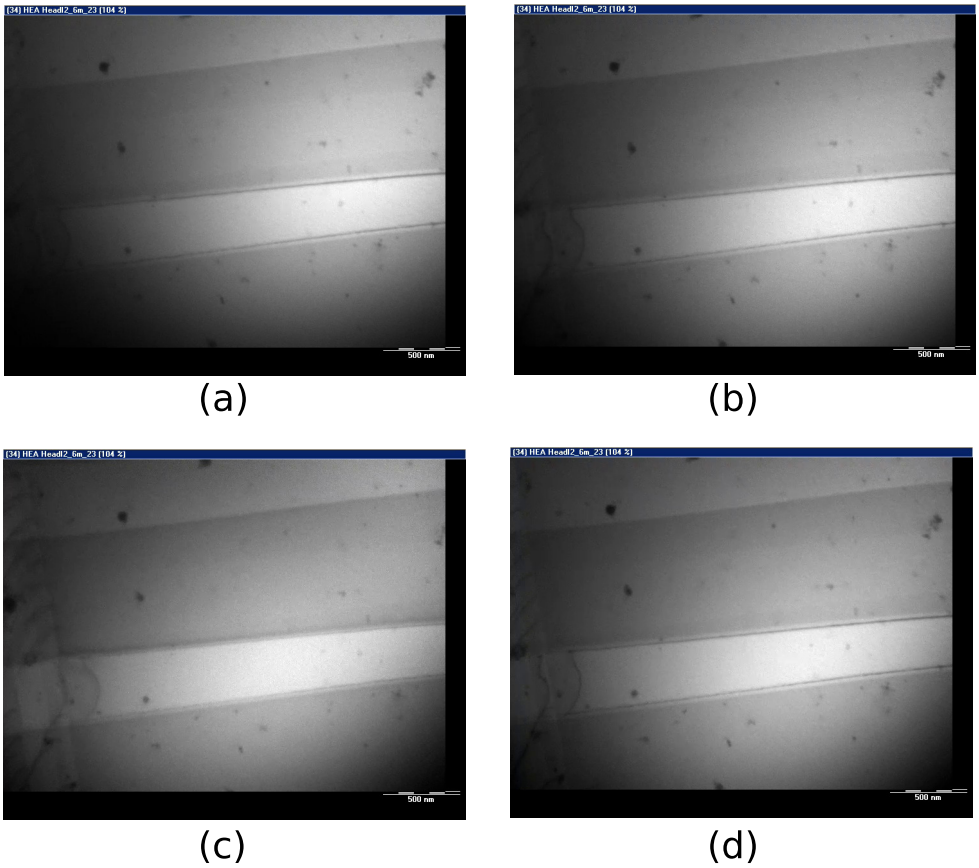


Figure 1.10: Representative frames from a TEM video illustrating varying image quality and contrast for dislocation visibility. (a) and (b) display low contrast, making dislocations difficult to discern; (c) shows motion blur that further obscures features; (d) exhibits sufficiently high contrast and sharpness to clearly resolve dislocation lines.

TEM images often suffer from low contrast, blurring, and noise, which hinder the accurate identification and analysis of dislocations in the context of Deep learning. Let us consider the four TEM images from an experiment as shown in Figure 1.10. The four images shows the same dislocation microstructure but vary in quality in terms of contract and blurriness. The images (a)

and (b) have very low contract and the dislocations are hardly visible. It happens because of adjustment of microscope during the experiment to visualize dislocations. Sometimes we might also get very blurry images shown as image (c). After making the adjustments we might get an image like (d) where dislocations of interest are very clearly visible.

Different materials and varying TEM operating conditions—such as beam energy, specimen thickness, and orientation—significantly affect the appearance of dislocations in TEM images. This variability introduces additional complexities in developing a generalized approach for quantitative analysis to perform high throughout studies, as models trained under specific conditions may not perform well under different settings.

We explore the effects of varying image quality in the following by using a classical image segmentation approach. When training data is scarce for deep learning approaches, such methods can become particularly valuable, as demonstrated by Sills and Medlin [39].

1.5.3 Classical Image Segmentation Approach: Sauvola Thresholding

As a foundational approach to dislocation segmentation in microstructural images, we first explore classical image segmentation techniques. These methods, such as thresholding algorithms, offer a straightforward and computationally efficient means to isolate features of interest within an image. To evaluate the effectiveness of classical image processing techniques, we consider two real images, as shown in Figure 1.11. Prior to applying thresholding methods, we performed histogram equalization on these images to enhance their contrast and quality. This pre-processing step is crucial for improving the visibility of dislocation lines, which often exhibit low contrast in raw images.

Classical approaches such as Canny edge detection, contour detection, watershed segmentation [41], and Chan-Vese segmentation [42] can be used for image segmentation. Among the various techniques available, we select the Sauvola thresholding method [43] due to its ability to handle images with varying illumination, and contrast—common characteristics in TEM images of dislocations.

The Sauvola thresholding method is an adaptive technique that computes a local threshold for each pixel based on the statistical properties within a neighborhood window. This method is particularly effective for images with uneven lighting or varying background intensities. The threshold $T(x, y)$ at a pixel location (x, y) is calculated using the following equation:

$$T(x, y) = m(x, y) \left[1 + k \left(\frac{s(x, y)}{R} - 1 \right) \right] \tag{1.5}$$

where:

- $m(x, y)$ is the mean intensity of the pixels within a local window, $W \times W$ centered at (x, y),
- $s(x, y)$ is the standard deviation of the pixel intensities within the same local window,
- R is the dynamic range of standard deviation, typically set to 128 for grayscale images,
- k is a parameter that controls the sensitivity to the local standard deviation, commonly set between 0.2 and 0.5.

The Sauvola method adjusts the threshold dynamically based on local image statistics. In regions where the standard deviation $s(x, y)$ is low, the threshold approaches the mean intensity $m(x, y)$, minimizing the detection of false edges due to noise. Conversely, in regions with high variability (edges and textures), the threshold is lowered, allowing for the detection of significant features like dislocation lines.

The parameter k plays a critical role in fine-tuning the thresholding sensitivity:

$$k = \begin{cases} \text{lower values (e.g., 0.2)} & \text{less sensitive to local variations, useful for noisy images,} \\ \text{higher values (e.g., 0.5)} & \text{more sensitive, useful for detecting fine details.} \end{cases}$$

The application of Sauvola's method, as shown in Figure 1.11, uses parameters $k = 0.2$, $R = 128$, and a window size of $W = 51$, which were found to yield effective segmentation of dislocations.

For image 1, while the method successfully segmented most dislocations, it also detected slip trace lines due to the inability to differentiate between dislocation lines and slip traces. Additionally, some dislocations were only partially segmented, resulting in incomplete dislocation line detection. In contrast, the results for image 2 were more consistent, with all five dislocations successfully segmented. However, in both images, the method incorrectly classified other artifacts as dislocations, highlighting limitations in its specificity.

Classical image segmentation methods can be very useful and provides a low computational means compared to deep learning based approaches but there are several limitations of classical image segmentation approaches like Sauvola's thresholding when applied to dislocation microstructures. Such methods rely heavily on local intensity variations and may struggle with images where dislocations have varying contrast or are obscured by noise and artifacts. This is usually the case with the dislocation image data where one might be changing the microscope settings which can result in varying imaging conditions. This can result in incomplete or inaccurate segmentation of critical features like dislocation endpoints. Improving the results of the two images shown in Figure 1.11 may involve refining the thresholding parameters, incorporating post-processing techniques such as morphological filtering, or using additional information like

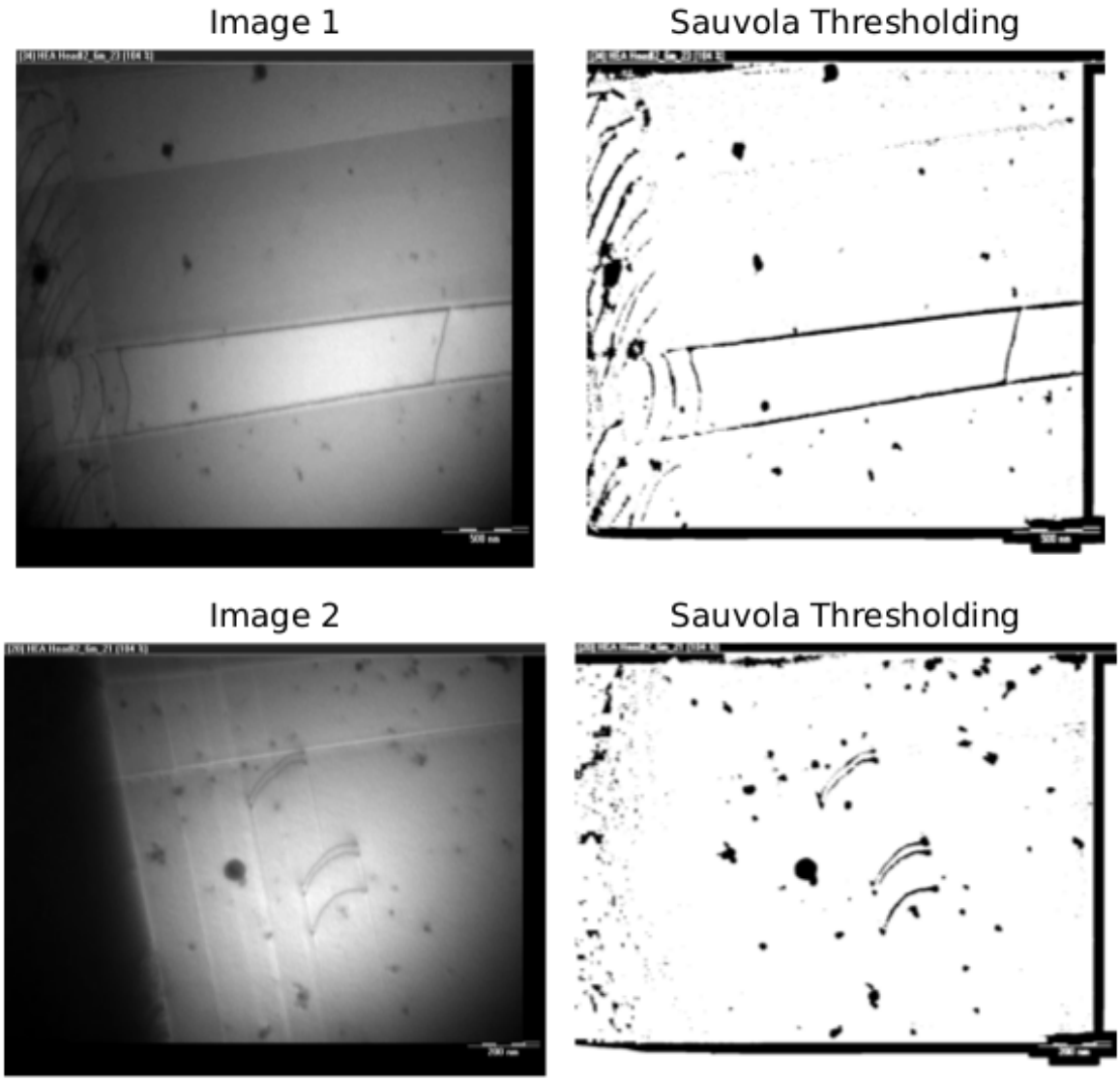


Figure 1.11: Segmentation of dislocations in a TEM image using Sauvola's binarization method. The original image has been histogram–equalized to enhance contrast before applying Sauvola's local adaptive thresholding. The result illustrates both the capability and limitations of this approach: while many dislocations are detected, low-contrast features or ones with subtle contrast variation may be missed or ambiguously segmented.

edge orientation or local intensity gradients to better distinguish dislocations from other features. Despite these enhancements, the inherent limitation of incomplete segmentation of dislocation lines is likely to persist due to the method's sensitivity to variations in intensity and noise around a dislocation line, which can disrupt continuity.

Classical methods lack inherent mechanisms to differentiate between actual dislocations and other linear or nonlinear artifacts present in the image, leading to false positives i.e., a grain boundary can be easily identified as a very long dislocation. This requires additional post-processing steps, such as filtering based on geometric properties, to improve accuracy. The effectiveness of these methods is highly sensitive to parameter selection (e.g., values of k, R, and W), which may not generalize well across different images or datasets. While classical segmentation techniques can provide a useful baseline, they may not offer the robustness and precision required for detailed analysis of complex dislocation microstructures and automating spline representation of dislocations in TEM.

1.5.4 The Need for Automated Analysis in High-Throughput TEM Imaging: Deep Learning Based Approaches

As discussed above, the traditional image processing methods are frequently insufficient for accurately segmenting and identifying dislocations, especially in images with low signal-to-noise ratios or significant feature overlaps. These techniques struggle with the nuances of dislocation structures, such as variations in contrast and the presence of defects, demanding more advanced algorithms for precise analysis.

Once high-quality dislocation image data is obtained, the next critical step is the digital representation of dislocations. Currently, this digitization is predominantly performed manually. Annotation tools like `labelme` [1] allow users to obtain dislocations by selecting points along the lines. However, as demonstrated in Zhang et al. [35], manually extracting dislocation information from hundreds of frames—each containing up to 20 dislocations—is a tough and time-consuming task.

Manual labeling is also difficult to reproduce and heavily depend on the experience and judgment of the individual performing the annotation. This subjectivity can introduce inconsistencies, which are particularly problematic for calculations sensitive to the local radius of curvature of dislocation lines, such as stress field estimations. With the appearance of faster detectors and cameras, TEM experiments now generate vast amounts of image data at unprecedented rates. This surge in data volume creates an urgent need for automated analysis techniques capable of handling high-throughput TEM imaging. Automation would not only expedite the analysis process but also enhance accuracy and reproducibility, enabling researchers to more effectively interpret complex dislocation behaviors and their impact on material properties.

To address these challenges, advanced computational methods, including machine learning and computer vision algorithms, are being explored. These methods have the potential to automatically segment and track dislocations across frames, handle low-quality images, and reduce the reliance on manual intervention. Implementing such automated analysis tools is essential for advancing materials science research and fully leveraging the capabilities of modern TEM technology.

Deep Learning methods, particularly convolutional neural networks like the U-Net proposed by Ronneberger et al. [44], have shown great promise in performing pixel-wise classification for image segmentation. Pixel-wise classification refers to the process where each pixel in an image is assigned a class label, enabling precise description of features within the image. This fine-grained approach allows for detailed segmentation of dislocations which is crucial to obtain spline representation. Methods utilizing pixel-wise classification have been successfully applied in various fields, including computer vision [45–47] and medical imaging [48–50], where they contribute to tasks like object detection, semantic segmentation, and lesion identification. In materials science, these methods have been utilized to segment nano-particles [51] and identify precipitates, voids, and simple dislocation networks [52]. Additionally, Shen et al. [53] demonstrated deep learning-based segmentation approach to identify small defect loops.

Despite their potential, Deep learning methods face significant challenges when applied to TEM images of dislocations. Deep learning models typically require extensive training data, which is often unavailable in this scientific domain. Generating ground truth data requires manual annotation, a process that is both time-consuming and labor-intensive. The appearance of dislocations in TEM images varies greatly due to differences in material properties and imaging conditions. Creating a representative training dataset that includes all possible configurations is impractical, as it would require a huge number of experiments, including those covering rare or transient phenomena. The scarcity of diverse datasets leads to models that perform well on training data but generalize poorly to new, unseen data—a problem known as overfitting. This issue is further complicated by the absence of publicly accessible datasets of dislocation microstructures, limiting opportunities for data augmentation or transfer learning. Sasaki et al. [54] used the first 100 frames of a TEM video for training and the subsequent 70 frames for testing. However, without quantitative performance evaluation and with training and testing data derived from the same video, overfitting remains a concern. Roberts et al. [52] divided two high-resolution images into five parts, using three for training, one for validation, and one for testing. Basic augmentation increased the dataset to 48 images, but the Intersection over Union (IoU) performance for dislocations was only 44%, indicating limited accuracy.

Deep learning-based methods such as binary segmentation [52, 54], instance segmentation [53], object detection, and keypoint detection [55] have gained significant popularity in providing a robust and automated means to analyze images in material science and perform data driven studies. In their study, Chowdhury et al. [56] applied deep convolutional networks for grain

boundary detection, addressing challenges in pixel-wise segmentation that often resulted in gaps and incomplete boundaries. The segmented masks were used to perform data driven study in a automated manner where microstructural features like grain size, grain shape were used to perform microstructural characterization.

These approaches highlight a significant challenge in applying deep learning to microstructure analysis: the difficulty of obtaining sufficient and varied training data to develop robust models. The scarcity of labeled datasets in materials science often hampers the performance and generalization of deep learning models. For instance, Chen et al. [57] trained their model on only nine samples and had to rely heavily on data augmentation techniques to improve the model's robustness. This limitation highlights the need for strategies to overcome data constraints, such as generating synthetic datasets, employing transfer learning from related domains, or developing semi-supervised and unsupervised learning methods that require less annotated data. Addressing these challenges is crucial for advancing automated microstructure analysis using deep learning.

1.6 Approaches to Obtain Synthetic Training Data

The state of the art deep learning methods described above are supervised machine learning methods and hence require training data (images and ground truths). One of the most straight forward method is to perform a number of experiments and obtain real images of dislocation microstructures. These images can then be labelled manually for dislocations, which will provide the training data for deep learning. But performing such a large number of experiments and then hand-labelling the data is usually not possible or feasible and hence there is need to explore alternate data generation methods. The lack of high-quality *and* high-quantity training data can be overcome by adopting synthetic data generation methods [58–60] which can be categorized as follows.

1.6.1 Generative AI Based Methods

Recently, generative AI based methods have made significant advances in generating high-quality synthetic images. These methods can be extended to generate not only images but also their corresponding ground truth in a paired manner. For instance, Thambawita et al. [61] used a Generative Adversarial Network (GAN) to generate synthetic colon polyp images along with their corresponding segmentation masks, enabling comprehensive data augmentation for medical imaging tasks.

Synthetic data generation is an approach which so far has only been rarely used in the field of materials science [62, 63]. There are several techniques available to generate synthetic images

ranging from domain randomization [64] where non realistic objects are added to force the machine learning models to learn important features, to machine learning guided methods such as, e.g., GAN [61] where a machine learning model learns to generate synthetic data with features similar to real data. Chun et al. [65] used a GAN to generate synthetic heterogeneous energetic material microstructures. Haribabu et al. [66] used more advanced Deep Convolutional Generative Adversarial Networks (DCGAN) and StyleGAN to generate synthetic images of microstructures. The model was trained on a limited amount of training data consisting of only 1225 images obtained from experiments. They observed that the model not only produced high quality synthetic images which were qualitatively similar to real images but the generated microstructure has statistical properties i.e., phase distribution, morphology similar to real images. These approaches are primarily effective at generating synthetic images alone. However, there is a need for specialized methods capable of simultaneously generating both synthetic images and their corresponding ground truth masks in a paired manner, enabling tasks that require both the synthetic image and its accurate segmentation mask.

More advanced deep learning approaches, such as text-to-image generative models like "Stable Diffusion" [67–69], offer high-quality synthetic images along with paired ground truth for segmentation tasks. Diffusion models [70], in particular, have emerged as one of the most prominent generative AI techniques due to their ability to generate realistic synthetic images. Diffusion models operate based on the principle of denoising, wherein they learn to iteratively reverse a process that adds Gaussian noise to the data over a series of timesteps.

To illustrate diffusion models, consider a TEM image of a dislocation as shown in Figure 1.12. Initially, we start with a clean TEM image, and throughout the forward diffusion process, Gaussian noise is incrementally added until the original structure is completely destroyed. During the reverse process, the model is trained to progressively remove noise, reconstructing the original image with high fidelity.

This process involves two main components: the forward process and the reverse process.

1. **Forward Process:** In the forward process, dislocation image is incrementally noised over T timesteps, transitioning from an image distribution $q(\mathbf{x}_0)$ to a noised distribution $q(\mathbf{x}_T)$. At each timestep t, Gaussian noise is added to the image, transforming $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ according to the distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$, where β_t are variance schedules specifying the amount of noise added at each step.

2. **Reverse Process:** The reverse process aims to learn the conditional distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, effectively denoising the image by learning to predict the noise that was added at each step and subtracting it. This is modeled using a neural network that parameterize the reverse transition. The network is trained to minimize the difference between the original data and the denoised output, using a loss function based on variational lower bounds.

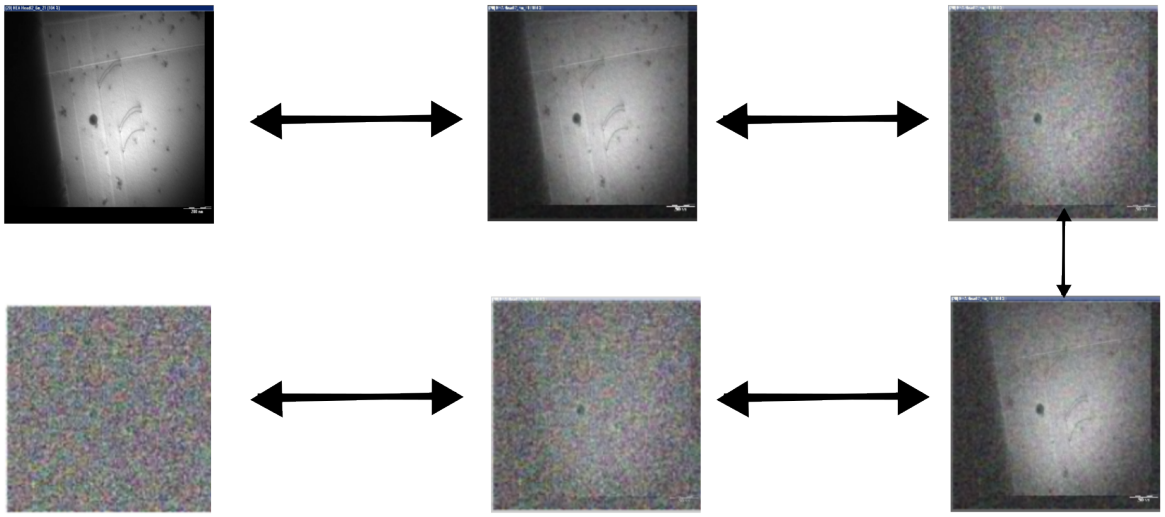


Figure 1.12: Illustration of the diffusion model applied to TEM images containing dislocations. The forward process progressively adds Gaussian noise to a clean TEM image over T discrete timesteps until the structural details are completely obscured. The reverse process is trained to undo this noise—restoring the image by predicting and removing the noise at each step. This enables generation of synthetic TEM images with dislocation microstructures starting from purely noisy inputs.

The training of diffusion models involves optimizing the parameters θ of the neural network to improve its ability to reconstruct the original image from noised versions. Once trained, generating new samples involves sampling from the noise distribution $q(\mathbf{x}_T)$ and iteratively denoising through the learned reverse process to obtain $\mathbf{x}_0$.

Diffusion models have demonstrated remarkable capabilities in generating detailed and diverse images, making them highly effective for tasks that require high-quality synthetic data, such as in medical imaging where they can generate both images and their corresponding segmentation masks [70].

Despite these advancements, the application of such deep learning approaches to specialized fields like dislocation microstructure remains limited, revealing a gap in their adaptability for specific datasets. One of the drawback of such methods is that they need a large amount of training data in the amount of atleast hundreds of samples to train the models. This requirement poses a significant challenge, particularly when the available training data is sparse. Studies, including [71], have documented instances where models, constrained by limited data, begin to generate repetitive images, failing to capture the diversity of the training examples. This phenomenon, known as mode collapse, indicates a need for methodological enhancements to ensure models can learn effectively from limited datasets without compromising the variability

and richness of the generated images.

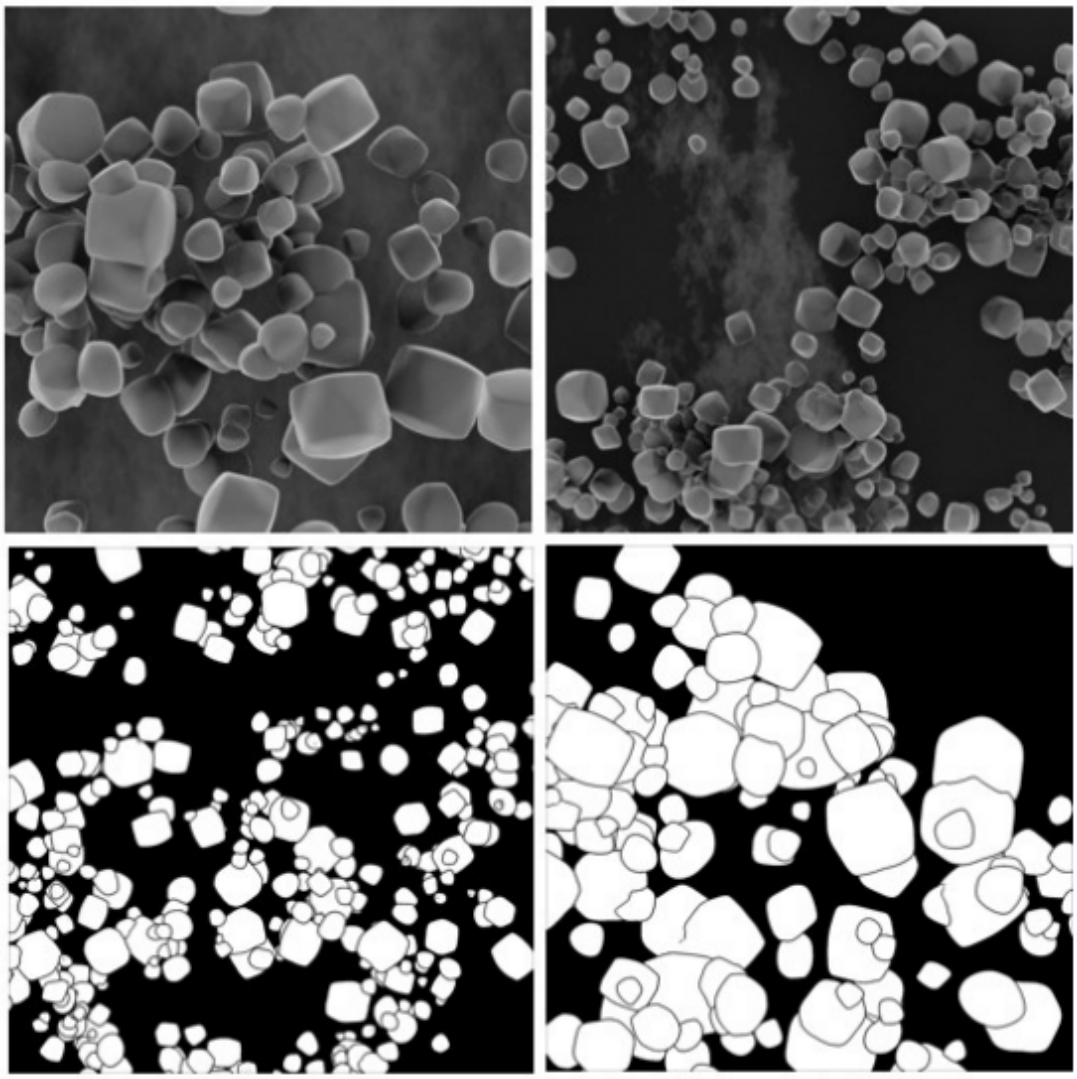


Figure 1.13: Synthetic nanoparticle images and corresponding ground truth segmentation masks generated by rendering 3D scenes using Blender. Realistic background textures, lighting, object shapes and arrangements are controlled to mimic experimental microscopy images. This semi-automated rendering approach enables generation of paired image-mask datasets useful for training segmentation models without extensive manual annotation. Adapted from Mill et al. [51]

1.6.2 Image Rendering Based Methods

A different class of approaches for image generation involves the use of rendering software, such as Blender [72]. While the focus is primarily on lighting and rendering three-dimensional objects, these tools can also be used to generate two-dimensional synthetic image data. In such methods, two important components are considered: the first is generating a realistic background, and the second is rendering the objects of interest onto that background. This controlled approach allows for the generation of synthetic images with corresponding ground truths.

Examples of applications involving highly realistic images as shown in Figure 1.13 in the context of nano-particles can be found in Mill et al. [51], Cid-Mejías et al. [73], where nano-particles were rendered using Blender to generate synthetic data. Such approaches are usually semi-automated, as rendering settings like lighting need to be manually adjusted to achieve realistic images.

In some cases, it might be difficult or time-consuming to develop models for the objects of interest in such software, but still, the approach has gained popularity for generating high-quality synthetic training data. In a slightly different approach, when we have access to a large variety of real images containing the objects of interest, we can render the objects directly from the real images, as done in Toda et al. [74]. In the referred work, barley seed patches obtained from real images were directly rendered onto synthetic backgrounds, allowing the generation of high-quality synthetic data without using any rendering software. The advantage of such approaches lies in their simplicity and ability to accurately capture the statistical distribution of the objects of interest. These approaches can be particularly helpful in high-density images, where there are tens of objects and manual labeling can be extremely time-consuming [75, 76].

1.6.3 Simulation Based Methods

Another possibility in the field of material science is to use simulation-based methods to create *synthetic images* [77]. Simulation-based methods can capture the underlying phenomena much more accurately compared to rendering software like Blender. This could be useful where trained ML models can also learn more about the physical laws governing the system. There are, in principle, many methods and models readily available, covering phenomena from atomistic to macroscopic scales. Such simulations can be used to generate synthetic data, as shown by Hajilounezhad et al. [78], where a scanning electron microscopy (SEM) simulation tool was used to obtain artificial images of a carbon nanotube forest along with the calculated properties of the structures used using actual data. In another study, microstructure representations were generated using simulation models, which were then rendered to obtain simulated images [79]. Trampert et al. [62] used Voronoi tessellations to generate synthetic polycrystalline microstructures, where the statistical properties of the grain size distributions were comparable to those reported for real

grain distributions. The degree of visual similarity and the type of contained features (e.g., triple points) were sufficient for an effective training process.

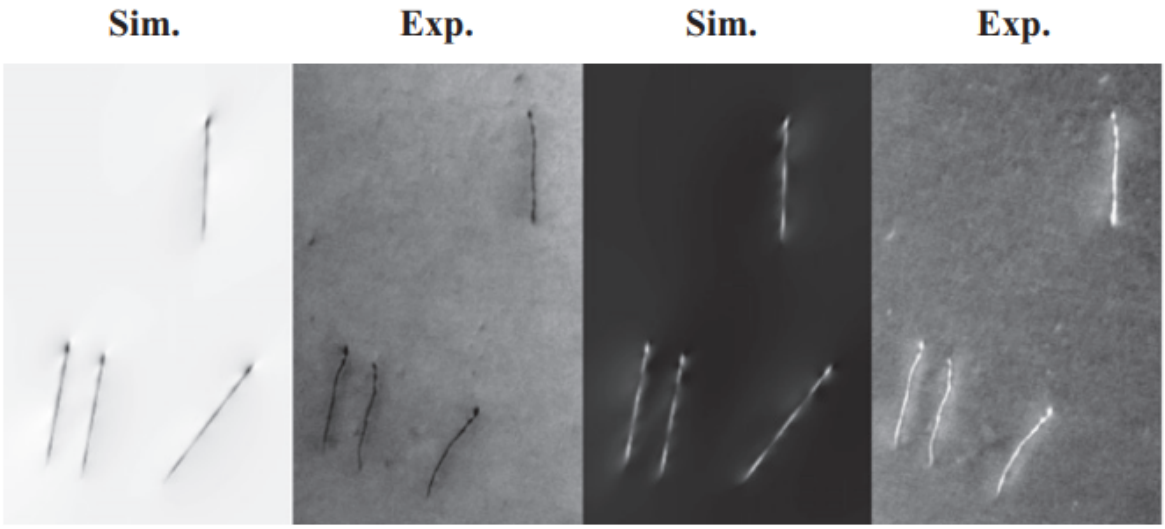


Figure 1.14: Experimental and simulated STEM images. The simulated images are generated by solving the Darwin–Howie–Whelan dynamical multi- beam equations. Adapted from Phillips et al. [80]

Developing similar simulation-based methods to model the TEM imaging of dislocation microstructures is much more complex due to the complicated and nonlinear interactions of electrons with atoms and defects present in the materials. Since the early 1900s, marked by seminal contributions such as those by Humble [81], the field has seen a transformative evolution in our understanding and capabilities. Initial approaches were grounded in simplified assumptions based on two-beam theory aimed at developing a foundational understanding of electron-material interactions within the TEM framework. This theory simplifies the complex interactions of electrons with the specimen by considering only two diffraction beams: the directly transmitted beam and one diffracted beam. These simplifications were necessary due to the limited computational resources and theoretical models available at the time, focusing primarily on linear approximations of electron scattering phenomena. Since then, simulation-based methodologies have emerged as pivotal tools for TEM imaging of dislocations, offering insights into the electron-material interaction dynamics within crystalline matrices. These methods, as referenced in the literature [80, 82–85], play a crucial role in deciphering the complex interplay between electrons and the myriad defects and atomic structures present in materials. Central to these simulations is the accurate modeling of both elastic and inelastic scattering phenomena encountered by electrons as they navigate through the specimen, which are instrumental in generating realistic TEM images and diffraction patterns. A cornerstone

among these techniques is the Multi-slice Method, detailed by Ishizuka [86, 87]. This approach conceptualizes the specimen as a series of thin, discrete layers, through which the electron wave function is meticulously calculated as it progresses.

By accounting for numerous scattering events, this method is exceptionally suited for high-resolution TEM (HRTEM) simulations, adept at handling complex specimen structures, including defects and interfaces, thus proving its significance in TEM imaging simulations. Complementing the Multi-slice Method, the state of the art Darwin-Howie-Whelan dynamical multibeam equations [88, 89] offer a robust mathematical framework for modeling electron scattering within crystalline substances. These equations delve into the intricate interactions between multiple electron beams and the crystal lattice, facilitating precise simulations of electron microscopy images. Unlike simpler kinematical scattering models, this dynamical approach comprehensively accounts for the potential multitude of scattering events, enabling electrons to "bounce" within the crystal lattice prior to exiting and contributing to the resultant image. This dynamical perspective is indispensable for achieving simulations that closely mirror the complexities observed in experimental TEM data as shown in Figure 1.14. The integration of advanced simulation-based methods, such as the Multi-slice Method and the Darwin–Howie–Whelan equations, into the TEM imaging significantly enhances our understanding of material structures. By precisely modeling the complex interactions between electrons and specimens, these methodologies enable the generation of detailed and realistic simulated TEM images.

One of the primary challenges in using such methods to generate synthetic training images is the difficulty of accurately modeling the full complexity and dynamic nature of defect configurations. Dislocations often exist in intricate networks and are influenced by local variations, strain fields, and the presence of other defects—phenomena that are not easily represented through straightforward atomic lattice models. These methods rely heavily on idealized atomic potentials and simplified boundary conditions. While such potentials can approximate average atomic interactions in a crystal, they often fail to reproduce subtle local bonding differences, compositional gradients, or other deviations found in real materials. The result is a simulation environment that, while physically grounded, may still differ substantially from the true complexity and variability encountered in experimental TEM data.

1.7 Objectives and contributions

Synthetic data can be a very useful way of obtaining training data where no suitable, "real" microscopy training data is available. However, up to date, there exists no systematic analysis of how to create synthetic training data for TEM images of dislocation microstructures and how to infer which features need to be included and which are superfluous.

In this work, we propose a parametric based synthetic data generation model to generate training data (images and ground truths[2]). The synthetic image can be conceptualized as consisting of two distinct components: the image background and the rendered representation of the dislocation microstructure superimposed onto this background. In this work, we have used two different methods to generate the background for the images. We use variations of Perlin noise along with some random white noise to generate a purely synthetic background for the image. In other method, we use a more realistic approach where we use background patches from real images and use them to generate a more realistic background. Once the background has been generated, we can start generating dislocation microstructure. In this work, we model a dislocation as a spline which is generated using a set of support points. These support points can either be generated using polynomial approximations of dislocation line or by using tools such as *Labelme* to manually pick points on a dislocation in a real image. We can generate much more complicated dislocation microstructures by generating a number of dislocation microstructures consisting of different slip width, slip direction, number of dislocations. The method is very powerful in generating dislocation microstructures that otherwise can not or only rarely observed in real TEM images.

Another major objective of this work is to conduct a systematic evaluation of deep learning-based methodologies for the characterization of dislocation microstructures. Specifically, we investigate and compare three distinct approaches: semantic segmentation which aims to classify every pixel of an image into categories corresponding to dislocation and background; instance segmentation which extends semantic segmentation by identifying individual dislocation lines as separate instances; and dislocation spline support point estimation which directly infers the parametric representation of dislocations as spline curves based on their support points. We analyze how these approaches perform under varying levels of complexity in the training data, assessing factors such as noise, background texture, and diversity in dislocation configurations. Furthermore, we explore how synthetic training data can augment model performance, with a focus on the interplay between synthetic and real datasets. By systematically varying the synthetic data used during training, we aim to understand its contribution to improving model generalization and robustness when applied to real TEM dislocation microstructures. This study provides critical insights into the strengths and limitations of each approach, laying the groundwork for optimal strategies in machine learning applications for dislocation analysis.

[2]Ground truths refer to the precise annotations or labels corresponding to the synthetic images, essential for supervised machine learning tasks. For dislocation image data it refers to pixel-level masks that define the precise segmentation of dislocations within the synthetic images.

1.8 Outline

The thesis is structured as follows. Chapter 2 provides a details description of the synthetic data generation method which includes both the background and synthetic dislocation microstructure of the images. This is followed by Chapter 3 where details of the three different machine learning approaches and the physics based metric which is based on the error in the lengths of the predicted dislocation splines. This is followed by Chapter 4 where we have performed two studies to answer some of the open questions related to the application of the synthetic datasets. To this end, we have provided details of generating three different synthetic datasets with increasing complexity and microstructure features. In Chapter 5, we compared the three models based on synthetic data and real data focusing on model robustness and ability to provide good results on new experimental data. The motivation is to analyze the models and applicability of synthetic data in developing a generalized framework for new unseen experimental data. Last but not least, we investigate how the results improve when the models are additionally specialized on real dataset using transfer learning approaches. In **??**, we conclude our studies and provide outlook of future work and further developments.

Chapter 2 Parametric Based Synthetic Data Generation

In dislocation image data, parametric based synthetic data emerges as a powerful alternative, enabling precise control over image complexity, variability, and the generation of ground truths. To be effective, synthetic data must satisfy two fundamental criteria. First, synthetic images must replicate the appearance of real TEM images, ensuring that machine learning models trained on synthetic data can generalize effectively to real datasets. Second, the synthetic data must capture the inherent variability of dislocation microstructures, including diverse textures, geometries, and spatial configurations, to adequately represent the complex nature of the underlying physics.

To address these requirements, we employ a structured three-step approach to generate synthetic data, designed to reproduce the essential characteristics of TEM images of dislocation microstructures while allowing for precise control and reproducibility. The steps are as follows:

1. **Generating a Background for the Synthetic Image:** Two fundamentally different approaches are used in this work. The first approach is completely artificial, using the superimposition of different types of noise and smoothing operations. The second approach involves using background patches from real microscopy images. This is discussed in details in Section 2.1.

2. **Generating the Geometry of the Artificial Dislocation Microstructure:** In this step, the position and shape of the dislocations, along with any other elements of interest, are determined. Details of the microstructure generation and parametric models can be found in Section 2.2.

3. **Rendering and Outputting the Image:** This is the final step where rendering of the dislocations onto the background, writing the resulting image to a file (i.e., in PNG format), generating corresponding ground truth masks, and recording all parameters in a JSON file to ensure full reproducibility.

An overview of the three steps together with a list of the most important parameters are shown in Figure 2.1. In the sections that follow, all the steps and the involved parameters are explained in more details.

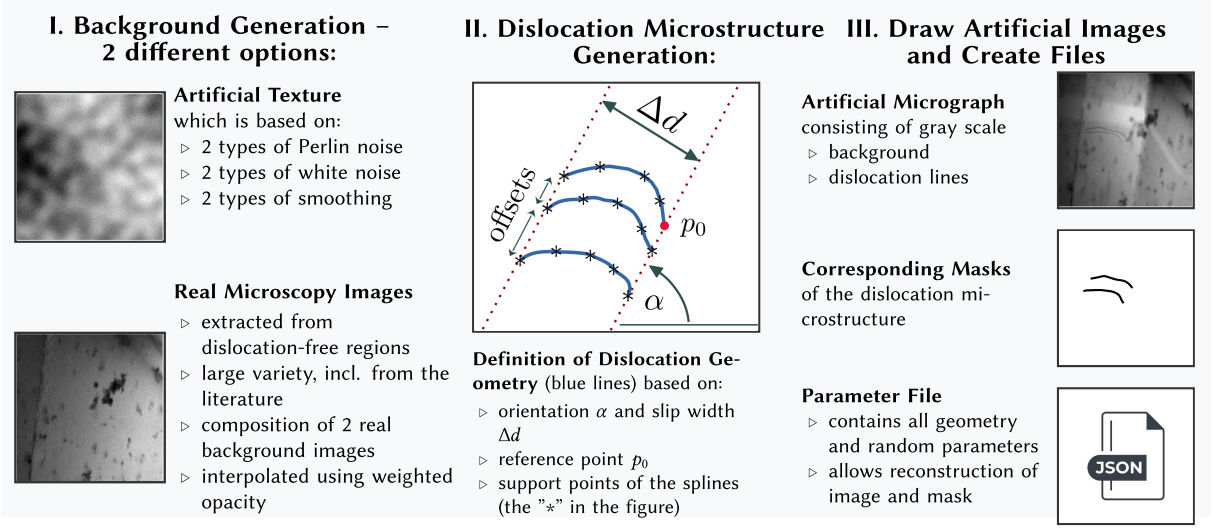


Figure 2.1: Synthetic images and masks are generated in three steps. First, a background image is created from artificial "noise generators", or by using background images from real microscopy data Then, the dislocation geometry is determined and subsequently "drawn" on top of the background providing the synthetic image, ground truths as well as the parameter file.

2.1 Modeling of Background for Synthetic Images

In this section, we will present two approaches to generate background from synthetic images. The first approach is synthetic in nature where we use Perlin noise to generate a synthetic background and the second approach is more realistic and based on background patches from real images. These two approaches are discussed in details below.

2.1.1 Synthetic Background: Perlin noise

Noise, randomness, and a well-chosen variance of different features in the training dataset are key ingredients for a successful training process. These factors help reduce over-fitting and enhance the generalization capability of the model to new unseen images. Consequently, generating suitable background textures is an essential task. We analyzed real microscopy images and observed that they often exhibit complex gray value gradients resulting from variations in imaging conditions, such as electron beam intensity or specimen thickness.

Additionally, random fluctuations in brightness, likely caused by noise inherent to the imaging process, further contribute to the heterogeneity of these images. The fluctuations may also arise

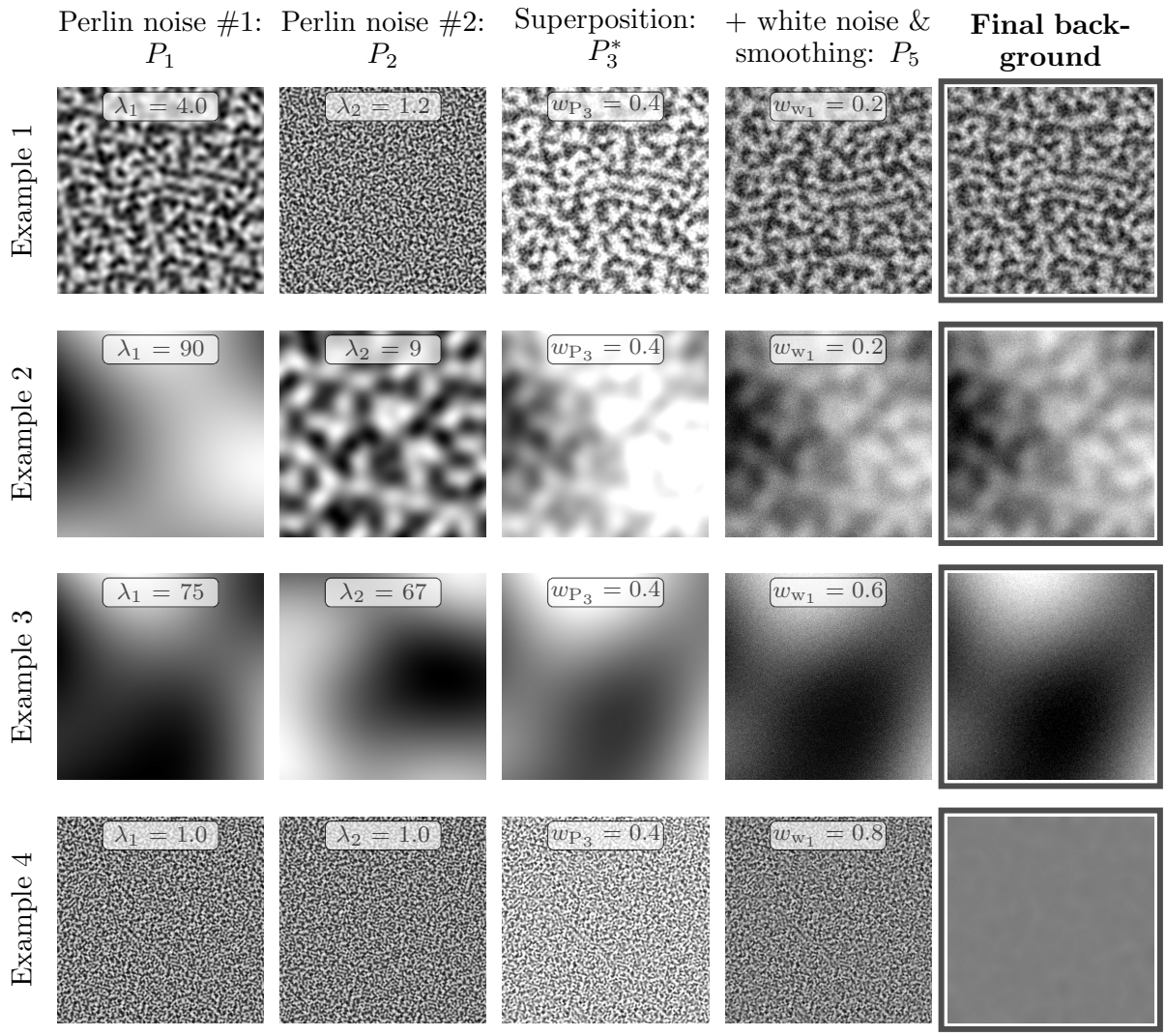


Figure 2.2: The rows show four examples for synthetically created background images. The most important steps of the background creation pipeline together with the used parameters are shown from left to right.

from random "dirt" particles on the specimen's surface or other barely visible defects caused by diffraction conditions. This variability is illustrated in Figure 2.3, which shows background patches extracted from several real microscopy images, highlighting the diverse textures and intensity distributions characteristic of TEM data.

To generate synthetic background textures, we used two-dimensional Perlin noise, a technique widely used in computer graphics for texture generation [90–92]. Perlin noise was selected due to its natural-looking structure and non-repetitive characteristics, making it suitable for simulating irregular patterns within an image. The Perlin noise algorithm works by assigning random gradients at fixed grid points in a two-dimensional space, and then using interpolation to produce noise with a dominant wavelength. In this work, we utilized the Python implementation provided by [93]. Although Perlin noise does not fully replicate the complex backgrounds of real TEM images, as shown in the first two columns of Figure 2.2, it provides a form of structured randomness that is notably superior to white noise. Unlike white noise, which lacks spatial correlation and produces abrupt, grainy patterns, Perlin noise generates smooth gradients and coherent textures by interpolating random gradients across a grid. This spatial coherence makes it more effective at simulating the gradual intensity variations and mid-frequency textures often observed in TEM images. While certain statistical properties, such as brightness distribution and wavelength spectrum, still differ from those of real TEM backgrounds, Perlin noise remains a versatile and computationally efficient tool for creating diverse synthetic textures that better approximate the irregular patterns seen in microscopy data.

In this study, we used superposition of Perlin noise at two distinct wavelengths. The longer wavelength simulates global gray value gradients typically seen across TEM images, which may result from variations in imaging conditions. The shorter wavelength, on the other hand, accounts for local fluctuations in texture. Additionally, to refine the texture, we introduced white noise and applied Gaussian filtering twice, with different parameter settings, to achieve more nuanced variations in the generated backgrounds.

To mathematically formulate the subsequent steps we introduce an image with $M \times N$ pixels as a function I that maps each discrete pair of pixel coordinates, $(i, j) \in \{1, \dots, M\} \times \{1, \dots, N\} \subset \mathbb{N} \times \mathbb{N}$ to a gray scale intensity that can take values within the unit interval $[0, 1] \subset \mathbb{R}$

$$I : \{1, \dots, M\} \times \{1, \dots, N\} \to [0, 1] \tag{2.1}$$

$$m, n \mapsto I(m, n) \ , \tag{2.2}$$

where (m, n) denotes an individual pixel with the intensity $I(m, n)$. Similarly, any image operation T is a function that maps the value of each pixel $I(m, n)$ of an image I to a new value $T(I(m, n))$. By chaining such functions one can also perform several subsequent operations on and with images.

For the problem at hand, we start with two image generating functions each of which results in an image with Perlin noise, P_1 and P_2 as shown in the first two columns of Figure 2.2. Structural

details of the Perlin noise image depend on a wavelength parameter λ_1 and λ_2, respectively. A new image P_3 (see third column of Figure 2.2) is obtained by adding the corresponding pixel values of two images and additional weighting P_2 with a factor $w_{\mathrm{rm}P}$, resulting in

$$P_3 = P_1 + w_{\mathrm{rm}P} \times P_2 \ . \tag{2.3}$$

Subsequently, the value range is scaled to the unit range by the following operation

$$P_3^* = \frac{P_3 - \min(P_3)}{\max(P_3) - \min(P_3)} \in [0, 1] \ , \tag{2.4}$$

where $\min(P)$ and $\min(P)$ give the minimum and maximum pixel value of an image P, respectively. In the next step, an image X_1 which contains white noise is added. It is obtained by sampling each pixel value from a uniform random distribution with values in between 1 and -1. The values of X_1 are then weighted with $w_{\mathrm{rm}w_1}$ and superimposed with the previous data to obtain

$$P_4 = P_3^* + w_{\mathrm{rm}w_1} \times X_1 \ , \tag{2.5}$$

The resultant is then (usually: only slightly) smoothed using a convolution with a discrete Gaussian filter kernel that has standard deviation s_1. This smoothing function G_1 then results in the image P_5:

$$P_5 = \ G_1(P_4; s_1) \tag{2.6}$$

The operations resulting in P_4 and P_5 lead to slightly "smeared out" random fluctuations which might, e.g., stem from the noise of the microscope electron optics or the camera noise. We then add additional white noise X_2 weighted by the standard deviation of the values of P_4, i.e., $w_{\mathrm{rm}w_2} = \sigma_{\mathrm{rm}P_4}$, and apply one more Gaussian filter G_2 with standard deviation s_2 to adjust the width of these spikes,

$$P_6 = \ P_5 + w_{\mathrm{rm}w_2} \times X_2 \tag{2.7}$$

$$P_7 = \ G_2(P_6; s_2) \ . \tag{2.8}$$

Due to the scaling of this final white noise, P_7 might also contain negative values. Along with a "clipping" of the data to the range of $[0, 1]$ this gives the final background image P:

$$P = \min(1, \max(0, P_7)) \ . \tag{2.9}$$

Adding the noise X_2 together with the smoothing of G_2 only slightly changes the image but in conjunction with the clipping, gives sharper, high frequency fluctuations. The combination of the two noise types X_1 and X_2 and the smoothing operations G_1 and G_2 is also a way to mimic artifacts from the usually lossy image compression often used in proprietary microscopy software.

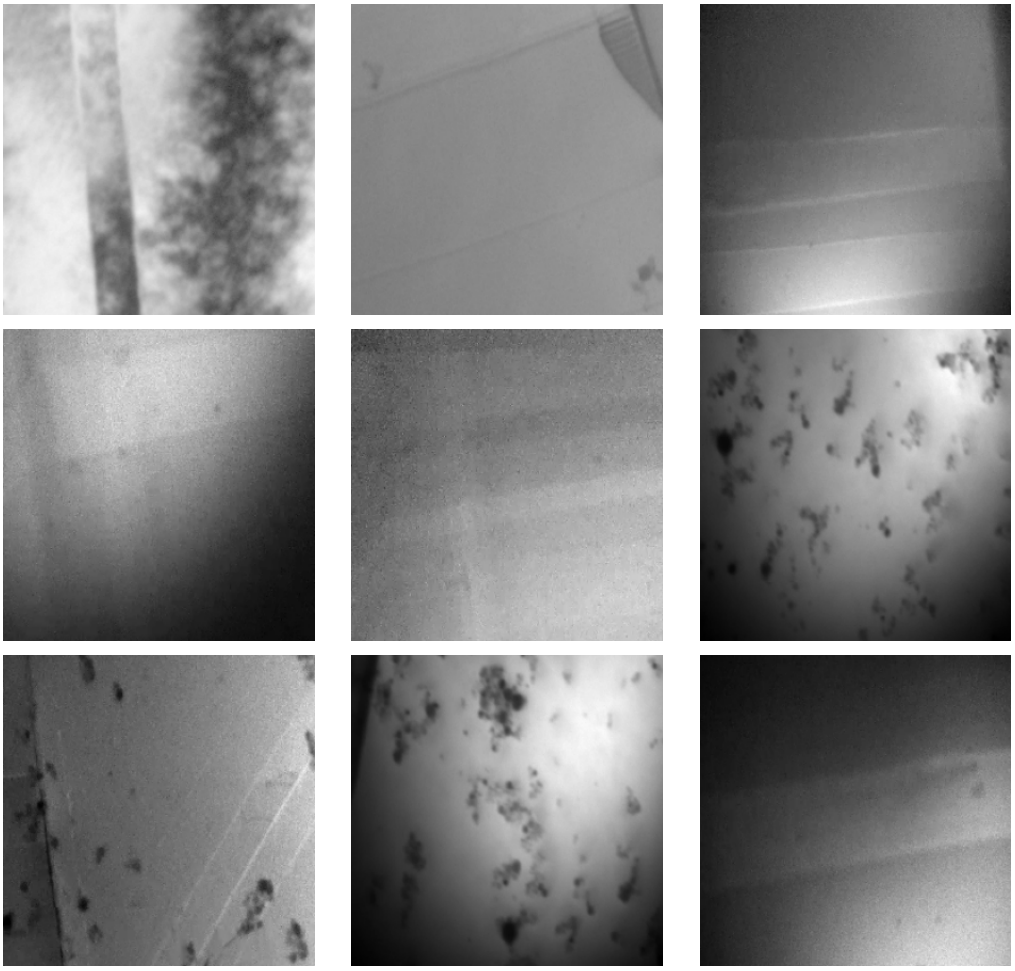


Figure 2.3: Example background image patches used to create realistic and contextually accurate backgrounds for synthetic image generation. These patches are gathered from diverse sources to ensure variability and enhance the realism of the synthesized images.

In Figure 2.2 we show some examples for intermediate steps and the final background together with the respective parameters. The images have a resolution of 512×512 pixels. In the first two examples, only the wavelengths of the Perlin noise contributions are varied.There, it can be observed that textures with a very different wavelength and characteristics can be reproduced.

Some of them resemble backgrounds of real microscopy images, but not all of them do. Below, we will also study the importance of the degree of realism. All parameters as well as the seeds for the random number generators were recorded and stored in a JSON file so that the synthetic background could be fully reconstructed.

2.1.2 Realistic Background: Real Image Patches

The second approach to generate background images involves utilizing patches from real images to produce highly realistic textures. We collected 170 TEM images featuring dislocation microstructures, focusing on regions that were largely devoid of dislocations. These "background-only" areas were cropped and extracted to obtain background patches. Some examples for these background patches are shown in Figure 2.3.

The images were gathered from an extensive body of literature [32, 94, 95] and supplemented with experiments conducted by our collaborators, resulting in a diverse dataset. Note, that the collection of background images was kept strictly separate from the images in the following chapters for ML predictions tasks and evaluations. In other words, we ensured that there is no hidden leakage between the training and testing datasets. To further enhance the diversity of the generated data, two background images are randomly selected and superimposed with a randomly assigned opacity value. While the resulting super-imposed image may appear somewhat atypical or "incorrect" to a trained expert, the presence of additional background features, when compared to typical real backgrounds, it might prove beneficial for improving the generalization capability of our ML model on real-world data.

To ensure the reproducibility of this background generation method, we document the two images and the opacity values used during the superimposition process. This comprehensive documentation facilitates replication of our methodology and enables other researchers to build on our findings in their own investigations.

2.2 Generation of Dislocation Microstructure

Typical dislocation microstructures considered in this work features three of the most important characteristics typically observed in crystalline materials under stress: (1) surface impurities, dirt, or second phase particles; (2) dislocations; and (3) slip trace lines left by moving dislocations. In bright field TEM, dislocations are seen as dark lines. An image of a typical dislocation microstructure is shown in Figure 2.4. The motion of dislocations under applied stress is often hindered by various obstacles, such as second phase particles, grain boundaries, and other imperfections, which prevent free movement and may lead to the formation of dislocation pileups. A dislocation pileup, as shown in Figure 2.4, consists of a group of dislocations moving along the same slip plane, accumulating at obstacles. The pileup occurs because successive dislocations cannot continue to move when blocked by e.g., grain boundaries, leading to a buildup of dislocations behind the leading one.

When dislocations exit the crystal lattice, they leave behind slip trace lines at the surface

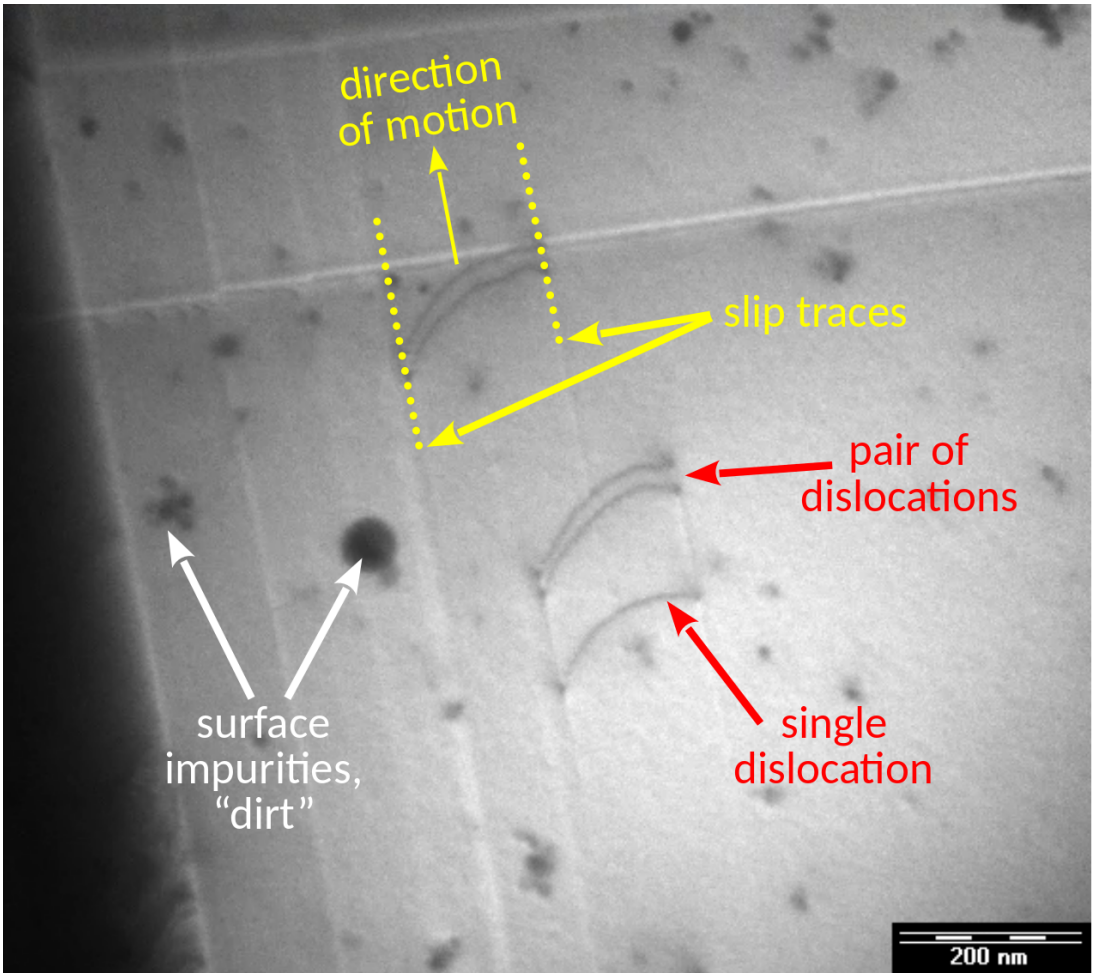


Figure 2.4: A typical real dislocation microstructure consists of one or many dislocation pileup. As the dislocations move, distinct slip trace lines could also be soon. Among other things, very often surface impurities/dirt/second phase particles can also be soon.

whose parts of the crystal are displaced relative to each other. This can also often be seen in TEM images as dark or light contract along with straight lines. These markings are located at the intersection of slip planes on which the dislocation moved and the surfaces of the thin film. Slip traces were one of the earliest significant observations that helped scientists understand the behavior of dislocations when electron microscopy began revealing the crystalline structure of metals in the mid-20th century. In Figure 2.4, these traces are visible in regions through which dislocations have already moved, but no traces are present in areas above the leading dislocation, as it has not yet passed through those regions. In addition to dislocations and slip trace lines, surface impurities, contamination, and second phase particles are visible. These features appear as dark, irregular spots, as marked in the figure. While other features such as grain boundaries or

interfaces may also be present in the TEM image, they are not primary focus of this particular analysis.

2.2.1 Parametric Model of Dislocation Based on Spline

Dislocations can be effectively modeled as smooth curves whose intricate shapes stem from their interactions with complex local stress fields and microstructural inhomogeneities. Representing these curves in a synthetic microstructure requires a flexible and robust mathematical approach. Spline interpolation offers an ideal solution, as it can accurately reproduce the wavy, erratic, and sometimes sharply turning points that characterize dislocations. By defining a set of control points and fitting a spline through them, we can construct curves that seamlessly capture both global and local variations in dislocation geometry, as illustrated in Figure 2.5.

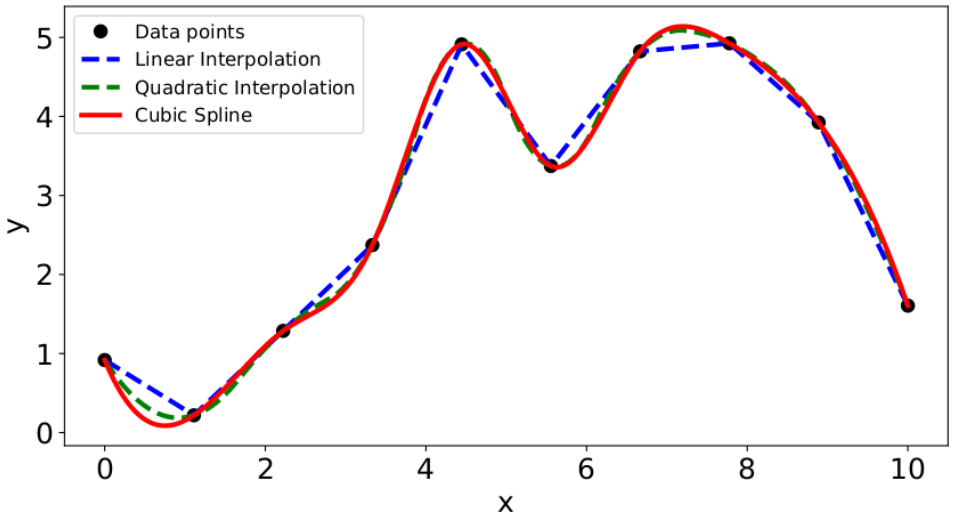


Figure 2.5: Comparison of linear, quadratic, and cubic spline interpolations applied to a set of randomly distributed data points. The cubic spline interpolation produces a smooth curve with continuous first and second derivatives, effectively capturing the intricate variations characteristic of dislocation geometries. In contrast, linear and quadratic interpolations exhibit less smoothness and fail to accurately model the complex curvature observed in dislocations. This illustrates the advantage of cubic splines in representing dislocation lines with high fidelity.

A spline curve is composed of piecewise polynomial segments joined at control points, ensuring continuity at each segment boundary. The degree of the polynomial segments governs the level of smoothness. For instance, a linear spline ensures only C^0 continuity (continuous function

value but not necessarily continuous slope), whereas a quadratic spline ensures C^1 continuity (continuous function and first derivative), producing smoother transitions as shown in 2.5 where we fit the two splines for some data points. Cubic splines are often preferred because they offer C^2 continuity, ensuring continuous function values, first derivatives, and second derivatives across segments. This high level of smoothness yields curves free from abrupt changes in slope or curvature, making cubic splines particularly suited for modeling the continuously varying curvature often observed in dislocations.

Mathematically, for a set of $n + 1$ control points $\{(x_i, y_i)\}$, the cubic spline is defined by n cubic polynomial segments of the form:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x \in [x_i, x_{i+1}]. \tag{2.10}$$

The coefficients a_i, b_i, c_i, and d_i are determined by enforcing continuity of the function and its first and second derivatives at the control points, along with specified boundary conditions. Typically, the second derivatives at the endpoints are set to zero, ensuring a natural cubic spline with minimal curvature at the boundaries. Additionally, known slopes at the endpoints can be incorporated to achieve desired boundary conditions. Although increasing the polynomial degree can increase flexibility, higher-degree splines risk introducing oscillatory behavior and instability, particularly when control points are unevenly spaced or numerous. Cubic splines provide an optimal balance between simplicity, computational efficiency, and flexibility, making them a robust choice for accurately modeling dislocation lines.

In this work, we model dislocations in a standardized local coordinate system to facilitate direct comparisons. Each dislocation is defined so that its first control point lies at $(x_{start}, 0)$ and its last control point at $(x_{end}, 1)$ in the local coordinates. By fixing these vertical references and normalizing the vertical extent to unity, any shape differences arise strictly from the horizontal and curvature variations rather than from scaling factors. Once the spline representing the dislocation is determined in this local frame, it can be transformed into the image coordinate system by applying suitable scaling and transformations. This approach ensures consistent, comparable representations of complex dislocation geometries, as shown in Figure 2.6, where the number and positions of control points vary to capture subtle or pronounced curvature changes.

The number of control points is determined by the complexity of the dislocation geometry. Simple geometries require fewer control points to accurately capture the overall shape, as the curvature changes are minimal or gradual. For instance, a straight or gently curved line can be represented with only a few control points, as illustrated for the dislocation in Figure 2.6a and Figure 2.6d. Complex dislocations with sharp bends, inflection points, or rapid curvature changes require more control points to accurately represent the shape as shown for Figure 2.6b and Figure 2.6c.

In this work, we manually extract support points along a dislocation curve using the software `labelme`. The process begins by importing a TEM image of the dislocation microstructure

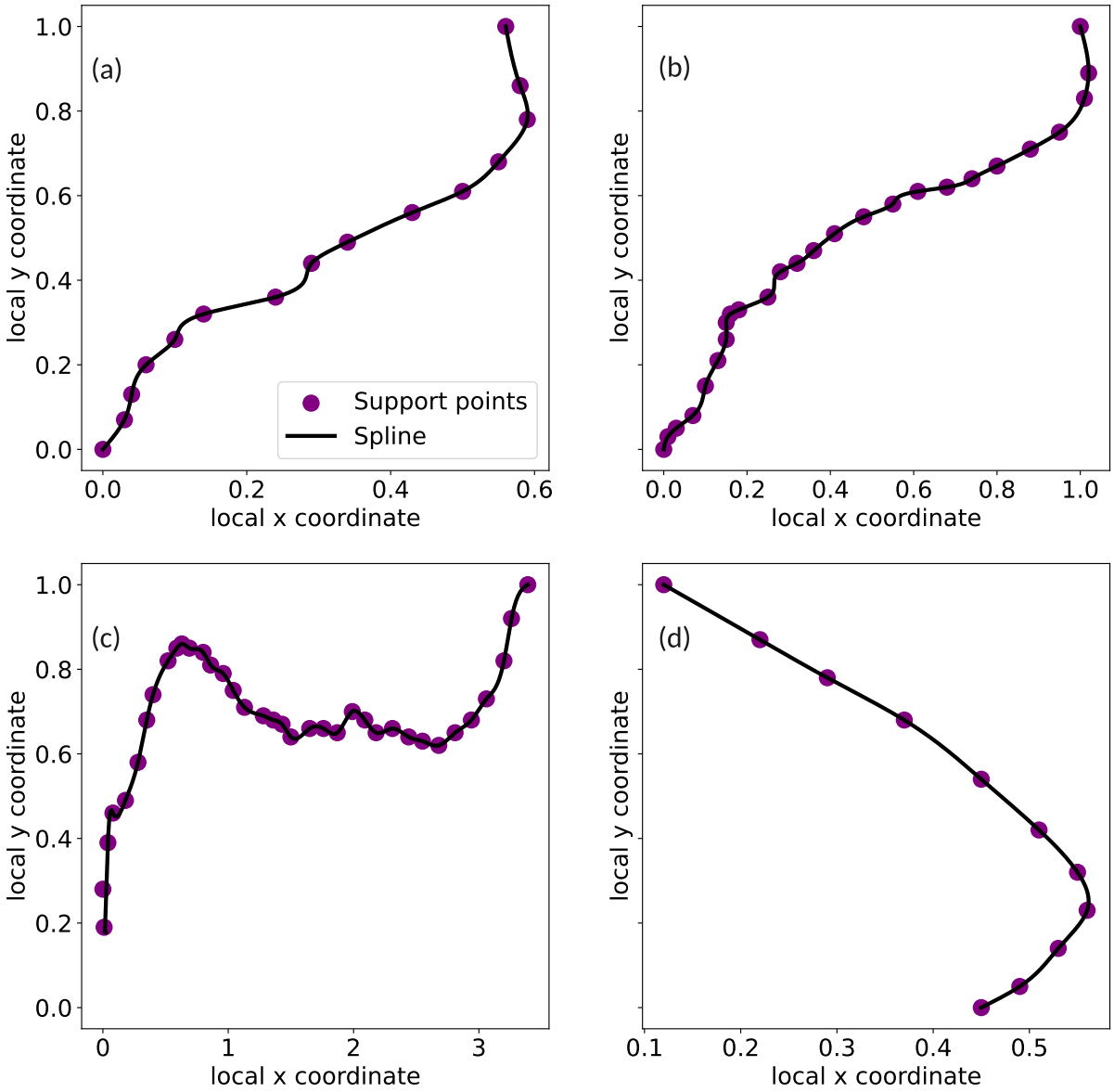


Figure 2.6: Visualization of support points and the corresponding cubic spline interpolations applied to dislocation geometries within a local coordinate system. The cubic spline interpolation ensures C^2 continuity, effectively capturing the intricate and continuously varying curvature characteristic of dislocations. This method provides a robust mathematical framework for modeling dislocation lines, accommodating both global and local geometric variations inherent in complex microstructures.

into `labelme`. A specific dislocation is selected, and is annotated as a polygon by manually placing points along the dislocation. Once the annotation is complete, a JSON file is generated,

containing the coordinates of the selected points in the image reference frame. These coordinates are then transformed into the standardized local coordinate system, providing the support points necessary for generating the synthetic microstructure.

2.2.2 Parametric Model for Dislocation Pileup

So far we have described how a dislocation can be modeled as a mathematical spline. In this work, we have assumed each dislocation being a part of a dislocation pileup. A single dislocation is a special case of a dislocation pileup consisting of just one dislocation. A real dislocation pileup in crystalline materials is characterized by a series of dislocations, that are located on the same slip plane and thus share the same Burgers vector. These dislocations are typically generated as a result of external stress, accumulating at an obstacle, such as a grain boundary, a precipitate. Apart from the direction along which dislocations are aligned in a pileup, the spacing between dislocations of the pileup is also an important characteristic of a pileup. The spacing between each dislocation in a realistic pileup is not arbitrary; it is governed by the balance of forces due to the stress field from external loading and the interactions between the dislocations themselves. These interactions are predominantly elastic, arising from the stress fields that each dislocation generates. The impact of these factors dictate the distribution of the dislocations in the pileup.

Very accurate and realistic modelling of dislocation pileup would require simulation methods based on dislocation mechanics where stress strain fields can be numerically solved for the dislocation shapes and dislocation distribution. In this work, our objective is to generate a representative dislocation microstructure for training machine learning models and hence it is not required to obtain the dislocation distribution and spacing from such methods which requires much detailed work from implementation to modelling of numerical equations.

Parameter	**Description**
N	Number of dislocations in the pileup.
Δd	Slip width (perpendicular distance between slip trace lines).
α	Orientation angle along which dislocations are aligned in the pileup.
p_0	Reference point on the slip trace line. Starting point for the first dislocation; defines initial position.
Dislocation spacing	Minimum pixel spacing between dislocations. Prescribed offset between adjacent dislocations based on distribution.

Table 2.1: Parameters for Modeling a Dislocation Pileup

The first step is to decide the values of the required parameters namely number of dislocations

to be part of the pileup, slip width Δd and orientation angle α of the direction along which dislocations are aligned in the pileup. The slip width is just the perpendicular distance between the two slip trace lines of the pileup which can be determined using the dislocation alignment direction. Along with this, we also need a reference point p_0 which is the point on of the slip trace line at which the first dislocation start. These parameters are shown in step 2 of Figure 2.1 and described in Table 2.1.

The steps to generate a dislocation pileups are as follows:

1. **Define Required Parameters**: Define the key parameters required for generating the dislocation pileup. The values of these parameters can be either obtained from a real dislocation pileup or from a distributions of the parameters which can be obtained from study of dislocation pileups of interests. Another approach would be to approximate the distribution using probability density functions which can be a good option in case distributions from real dislocation pileups is not available.

2. **Generate Dislocation Spline in Local Coordinate System**: Generate the dislocation spline based on the defined spline support points as described earlier. The spline represents the dislocation in a local coordinate system . This spline serves as a mathematical representation of the dislocation line, capturing its curvature and complexity.

3. **Transform to Image Coordinate System**: Convert the generated spline from the local coordinate system to the image coordinate system. First, multiply the spline points by the slip width (Δd). This step scales the spline to ensure that the dislocations are accurately scaled wrt. to the image coordinate system. Next, rotate the points using the orientation angle (α). This rotation aligns the dislocation spline with the defined pileup direction, ensuring that the dislocation reflect the desired orientation relative to the dislocation pileup. At this step we have a dislocation which is aligned with the slip direction of the pileup.

4. **Move First Point of Dislocation to Reference Point (p_0) of Pileup starting point**: Adjust the dislocation spline so that it coincides with the reference point of the first dislocation of pileup. This adjustment ensures that the first dislocation lies precisely where we want the dislocation pileup. At this step we have the first dislocation of the pileup which is correctly oriented and placed at the desired location in the dislocation pileup.

5. **Generate Additional Dislocations to Complete the Pileup**:Use the distribution of dislocation offsets to generate additional dislocations in the pileup. First, determine the dislocation offsets, which define the minimum spacing between consecutive dislocations. Generate subsequent dislocations by repeating the earlier steps for each new dislocation. Each new dislocation must be positioned along the dislocation direction while maintaining the prescribed pixel spacing to ensure a realistic arrangement of dislocations in the pileup.

To create more complex structures with multiple pileups, we can define more pileups. Each

pileup can have its own unique set of parameters. By carefully selecting these parameters, overlapping pileups can be generated, resulting in more complex and realistic configurations. The locations of the pileups can be strategically chosen to mimic realistic distributions observed in real microstructures. This flexibility allows for creating synthetic images that represent a wide variety of dislocation arrangements.

2.2.3 Parametric Model for Slip Trace Line

To include slip trace lines in the synthetic microstructure, we model each slip trace line as a straight line, which can be described by any two points lying on the line. Once the pileup has been generated, we can generate slip trace lines for the pileup. The first slip trace line is determined by connecting p_0, the starting point of the first dislocation, with the first point of the last dislocation in the pileup. The second slip trace line is formed by connecting the last point of the first dislocation with the last point of the last dislocation.

For both slip trace lines, we perform one-dimensional interpolation between the two defining points to compute equidistant points along each line. This process generates a set of line segments, which can be used during rendering. By assigning different grayscale values to each segment, we can mimic the appearance of realistic slip trace lines, capturing the subtle variations in contrast typically observed in real TEM images.

2.2.4 Parametric Model for Spots/Second Phase Particles

In this model, we aim to simulate the spots seen in the real TEM images of dislocation microstructure. These spots can be modeled as irregular polygons that are generated through a two-step process starting from a perfect circle.

The process begins by placing vertices equally spaced along the circumference of a circle with an average radius r_0. The total number of vertices is denoted by N, and each vertex is positioned at an angle θ_i, where $i = 1, 2, \ldots, N$. The coordinates of each vertex are given by:

$$(x_i, y_i) = (r_0 \cos(\theta_i), r_0 \sin(\theta_i)),$$

where the angle θ_i is defined as:

$$\theta_i = \frac{2\pi(i-1)}{N}.$$

This step generates a regular polygon with N vertices, all placed at an equal distance from the center of the circle as shown in step 1 of Figure 2.7.

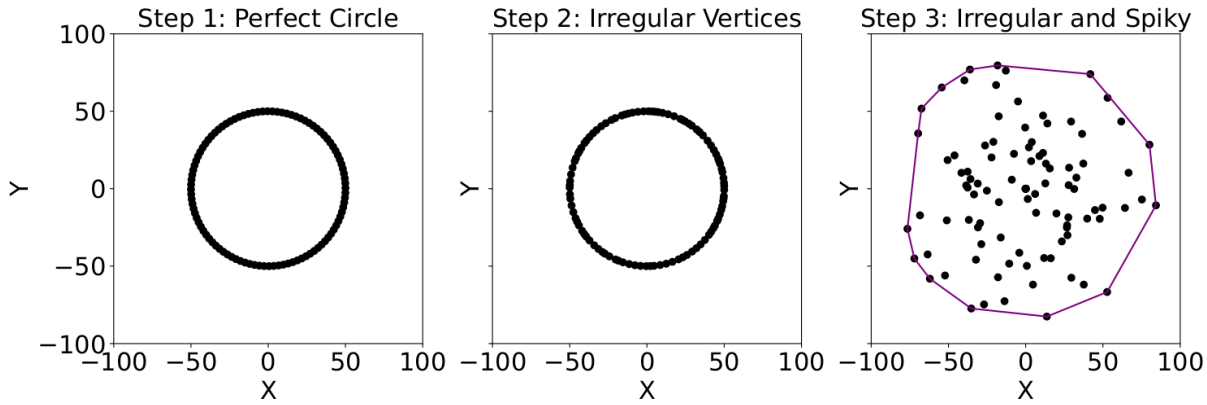


Figure 2.7: Generation of spots based on polygon. Step 1 shows a perfect circle with evenly spaced vertices (no irregularity or spikiness). Step 2 introduces irregularity, resulting in a polygon with vertices that deviate slightly from a perfect circle, while maintaining smooth edges. Step 3 adds spikiness, creating a more jagged and irregular shape with pronounced spikes along the perimeter.

Next, the angular spacing between consecutive vertices is altered by introducing irregularity. Each angle θ_i is perturbed by a random value $\Delta\theta_i$, where:

$$\theta_i' = \theta_i + \Delta\theta_i,$$

and $\Delta\theta_i$ is sampled from a uniform distribution:

$$\Delta\theta_i \sim \text{Uniform}(-\delta, \delta),$$

where δ is a parameter controlling the degree of irregularity. This step distorts the angular positions of the vertices, causing the spacing between them non-uniform, while still maintaining a roughly circular shape as shown in step 2 of Figure 2.7.

To further distort the shape, the radial distance of each vertex from the center is modified. Each vertex's distance is perturbed by a random value Δr_i, where:

$$r_i' = r_0 + \Delta r_i,$$

and Δr_i is drawn from a normal distribution:

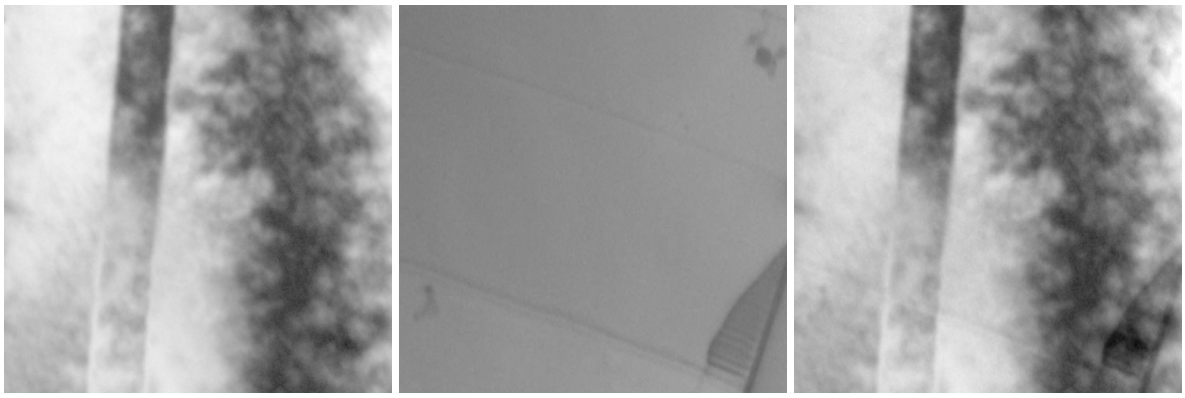
$$\Delta r_i \sim \text{Normal}(0, \sigma),$$

where σ is a parameter that controls the spikeyness of the shape. A higher spikeyness value results in larger variations in the radial distances, giving the spot a more jagged or spiked appearance as shown in step 3 of Figure 2.7.

Once the irregular vertices are generated, the convex hull of the set of points is computed. The convex hull is the smallest convex polygon that encloses all the points. This convex hull represents the outer boundary of the irregular spot, and it is the final spot shape that captures the outermost boundary of the distorted points as shown in step 3 of Figure 2.7.

2.3 Rendering of Dislocation Microstructure onto the Background

So far, we have described how to generate a background for synthetic image using the two methods and how to model different structures (i.e., dislocation pileup, slip trace line and spots).



(a) (b) (c)

Figure 2.8: Background patch images (a) and (b) are superimposed using opacity value of 0.5 to get background image (c) for synthetic image generation.

Let us understand what we have so far and how we can obtain the final synthetic image. Let us assume that we want to generate a synthetic image using background patches to generate the background for the image and a microstructure of a single pileup with three dislocations. The first step is to generate the background. We take the two images of the background patches as shown in Figure 2.8 and superimpose them using opacity value of 0.5. The resulting background image shown as Figure 2.8(c) still looks very realistic and will be used as background image for further steps.

The next step is the microstructure generation. To generate the dislocation pileup, we need to first obtain support points for the three dislocations of the pileup. The support points for the three dislocations used for this example are shown in Figure 2.9. We follow the steps described in Section 2.2.2 to generate the pileup by providing values of parameters (i.e., slip width, orientation

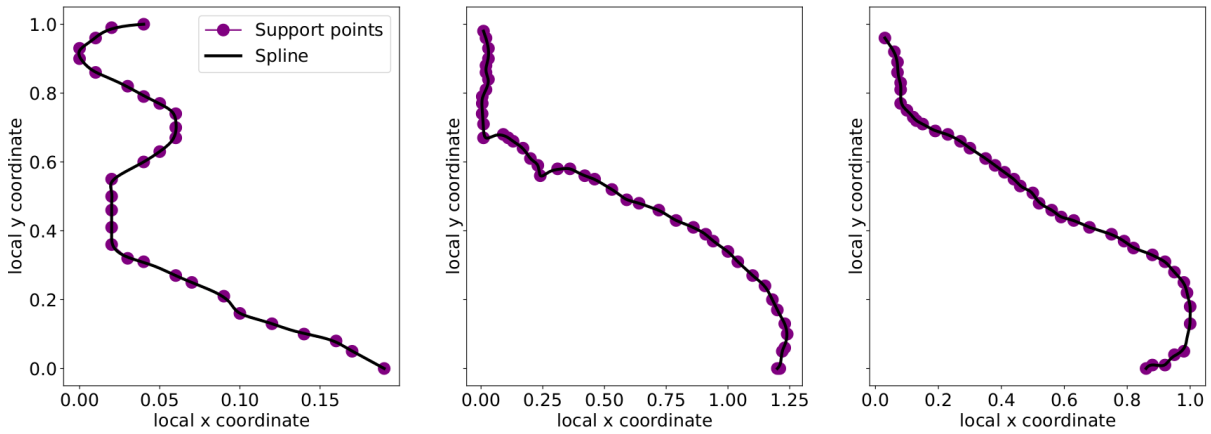


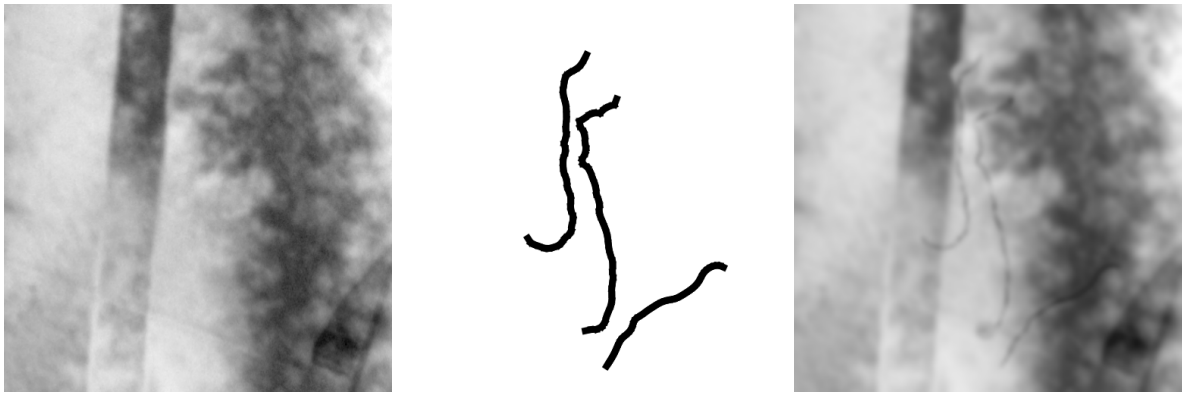
Figure 2.9: Support points and dislocation splines for the three dislocations used to generate pileup.

angle and dislocation spacings). This step provides us with $N = 100$ points on each dislocation of the pileup as shown in Figure 2.10b which shows scatter plot of points of the dislocations.

To synthesize a realistic bright-field micrograph (Figure 2.10c), the dislocation microstructure (Figure 2.10b) is rendered onto the background image (Figure 2.10a). In bright-field imaging, dislocations typically appear darker due to diffraction contrast. To mimic this effect, dislocations are rendered segment by segment, with appropriate intensity adjustments to mimic the darker appearance caused by diffraction contrast. This is done in the following manner.

1. For each line segment in the dislocation structure, the gray value at a pixel location (x, y), denoted as $\text{GrayValue}(x, y)$, is derived from the corresponding position on the background image. This gray value ranges between 0 and 1, where a value closer to 0 corresponds to a darker pixel, and a value closer to 1 represents a brighter pixel.

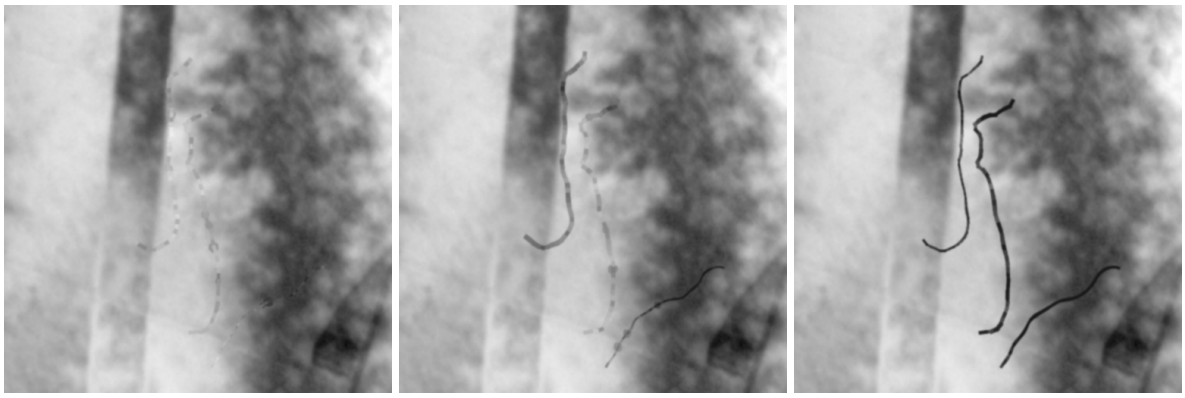
2. A user-defined parameter, referred to as the *gray value factor*, is introduced to darken the gray value of the dislocation for enhanced contrast. The adjusted gray value at the dislocation is calculated as: $\text{GrayValue}_{\text{dislocation}}(x, y) = \text{GrayValue}(x, y) - \text{gray value factor} \times \text{random.uniform}(0.75, 1.0)$, where $\text{random.uniform}(0.75, 1.0)$ introduces a degree of randomness, ensuring subtle local variations in contrast along each dislocation segment. These variations contribute to a more realistic appearance, as shown in Figure 2.11b. Lower values of the gray value factor produce lighter dislocation lines, while higher values yield darker and more prominent lines. When the gray value factor is close to 0, dislocations become almost indistinguishable from the background, as illustrated in Figure 2.11a. We start to see more darker dislocations as we use higher value. For very large value, the dislocation lines appear almost as black lines as shown in Figure 2.11c.



(a) (b) (c)

Figure 2.10: (a) Background image for synthetic image. (b) Dislocation microstructure of pileup of three dislocations. (c) Final Rendered synthetic image of microstructure onto the background.

3. To simulate the physical appearance of dislocations, a parameter controlling line thickness is applied. This parameter determines the width of each dislocation segment. To prevent an overly uniform appearance, minor random variations in thickness are introduced on a segment-by-segment basis, further enhancing realism.

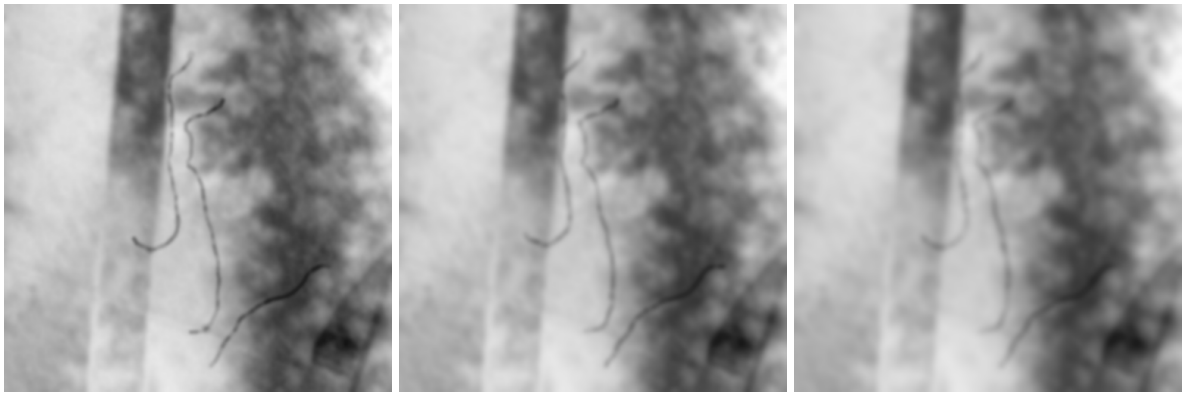


(a) (b) (c)

Figure 2.11: Rendering of dislocations using gray value factor ranging between (a) 0.0 and 0.1, (b) 0.1 and 0.4 and (c) 0.4 and 0.7. Increase in the value leads to darker dislocation lines.

To further enhance realism, a Gaussian filter is applied to this rendered image. While this step inevitably suppresses some local details—potentially leading to a slight loss of sharpness at the dislocation boundaries—it is instrumental in achieving more natural transitions between

the dislocations and their background. Three synthetic images are shown for different values of filter in Figure 2.12. Excessive smoothing with large values of filter (shown as Figure 2.12c) can, however, diminish the distinctive features that are crucial to mimic a real image. Thus, an appropriate filter size must be carefully chosen, guided by a balance between achieving realistic appearance and retaining sufficient dislocation detail for machine learning tasks.



(a) (b) (c)

Figure 2.12: Improving realism of rendered dislocations using Gaussian filter of size (a) 2.0 (b) 3.0 and (c) 4.0.

In addition to creating the synthetic images, this approach simultaneously generates precise ground truth masks for dislocations. Reproducibility and parameter management are ensured via the consistent documentation of all relevant parameters in a JSON file. This design choice is crucial for fostering transparent research practices and allowing future investigators to replicate or expand upon the generated datasets. Detailed records of parameter values—such as the ranges used for line thickness, the *gray value factor*, or the degree of Gaussian smoothing—enable exact replication of each synthetic image, as well as the capacity to systematically explore parameter spaces for sensitivity analyses.

Despite the level of realism achieved through parametric modeling and contrast manipulation, it is important to acknowledge the cases in which synthetic images may still appear recognizable as artificial. One such scenario is when the local pixel distribution on or near dislocations lacks the complex variation seen in real microstructures. Moreover, when the parameters, such as line thickness or contrast factor, are too narrowly sampled, the resulting images may fail to represent the full range of morphologies characteristic of real dislocations. These limitations highlight the challenges intrinsic to purely parametric methods in capturing high-fidelity realistic images.

Nonetheless, this synthetic data generation framework is highly adaptable and can be extended beyond dislocations in crystalline materials. The spline-based or polygonal representation can be transferred to represent other line-like or near-circular shapes, depending on the application.

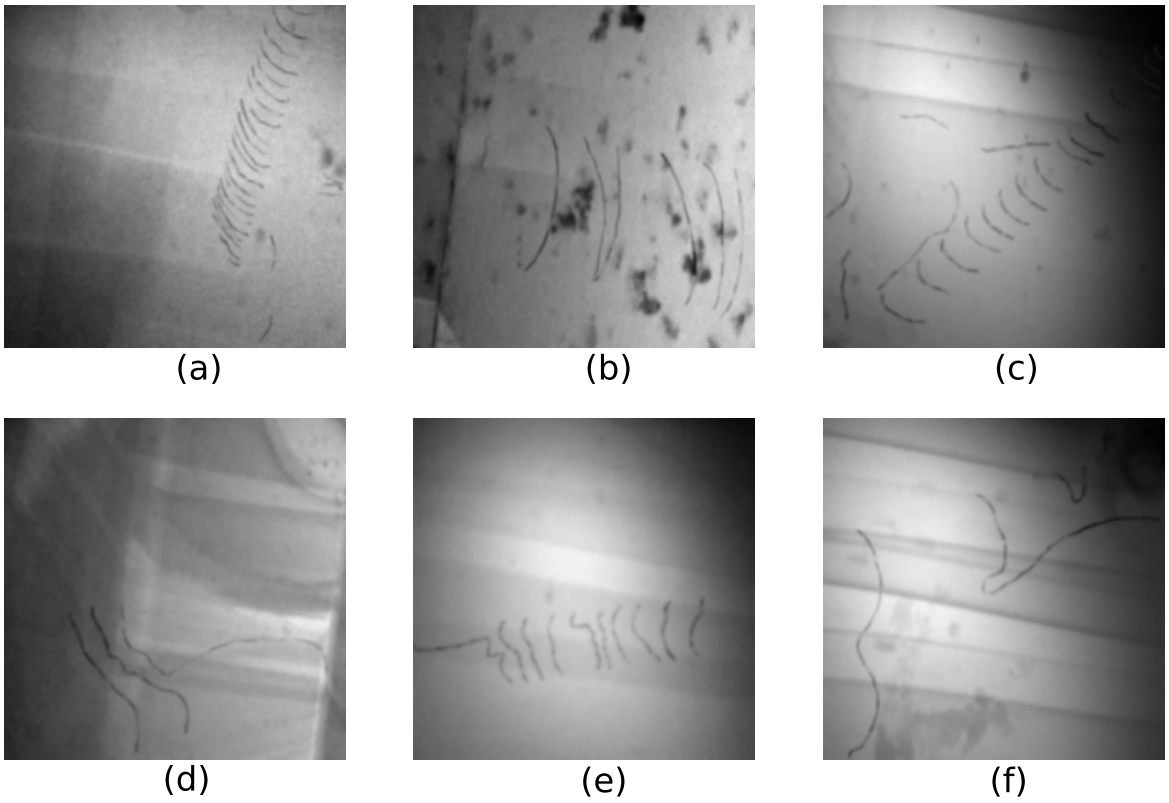


Figure 2.13: A wide range of synthetic images can be generated by varying parameter values of dislocation microstructure. Six sample images are shown with different dislocation shapes, sizes and orientations of pileup.

For instance, in the context of additive manufacturing, where bubble formation during the process can lead to defects observable via X-ray imaging, the same parametric method can be employed by using ellipses or splines to approximate bubble boundaries [96]. Similar principles apply in biological imaging, where worm-like structures can be faithfully synthesized via spline representations, providing annotated datasets for training segmentation or detection models [97, 98]. Thus, by tailoring the underlying parametric shapes, contrast rules, and randomness factors, the framework supports a variety of domains in need of synthetic training data.

The synthetic generation of dislocation microstructures presented here is driven by the need to create controllable, diverse, and high-fidelity training data for machine learning models. While inevitably introducing some nonphysical features, the deliberate perturbations and carefully tuned rendering parameters enhance dataset variability without compromising model performance. Examples of the versatility of this synthetic image generation approach are illustrated in Figure 2.13, which shows the diverse range of dislocation microstructures we can generate by carefully selecting parameter values.

Chapter 3 Machine Learning Approaches

The purpose of using machine learning in this work is to process a TEM image containing a dislocation microstructure and automatically extract the dislocations as curvilinear features (for example, represented as splines), as illustrated in Figure 3.1. This can either be done directly, where a machine learning model is designed in such a way that it directly predicts the support points of the splines or indirectly, where the machine learning model predicts the dislocation masks and then the masks are post-processed to obtain the dislocations as splines.

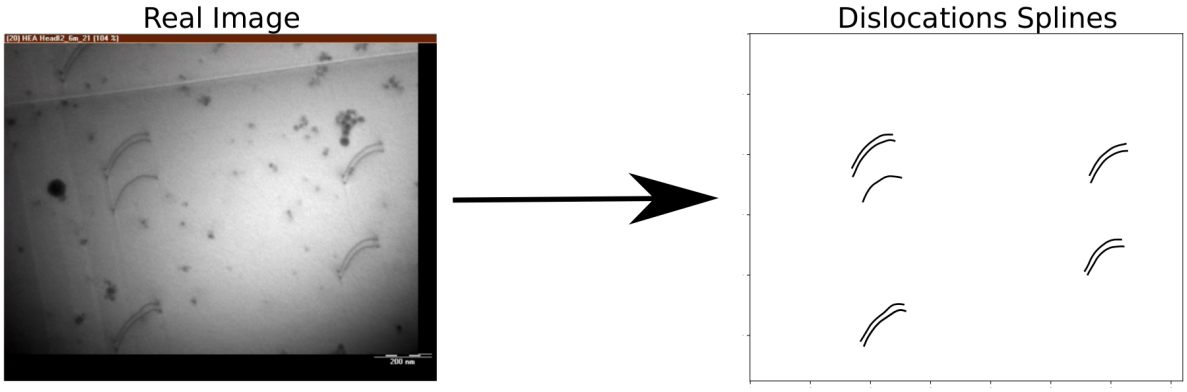


Figure 3.1: Illustration of digital representation of dislocations in a TEM image as spline curves. The process involves either direct prediction of spline support points by a machine learning model or indirect prediction of dislocation masks followed by post-processing to extract spline representations. This approach facilitates accurate modeling of dislocation geometries for subsequent data-driven analysis.

In this work, we focus on three different machine learning based approaches as described below. We will discuss one of the most versatile and popular approach, instance segmentation, in Section 3.1 based on Mask Region-Based Convolutional Neural Network (Mask R-CNN) model. We also provide details of the architecture of the model and some of its most important features. This is followed by a slight variation where we use Mask R-CNN but try to predict the spline support points directly for each dislocation. This does not require any post-processing of a mask to obtain the splines. The last approach discussed in details is multi-label segmentation where we predict a single dislocation per mask.

3.1 Instance Segmentation Approach: Mask R-CNN model

Instance segmentation is an important method in the field of computer vision where the object detection task is performed by predicting the bounding boxes around objects of interest as well as segment instances of each object. A real image of dislocation microstructure consisting of a pileup of three dislocations is shown in the left image of Figure 3.2. We obtain bounding boxes around each dislocation (middle image of Figure 3.2) along with the segmentation masks using the instance segmentation approach. We can then post-process the masks to represent each dislocation as spline as shown in right image of Figure 3.2. Unlike semantic segmentation, which classifies objects of the same type into a single category, instance segmentation identifies each instance of a category separately [99]. This means it can distinguish between two dislocations as distinct objects as shown in the example. This aspect is particularly useful in distinguishing between dislocations that are very close or even overlapping. This is one of the major challenge working with dislocation image data, which can be solved by using instance segmentation approach.

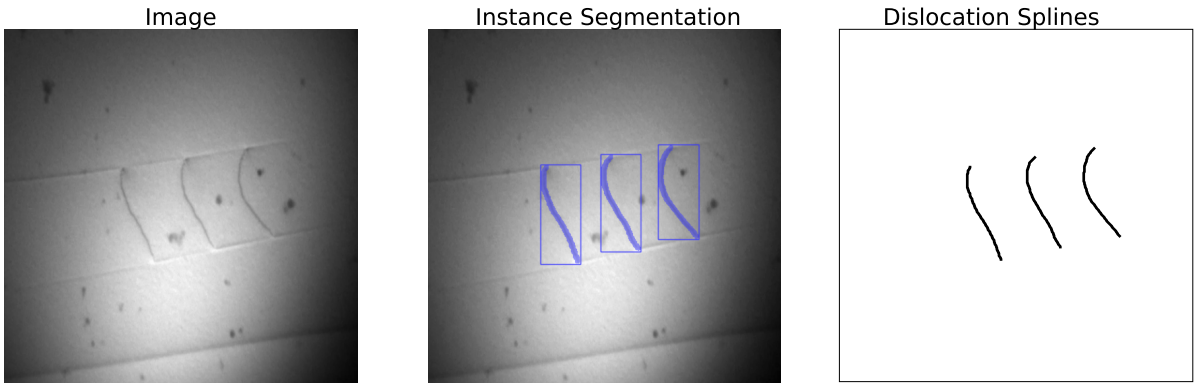


Figure 3.2: The left image shows the original image, which contains a pileup of three dislocations. The middle image highlights the output of the instance segmentation model, displaying bounding boxes and segmented masks for each dislocation. The right image demonstrates the post-processing step, where the segmented masks are converted into splines for precise representation of individual dislocations.

The R-CNN framework, initially introduced by Girshick et al. [100], employs a two-stage approach for object detection. In the first stage, it generates a set of candidate object regions (referred to as region proposals) within an image. In earlier iterations of the model, the production of these proposals relied heavily on a selective search algorithm. The selective search technique, as proposed by Felzenszwalb and Huttenlocher [101], initially over-segments the input image into a large number of small, superpixel-like regions. These regions are subsequently merged hierarchically based on carefully defined similarity metrics, including shape, size, color, and texture, ultimately yielding a set of roughly 2,000 candidate proposals per image. At each

iteration of the merging process, two adjacent and similar regions are combined into a larger region, thereby reducing the total number of proposals while progressively increasing the spatial scale of the candidate regions. Although selective search represents a more efficient alternative to exhaustive search methods, it remains computationally expensive, particularly for high-resolution images. Furthermore, because it operates as a pre-processing step external to the learning process, selective search cannot be trained end-to-end, thus introducing inefficiencies and complications into the overall training pipeline.

Once the region proposals have been obtained, the second stage of the R-CNN framework involves extracting discriminative features from each proposed region. To achieve this, each region proposal is first warped or resized to a fixed dimension, ensuring compatibility with standard CNN architectures. A pre-trained CNN, often AlexNet or VGG16, is then applied to each region, producing a 4096-dimensional feature vector. This feature extraction step, repeated for every proposal, significantly contributes to the computational overhead of the system. While parallelization strategies can partially mitigate this cost, the repeated processing of a large number of proposals remains a notable bottleneck.

Following feature extraction, the high-level CNN features corresponding to each proposal are passed through a set of class-specific linear support vector machines (SVMs) to determine the presence or absence of particular object classes. To refine the localization of objects within these proposals, a separate bounding box regression model is used. The regression model adjusts the spatial coordinates of the initially proposed bounding boxes to more accurately align with the boundaries of detected objects. This combination of feature extraction, classification, and bounding box refinement proved effective, leading to a substantial improvement in detection performance. The original R-CNN work, for instance, achieved an approximate 30% increase in mean average precision (mAP[1]) compared to previous state-of-the-art methods.

Despite these advancements, the R-CNN approach suffered from several limitations that hinder its practical utility. Notably, the reliance on a separate, non-trainable selective search process complicates training and prevents an entirely end-to-end learnable system. Moreover, performing computationally expensive CNN-based feature extraction for thousands of region proposals results in high latency and limits the suitability of the R-CNN model for real-time applications. As the field of object detection advanced, subsequent methods sought to address these issues by integrating region proposal generation into the network itself and streamlining the feature extraction process, paving the way for more efficient, end-to-end trainable models.

The introduction of Region Proposal Network (RPN) by Ren et al. [102] in Faster R-CNN model marked a new era in instance segmentation. This advancement enabled end-to-end training, significantly simplifying the training process and reducing model complexity. The RPN has

[1]Mean Average Precision is a standard evaluation metric for object detection tasks, calculated as the mean of the average precision (AP) scores across all object classes. It evaluates both precision and recall to provide a comprehensive measure of detection accuracy.

proven highly effective for object detection tasks. Convolutional layers, through kernels of various sizes, excel at capturing image features, learning essential patterns in the data [103, 104]. Faster R-CNN improved object detection by incorporating the RPN to generate region proposals, a significant advancement over R-CNN, which applied CNNs individually to thousands of region proposals obtained via selective search. Faster R-CNN, in contrast, passes the entire image through the CNN only once to create a feature map, from which it extracts features for each region, greatly enhancing training and inference speeds. In this unified model, all tasks—region proposal generation, feature map creation, object detection, and bounding box regression—are streamlined into a single framework, further improving performance by using a multi-task loss for classification and bounding box regression.

Faster R-CNN was later extended to include high-quality instance segmentation with the development of Mask R-CNN [55]. Mask R-CNN introduced a flexible design that allows additional branches for parallel tasks, enabling instance segmentation along with other tasks, such as keypoint detection. Since its introduction, newer and faster models have been developed, like YOLO [105], designed for speed and accuracy, with some models specifically optimized for real-time detection in practical computer vision applications [106]. These developments have greatly expanded the scope and application of deep learning-based models in real-world detection and segmentation tasks.

In this work, we use the Mask R-CNN deep learning model for instance segmentation. The architecture, shown in Figure 3.3, consists of three key components: a backbone network for feature extraction, a RPN for generating object candidate regions, and a Region of Interest (RoI) alignment layer. The model then branches into two paths: one for bounding box detection and classification, and the other for mask generation and keypoint detection. Each of these components are discussed in details in the following subsections.

3.1.1 Backbone of Mask R-CNN Model

The process begins by passing an input image through a pre-trained convolutional neural network, known as the backbone network. The backbone is responsible for feature extraction, where it transforms the input image into a set of feature maps that capture different levels of abstraction from the image. Several backbone architectures can be employed, including ResNet [107] and VGG19 [108], each offering distinct advantages in terms of feature representation and computational complexity. ResNet50, a deep convolutional neural network with 50 layers, is used in this work as the backbone architecture due to its efficient balance between model depth and computational demands. By incorporating residual connections, ResNet50 addresses the challenge of vanishing gradients, facilitating the training of deeper networks and enhancing gradient propagation. This architecture effectively extracts hierarchical features, progressing from low-level features such as edges and textures in the initial layers to high-level, abstract representations in the deeper

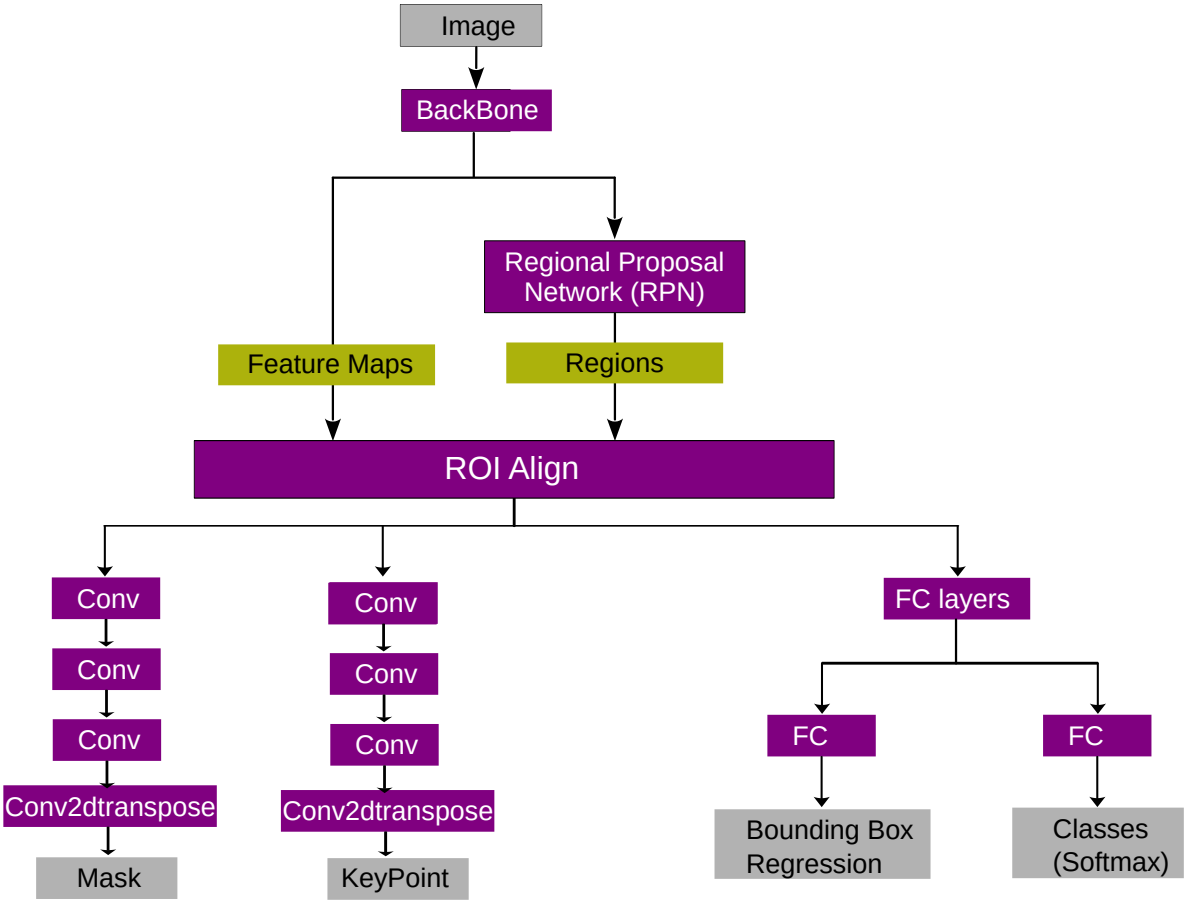


Figure 3.3: Schematic representation of Mask R-CNN architecture. The model is composed of three primary components: (1) a backbone network for extracting high-level features from input images, (2) a Region Proposal Network (RPN) for identifying potential object regions, and (3) an ROI Align layer for precise spatial alignment of the proposed regions. Following the ROI Align step, the architecture branches into two parallel paths: one dedicated to object detection and classification, producing bounding boxes and class labels, and the other dedicated to instance segmentation, generating pixel-level masks for each detected object.

layers. Although training the model on a dataset specific to dislocation images would yield more tailored feature maps, pretrained ResNet50 provides valuable insights into the learned image representations.

To further enhance the model's capability to handle multi-scale features, a Feature Pyramid Network(FPN) first proposed by Lin et al. [109], was incorporated into the architecture alongside the ResNet50 backbone. FPN is specifically designed to capture and represent features at multiple

scales by constructing a pyramid of features from a single input image, which is particularly beneficial for complex images with hierarchical structures. This pyramid includes feature maps with varying spatial resolutions, spanning from high-resolution features with precise spatial details to low-resolution features rich in semantic information. The high-resolution layers, derived from the early stages of the network, capture fine-grained spatial details like edges and small patterns, which are crucial for distinguishing textures and small-scale patterns. In contrast, low-resolution layers from the deeper network stages focus on high-level semantic content, representing more abstract structures.

FPN achieves this hierarchical representation by building a top-down pathway that fuses information from both high and low-resolution feature maps. It utilizes lateral connections that combine feature maps from each stage of ResNet50 with upsampled maps from higher layers, creating multi-scale features that integrate both fine details and contextual understanding at each level. This combination enhances the model's ability to detect objects and structural patterns across scales.

3.1.2 Region Proposal Network (RPN)

The quality and richness of feature maps generated by the backbone network play a crucial role in the performance of subsequent stages in object detection models, particularly in the RPN and RoI Align layer. The feature maps produced by the backbone network serve as inputs to the RPN, which generates object proposals that are refined in later stages. For the RPN to function effectively, these feature maps must be of high quality. This quality is enhanced by using feature maps from multiple layers of the backbone network instead of relying solely on the final feature layer. The RPN leverages features from various layers, enhancing its ability to detect objects of different sizes and scales. This multi-layer approach allows the RPN to capture both fine details and broader contextual information, improving the accuracy of the generated proposals.

The RPN is designed to propose candidate object bounding boxes (regions) from the input feature maps. Its operation can be broken down into several steps. At the core of the RPN are anchor boxes, which are predefined bounding boxes that serve as reference points for identifying potential object regions within the image. The RPN generates anchor boxes for each point on the feature map produced by the backbone. Anchor boxes are generated based on specific scales and aspect ratios: scales of 128, 256, and 512, and aspect ratios of 1:2, 1:1, and 2:1. The feature map is a downscaled version of the original image (e.g., reduced by a factor of 16 in width and height when using VGG16), meaning each point on the feature map corresponds to a larger region in the original image. To ensure that the anchor boxes generated on the feature map align correctly with regions in the original image, their dimensions and positions must be adjusted by the same scaling factor. This adjustment guarantees that the anchor boxes accurately represent their intended areas in the original image. This is accomplished by using the downsampling

factor as a stride when positioning anchor boxes across the image. The placement of anchor boxes ensures they correspond accurately to the scale of objects in the original image. For a feature map with a spatial resolution that is 1/16th of the original image, anchor boxes must be appropriately positioned to align with the true object locations in the original image.

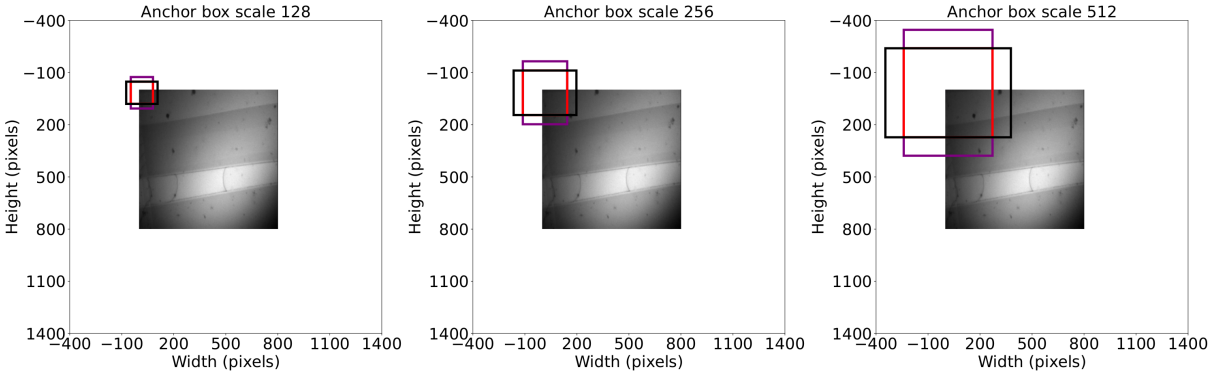


Figure 3.4: Illustration of anchor box generation for the initial point of a 25×25 feature map derived from an 800×800 input image. The anchor boxes are centered at position (16,16) and are categorized into three aspect ratios—1:2, 1:1, and 2:1—represented by purple, red, and black colors, respectively. This configuration results in nine anchor boxes per feature map point, facilitating the identification of potential object regions within the image.

To illustrate the concept of anchor boxes, let us consider an example where we have feature maps of size 25×25 for an input image of size 800×800 as shown in Figure 3.4. For each position on the 25×25 feature map, we need to generate anchor boxes. For the 3 different scales and 3 different aspect ratios, we will generate a total of $3 \times 3 = 9$ anchor boxes for each position of the feature map. The total number of positions on the feature map is 625. Since we generate 9 anchor boxes for each position on the feature map, the total number of anchor boxes generated will be $625 \times 9 = 5625$.

The coordinates for the first 9 anchor boxes, centered around the top-left corner of the feature map, will vary based on the calculated dimensions but will be anchored at the center of the feature map's first position. Each anchor box is defined by its width and height, calculated based on the chosen scales and aspect ratios shown in the figure. To accurately place these anchor boxes in relation to the original input image size, we need to consider the down-sampling that occurs when generating the feature map. For the feature map, each position in the feature map corresponds to a 32×32 pixel region in the original image, since the image is down-sampled by a factor of 32. Therefore, the anchor boxes centered at the first feature map position (1, 1) will be positioned at the coordinates corresponding to the top-left 32×32 pixel block of the original image as shown in the figure which will have center at 16, 16. Consequently, each of the 9 anchor boxes will extend outward from this center point according to their calculated dimensions. The

precise placement ensures that these anchor boxes are positioned correctly to potentially overlap with any objects present in the original image, thereby improving the likelihood of accurate object detection in subsequent processing steps.

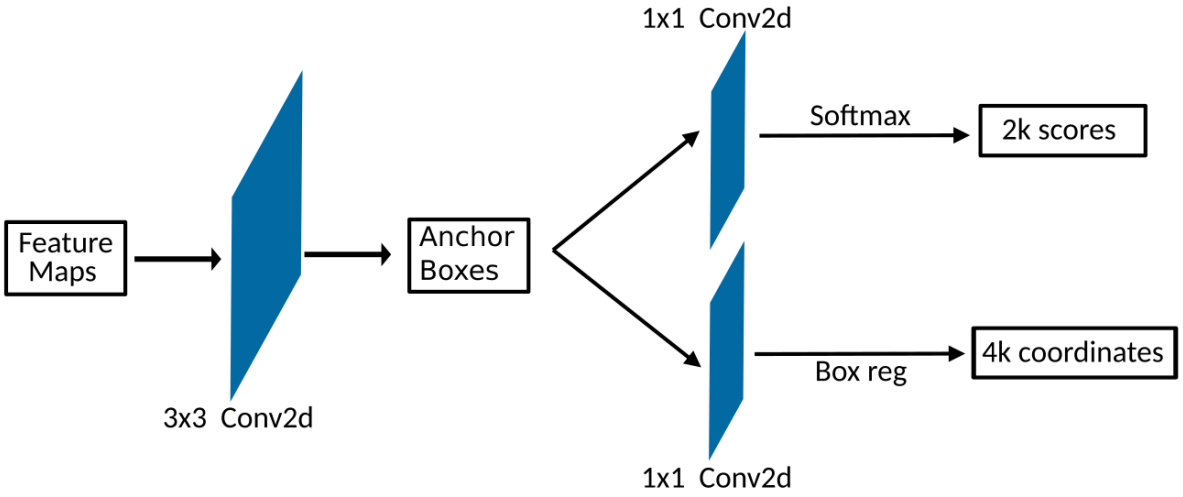


Figure 3.5: Schematic representation of a Region Proposal Network architecture. The RPN generates object proposals by sliding a small network over the convolutional feature map produced by the backbone network. For each sliding window, it predicts multiple region proposals, each with an objectness score and bounding box coordinates. These proposals serve as candidate regions for further processing in the subsequent stages of the Mask R-CNN pipeline.

At this stage, the RPN generates a large number of potential anchor boxes that could contain objects. However, two critical issues need to be addressed. First, we must classify each anchor box to determine whether it contains an object. This classification is essential for distinguishing between foreground objects and background noise. Second, even when an anchor box is found to contain an object, it may not perfectly fit the object due to variations in shape and size. Therefore, adjustments to the anchor boxes are necessary to ensure that the entire object is accurately encompassed. The RPN tackles these challenges by making two key predictions for each anchor box, as illustrated in Figure 3.5 where we have a classifier and regressor.

The classifier predicts the objectness score, which serves as probability for the binary classification indicating whether the anchor box contains an object or not. This score helps in effectively distinguishing between foreground objects and background areas. The regressor provide refinements to the anchor boxes in the form of offsets relative to their original positions and sizes. These offsets allow the network to adjust the anchor boxes to more accurately fit the detected objects. During training, the RPN incorporates both predictions into its loss function,

balancing the contributions from the classification accuracy and the bounding box precision. This balance is crucial for the effective performance of the RPN, as it ensures that the model not only identifies regions with potential objects but also identify their boundaries with high precision. The RPN outputs K proposals, each defined by four values that specify the bounding box coordinates and two scores for the objectness score.

3.1.3 Generating Proposal Regions from RPN

After the RPN generates K proposals along with their associated objectness scores and corrections for the boxes, several important post-processing steps are taken to refine these proposals for use in subsequent stages of the pipeline. The first step involves selecting the top 200 anchor boxes based on their objectness scores. This selection process helps to reduce the number of proposals to a manageable amount while retaining those that are most likely to correspond to objects in the image. The choice of retaining 200 proposals is a hyper-parameter that can be adjusted based on the specific requirements of the application, ensuring a balance between computational efficiency and detection performance.

Following the selection of the top proposals, we apply a threshold based on the minimum area of the anchor boxes to remove any proposals that are too small to be meaningful. This step is crucial, as very small anchor boxes may not correspond to any significant object, leading to noise in the final output. By discarding proposals that do not meet the minimum area requirement, the model focuses on regions that are more likely to contain relevant objects, thereby improving the quality of the proposals that will be considered for further processing.

The final refinement of the proposals is accomplished through the use of Non-Maximum Suppression (NMS), which filters out redundant proposals that may overlap significantly. NMS uses the Intersection over Union (IoU) [2] metric to evaluate the overlap between anchor boxes, with a predefined IoU threshold—commonly set to 0.5 — used to determine whether to retain or discard an overlapping box. The NMS works as follows:

1. **Sort by Objectness Score:** All candidate proposals are sorted in descending order based on their objectness scores.
2. **Select the Top Box:** The bounding box with the highest objectness score is selected first.
3. **Calculate IoU:** The IoU between the selected box and all other boxes is calculated.

[2] Intersection over Union (IoU) is a metric used to evaluate the overlap between two bounding boxes, typically in the context of object detection. It is calculated by dividing the area of overlap between two boxes by the area of their union. The resulting value ranges from 0 (no overlap) to 1 (perfect overlap). In object detection, a higher IoU indicates a better match between the predicted and ground truth bounding boxes. For details, please refer Section 3.4.1.

4. **Remove Overlapping Boxes:** Any boxes with an IoU greater than the predefined threshold with the selected box are discarded.

5. **Repeat:** The process is repeated with the next highest-scoring box, iteratively removing any boxes that overlap too much with the selected boxes.

By filtering out redundant boxes, NMS ensures that only the most accurate proposals remain, improving the efficiency and accuracy of object detection models.

3.1.4 Region of Interest Alignment (RoI Align)

After the proposals are generated by RPN we use RoI Align. RoI Align addresses a key challenge in object detection models: extracting a fixed-size feature vector from an arbitrary-sized region proposal. This is crucial for applying a classifier and bounding box repressor, which requires inputs of a consistent size. The primary function of RoI Align is to accurately extract feature maps from each proposed region, ensuring that the spatial information is precisely preserved which is required to improve the accuracy.

To understand it better, let us consider the previous example where we have a feature map of size 25×25 for the image of size 800×800. Let us assume that one of the proposed region has size 156×220. When we try to map it onto the feature map, the proposed region is down scaled by a factor of 32 and we get 4.875×6.875 which does not align with the grid of the feature map. The spatial miss alignments can adversely affect the model's performance, particularly in tasks requiring precise localization, such as instance segmentation. RoI Align works by dividing each proposal region into a fixed number of bins (e.g., a 7x7 grid). For each bin, it calculates the exact values of the input features at four regularly sampled locations within the bin. At each of these four locations, RoI Align uses bilinear interpolation to compute the exact value of the input feature map. Bilinear interpolation considers the closest four feature points in the feature map and computes the value based on their weighted average. This method preserves the fractional part of the coordinates, ensuring that the spatial information is not lost. After computing the interpolated values, RoI Align aggregates them (usually by averaging) to produce a single output value for each bin. The results from all bins form a fixed-size feature map for each RoI. At the end of the RoI alignment, we will have a fixed size feature map for each of the proposals obtained from RPN having same number of features as discussed above.

3.1.5 Classification and Regression Heads

The RoI Align provides fixed size features for each proposal region which can now be used to predict the object category of the region and the corresponding bounding box for each category.

It is important to note that the object detection pipeline performs classification twice, first within the RPN and subsequently in the final detection head. In the RPN, classification is used to assign an objectness score to each anchor box, indicating whether it likely contains an object or not. This initial classification step enables the RPN to select the most promising candidate proposals, which are then passed on to the next stage of the network. These proposals, however, do not yet have specific object category labels assigned to them; they are only filtered and refined based on objectness scores and bounding box adjustments.

The fixed-size feature maps generated by RoI Align are fed into a classification and bounding box regression network, referred to as the detection head. In this final detection head, the model performs a second classification, this time to predict the specific object category for each proposal. The detection head uses a series of fully connected layers that map the feature vectors to discrete object classes, similar to the approach used in standard image classification tasks. This allows the model to assign a specific category label to each proposal region.

In addition to classification, the detection head also performs a second bounding box regression, providing refined bounding box coordinates tailored to the predicted object category. This final bounding box adjustment helps to more accurately enclose the detected object within the proposal region, further improving localization precision. By performing both classification and bounding box regression in the detection head, the model is able to output highly accurate object labels and precise bounding box coordinates for each detected object, leveraging both the initial proposals from the RPN and the fixed-size feature maps from RoI Align. This dual-stage classification and refinement process enhances the overall accuracy and robustness of the object detection pipeline.

3.1.6 Mask Detection Head

Once the bounding box detection task has been completed and the object category and bounding box coordinates have been predicted for each proposal, we can proceed with mask generation. To begin, the outputs of the bounding-box detection step are analyzed, and only the proposals that contain objects (based on a predefined probability threshold) are retained. After filtering these proposals, we perform another RoI alignment. However, unlike the previous RoI alignment used in bounding box detection, this alignment operates on the proposals refined by the detection head, rather than those initially generated by the RPN. Additionally, for this alignment task, a larger grid size (e.g., a 56×56 grid) is used, allowing for a higher-resolution feature map that is well-suited for precise mask generation.

This dual application of RoI Align is crucial in enabling Mask R-CNN to perform both object detection and instance segmentation with high accuracy. After obtaining the larger, aligned feature maps for each proposal, a small CNN composed of several convolutional layers and a transposed convolutional layer is applied to generate the segmentation mask. This network

outputs a binary segmentation mask for each object class, indicating the precise pixel-level locations of the detected object within each proposal region.

It is important to note that the predicted mask is produced at a lower resolution, typically smaller than the original input image. To achieve a final mask with the same dimensions as the input, a post-processing step is applied to upsample the mask. This upscaling operation ensures that the mask aligns accurately with the original image dimensions, producing a high-resolution, instance-specific mask for each detected object. This mask generation process, combined with bounding box detection, forms the core of Mask R-CNN's effectiveness in both object detection and instance segmentation tasks.

3.2 Spline Support Point Detection: Keypoint R-CNN

We have discussed above how Mask R-CNN can be used to perform the instance segmentation task. The model can be modified with relative ease to also perform keypoint detection, resulting in the modified model known as Keypoint R-CNN. The problem of representing each dislocation as a spline can also be modeled as a keypoint detection task, where the goal is to predict the support points of dislocation splines. This formulation allows us to directly use the model's output to construct the spline without requiring additional post-processing of the results, as is necessary for Mask R-CNN. Phrasing the problem as support point detection offers the advantage of simplicity and efficiency in obtaining the desired spline structure. This approach has been discussed in details by [36]. In this work, we aim to predict 20 support points for each dislocation, based on the average number of points required to annotate a dislocation using the labeling software *labelme*. This choice is made to strike a balance between accuracy and efficiency. Predicting a dislocation with too few points could result in an overly simplified representation, while using an excessive number of points might lead to inaccurate predictions of the support points. Predicting large number of keypoints requires the model to maintain high accuracy in spatial localization, which becomes more challenging as the number of keypoints increases. By selecting 20 points, we ensure that the model can effectively capture the essential details of the dislocation's structure without compromising accuracy.

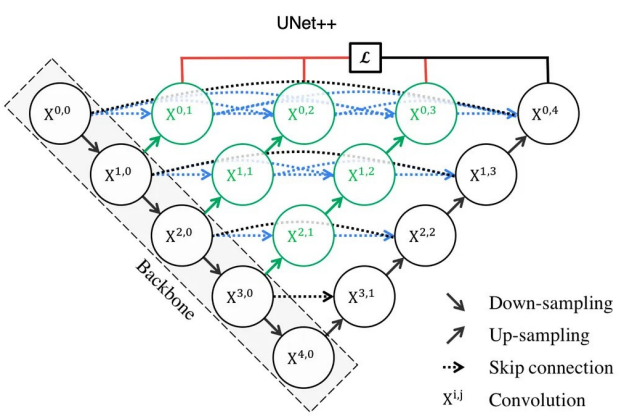
The keypoint detection head in Keypoint R-CNN is structured as a specialized network head that operates on the feature maps obtained from the RoI Align layer, much like the mask head. However, instead of producing binary masks or category labels, this head is designed to output precise coordinates for each keypoint associated with an object. The head typically consists of a series of convolutional layers followed by an output layer with a size corresponding to the number of keypoints per object.

Keypoint detection in Keypoint R-CNN includes a specific loss function, often a mean squared

error or L2 loss , applied at each keypoint location to encourage accurate prediction of each support point. This head outputs a heatmap for each keypoint, which is then converted to precise coordinates corresponding to the support points on the dislocation splines. The model's flexibility to predict a fixed set of support points ensures a consistent and structured output for spline representation, enabling the straightforward reconstruction of each dislocation's spline from the detected keypoints.

3.3 Multi-label Dislocation Segmentation: U-Net++

Segmenting individual dislocations is especially challenging when multiple dislocations are close to one another or even overlap. To address this, we generate a separate mask for each dislocation, ensuring that all pixels within a mask belong to the same object. Our multi-label dislocation segmentation approach uses U-Net++ architecture proposed by Iakubovskii [110] using a ResNet50 backbone, implemented in PyTorch.



Dense Skip connections (blue color): Inspired by DensetNet to improve semantic accuracy and gradient flow. All previous feature maps arrive at same node.

Redesigned skip pathway (green color): Bridge semantic gap between encoder and decoder pathways.

Deep supervision (red color): For accurate mode, the output from all the 4 segmentation branch is averaged.
For fast mode, the final segmentation map is selected from one of the 4 segmentation branches.

Figure 3.6: Overview of the U-Net++ architecture, highlighting its nested skip pathways and deep supervision. Multiple intermediate nodes at each level refine the feature maps before merging with upsampled decoder outputs, enhancing segmentation accuracy for closely spaced or overlapping objects such as dislocations.

U-Net++ builds upon the standard U-Net [44] by introducing three main enhancements: nested skip connections, deep supervision, and refinement of the skip connections as shown in 3.6. The

[2]L2 loss, also known as the mean squared error loss, is a common loss function used to measure the squared differences between predicted and actual values. It is calculated as the average of the squared differences between the predicted and true coordinates of each keypoint. This helps penalize large deviations from the true keypoint locations, promoting accurate predictions.

nested skip connections insert additional convolutional blocks along the pathways that connect the encoder and decoder, allowing a more gradual transition in feature space. Unlike the single skip connection in U-Net, U-Net++ creates multiple nodes at each level, each of which is refined through convolutional layers before being merged with upsampled decoder features. The deep supervision mechanism further improves training by incorporating partial segmentation outputs from multiple layers in the decoder into the loss function, thereby enhancing gradient flow and promoting multi-scale feature learning. This approach compels the model to generate consistent segmentation results at several stages of the decoder, ultimately improving robustness.

By refining the skip connections using convolutional operations before merging with decoder outputs, U-Net++ reduces the semantic gap between low-level features (e.g., edges and textures) and high-level semantic cues (e.g., object shapes). This hierarchical approach helps capture small boundary details while retaining global context. In the context of dislocation segmentation, such multi-scale feature blending enables the model to distinguish closely spaced or partially overlapping dislocations, ensuring each dislocation is accurately segmented as a separate entity.

Our experiments suggest that U-Net++ also offers flexibility in terms of computational requirements. For instance, the network can be pruned to remove certain intermediate nodes, thus lowering computational demands. Moreover, it allows for interchangeable backbone architectures (e.g., VGG or EfficientNet) to adapt to various performance and hardware constraints. In this work, we use ResNet50 as the backbone, striking a practical balance between accuracy and efficiency.

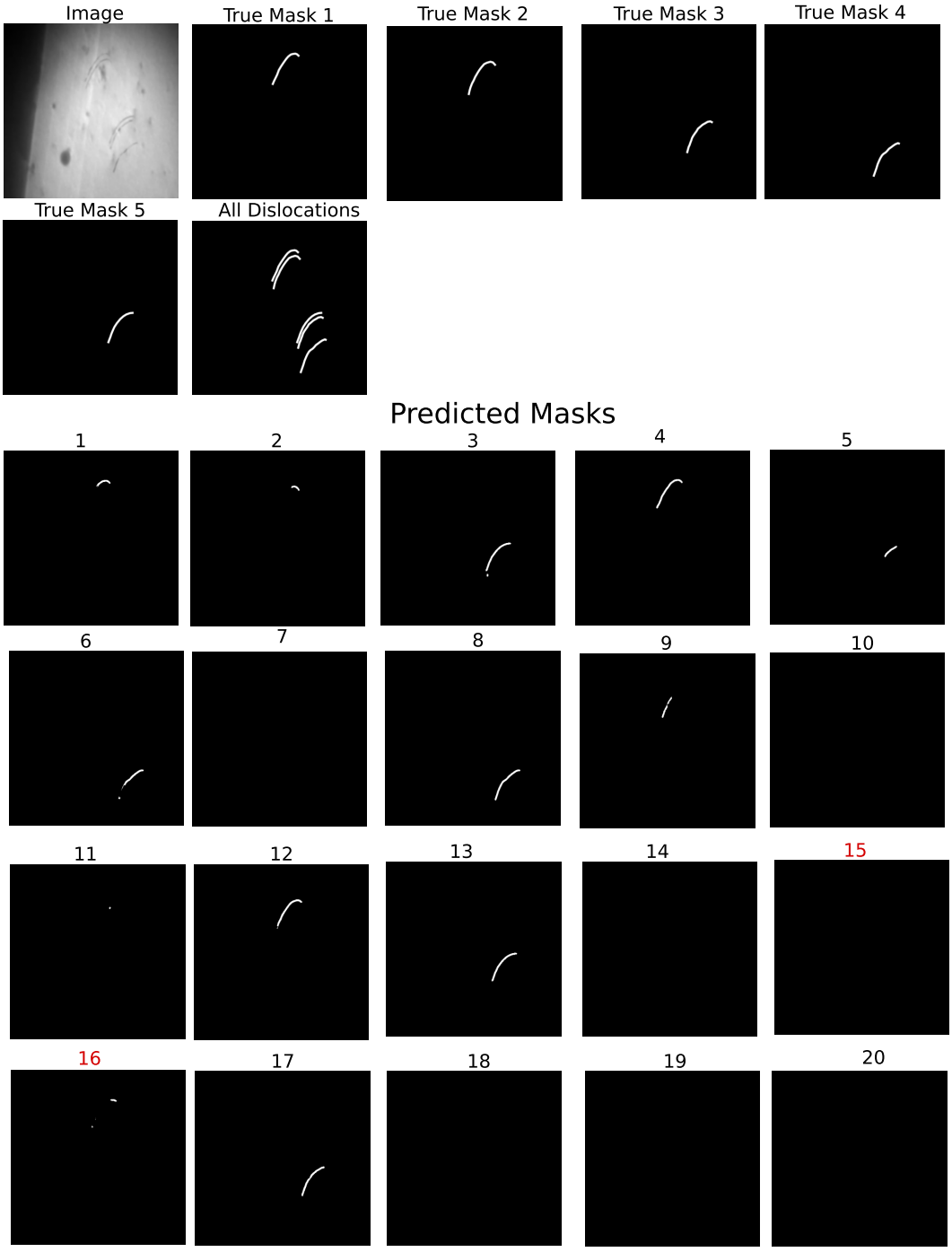


Figure 3.7: Synthetic image of a dislocation microstructure with five dislocations, showing both ground truth and predicted masks. **Top rows:** Each individual dislocation is represented by its own ground truth mask. **Bottom rows:** 20 predicted masks, where the model attempts to predict a single dislocation in each mask.

The model takes images of 512×512 pixels as input and outputs N masks of the same spatial dimensions, where N is the maximum estimated number of dislocations in an image. By assigning a separate mask to each dislocation, we enable subsequent analysis that requires precise identification and localization of individual dislocations.

Let us consider a synthetic image, along with the corresponding true masks as depicted in Figure 3.7. This image shows a dislocation pile-up of five dislocations, numbered from 1 to 5. Ideally, we would expect that the machine learning model predict the five dislocations in five individual masks, and the remaining 15 predictions should be empty masks.

Algorithm 1 Compute Average Image Loss for all Dislocations

- `true_masks` ▷ list of ground truth masks, e.g., the i-th mask is `true_masks[i]`
- `pred_masks` ▷ list of predicted masks, e.g., the j-th mask is `pred_masks[j]`

```
function ComputeAverageLoss(true_masks, pred_masks)
    M ← number of elements in the list true_masks
    avg_loss ← 0

    ▷ Iterate through all ground truth masks and find the best, predicted mask for each
    for i ← 1, M do
        ▷ obtain the loss for all predicted masks and the ground truth mask i
        N ← number of elements in the list pred_masks
        loss_values ← [DiceLoss(true_masks[i], pred_masks[1]),
                                        ⋮                ⋮
                       DiceLoss(true_masks[i], pred_masks[N])]

        ▷ find the index of the predicted mask with the lowest loss value
        idx ← argmin(loss_values)

        ▷ update the resulting loss of the image, given that there are M masks
        avg_loss ← avg_loss + loss_values[idx] / M

        ▷ ... and exclude the predicted mask to avoid double counting:
        remove the mask at index idx from list pred_masks
    return avg_loss
```

However, for simplicity, this is not enforced, and we also allow that these masks may contain

dislocations. During our work we found that enforcing this additional condition results in poor optimization of the models since the model does not know how to assign the dislocation in the masks. In that case, any duplicate dislocation can be easily filtered out in a subsequent post-processing step.

To train the model to predict only one dislocation per mask, we propose a novel loss function which, at the core, is based on the widely-used Dice loss [111]. The Dice loss is calculated as

$$\text{Dice loss} = 1 - \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}\,, \tag{3.1}$$

where TP (True Positives) is the number of pixel correctly predicted as part of a dislocation, FP (False Positives) is the number of pixel incorrectly predicted as part of a dislocation, FN (False Negatives) is the number of dislocation pixels that were missed. Our loss function is based on the fact that each ground truth mask contains exactly one dislocation. The number of possible masks for the prediction is a parameter that is specific to the dataset and needs to be equal or larger than the number of dislocations in the image (usually, it is easy to determine, if the material scientific experiment considers 2 or 20 dislocations). Figure 3.7 also shows an additional aspect of the prediction of multiple masks: e.g., the predicted dislocation number 1 does not necessarily corresponds to ground truth mask number at location 1. Hence an important step is to find for each dislocation the most suitable mask. This is done by calculating the Dice loss for all combinations of true masks and predicted masks. For a specific dislocation (i.e., true mask), the corresponding predicted mask with the lowest Dice score is chosen as the most suitable prediction. This approach of our newly designed loss function is presented in Algorithm 1. Using this approach we can predict very close or overlapping dislocations by predicting each dislocation in separate masks.

In the previous three sections, we discussed various machine learning approaches that we can use either directly or indirectly to represent dislocations as splines. However, one of the critical aspects of testing these models on unseen data is defining the evaluation criteria to quantitatively measure their performance. Proper evaluation metrics help in assessing model quality, comparing performance across models, and identifying areas for improvement. In the next section, we discuss in details these metrics and introduce a custom metric specifically tailored to dislocation length in segmented instances.

3.4 Metric Based on Dislocation Length

3.4.1 Common Metrics for Segmentation

In segmentation tasks, evaluating how well a predicted mask aligns with the ground truth mask is crucial. The most commonly used metrics include the Intersection over Union (*IoU*), the mean Average Precision (*mAP*), and the Dice score. Each metric captures different aspects of a model's performance in detecting and localizing the objects or regions of interest.

The IoU metric, also known as the Jaccard index, measures the overlap between the predicted mask and the ground truth mask. It is defined as:

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|}, \tag{3.2}$$

where P is the set of pixels in the predicted mask, G is the set of pixels in the ground truth mask, $|P \cap G|$ is the area of overlap, and $|P \cup G|$ is the total area covered by both the predicted and ground truth masks. The IoU ranges from 0 to 1, with higher values indicating a more accurate overlap. Models achieving higher IoU values are generally considered more precise at delineating object boundaries.

The mAP metric is widely used in both object detection and segmentation tasks. It considers the model's precision and recall at various IoU thresholds and then takes the average over these thresholds. The mAP is defined as:

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^{n} \text{Precision}(i), \tag{3.3}$$

where $\text{Precision}(i)$ is the precision at the i-th IoU threshold, and n is the number of IoU thresholds used.

Precision measures how many of the predicted positives are actually correct, and is given by:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{3.4}$$

where TP (true positives) is the number of correctly predicted instances, and FP (false positives) is the number of predicted instances that do not match the ground truth. A prediction is often considered a true positive if the IoU between the predicted and ground truth masks exceeds

a chosen threshold (e.g., 0.5). If the IoU is below this threshold, the prediction is counted as a false positive.

Recall (also called sensitivity or true positive rate) measures the proportion of actual positives that the model correctly identifies:

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{3.5}$$

where FN (false negatives) denotes the number of instances that were missed by the model. A high recall indicates that the model successfully detects most of the true objects in the ground truth.

By aggregating precision at multiple IoU thresholds, mAP provides a robust evaluation of how effectively the model detects and localizes objects over varying degrees of overlap requirements.

The Dice score measures the overlap between predicted and ground truth masks but places greater emphasis on correctly predicted pixels, making it particularly useful where precise boundary alignment is essential. The Dice score is defined as:

$$\text{Dice score} = \frac{2 \times TP}{2 \times TP + FP + FN}, \tag{3.6}$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively. A Dice score closer to 1 indicates a high degree of overlap between the predicted and ground truth regions, whereas a score near 0 indicates poor agreement. Because the Dice score balances overlap with penalties for both false positives and false negatives, it is especially relevant in medical image segmentation and other scenarios where small boundary misalignments can critically affect the result.

3.4.2 Need for Custom Metric

Typically metrics such as Precision, Recall, IoU, and Dice Score are used to evaluate segmentation quality, but they may not fully capture the nuances needed to measure the performance of the models based on how useful the predicted results might be in representing the dislocations as splines.

To understand it better, let us consider a TEM image consisting of a single dislocation along with the true mask shown in Figure 3.8. The dislocation constitutes only 222 pixels out of 4096 total pixels of the image while the rest of the pixels belong to the background. Values of different

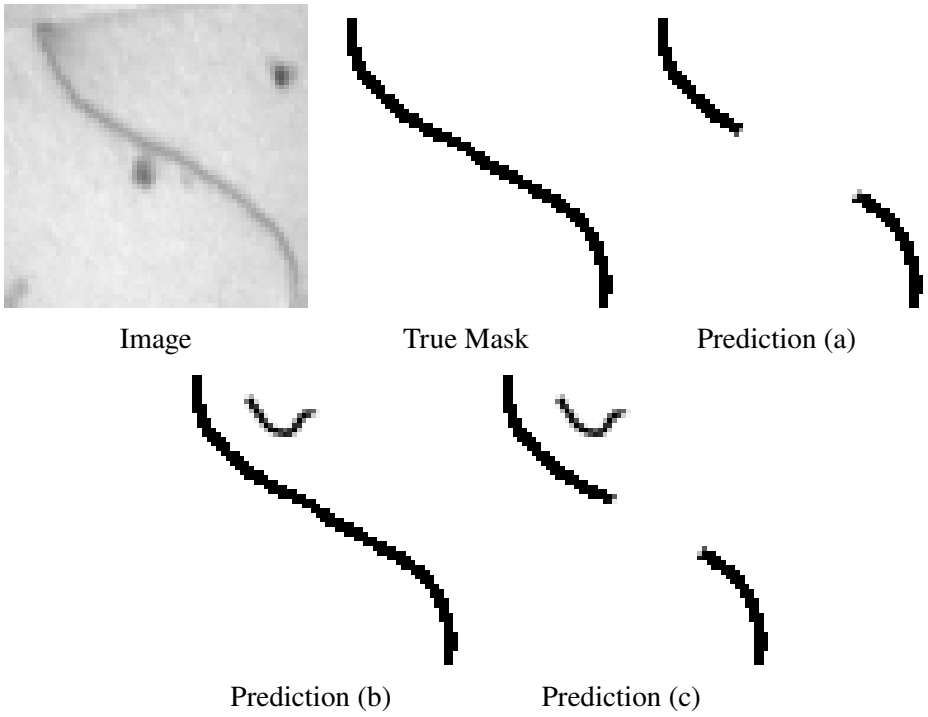


Figure 3.8: TEM image of a single dislocation with the corresponding true mask. (a) Model prediction showing underestimation of the dislocation components. (b) Model prediction where a portion of the background is incorrectly identified as part of the dislocation. (c) Model prediction capturing both false positives and false negatives.

metrics for the 3 different predictions for the dislocation are presented in Table 3.1. From these values, we can make the following observations:

1. The prediction shown in (a) does not contain any false positives (FP), which means that every pixel identified as a dislocation is indeed a part of the dislocation. This results in a Precision score of one, indicating perfect precision. However, it's crucial to understand that precision alone does not give a comprehensive picture of the model's performance. This is reflected in the Recall score, which at 0.68 indicates that some of the dislocation pixels were not correctly identified, i.e., a substantial proportion of true positives were missed by the model. Therefore, despite perfect precision, the model's overall performance is not ideal due to the lower recall.

2. In contrast, the prediction shown in (b) reflects a different bias in the model's performance. Here, Recall is one, meaning the model has successfully identified all true dislocation pixels. However, this comes at the expense of precision, which has dropped to 0.888 due to the model predicting parts of the background as dislocation, thereby increasing the number of false positives. Despite this, the Dice score has improved significantly, surpassing 0.94

Table 3.1: Metrics on 3 different predictions as shown in Figure 3.8

Predictions	IoU	Dice	Precision	Recall	TP	FP	TN	FN
Prediction (a)	0.68	0.81	1.00	0.68	151	0	3874	71
Prediction (b)	0.88	0.94	0.88	1.00	222	28	3846	0
Prediction (c)	0.67	0.80	0.85	0.75	168	28	3846	54

compared to earlier value of 0.81, demonstrating that the model has a relatively balanced performance in terms of precision and recall, leading to a higher Dice score.

3. The prediction shown in (c) presents a case where both false negatives and false positives exist, implying a blend of the issues seen in predictions (a) and (b). The model in this case exhibits both under-prediction and over-prediction, causing an imbalance in recall and precision. Despite this, both the IoU and Dice scores suggest comparable performance to prediction (a), which underscores the limitations of relying on a single metric for model evaluation. It's critical to take into account both type I (false positive) and type II (false negative) errors for a more holistic understanding of model performance.

From the above example, we can see that even though a number of metrics are available, each provides unique information and we cannot use just one of them to compare the predictions. Considering this, we developed the physics based metric to evaluate the performance of the machine learning models.

How about adding more geometrical features to the loss, such as the orientation or even the (local) curvature? This might seem like a useful approach since the stress locally acting on a dislocation can depend on such details. However, we found that errors from orientation differences are already indirectly considered in the Dice score since it uses the intersection of the ground truth with the prediction. Additionally, the computational cost for evaluating more complex metrics can also be significant, including additional geometric features was not considered any further.

3.4.3 Custom Metric Formulation

When dealing with dislocation segmentation, the line-like nature of dislocations often renders standard pixel-based metrics less effective as discussed above. Dislocations, being one-dimensional defects within a crystal lattice, are best quantified by measures that capture their length and geometry. To address these challenges, we propose a custom metric that operates on the principle of comparing the predicted dislocation length to the ground truth.

Let us consider an image as shown in Figure 3.9. The image has two dislocations which are

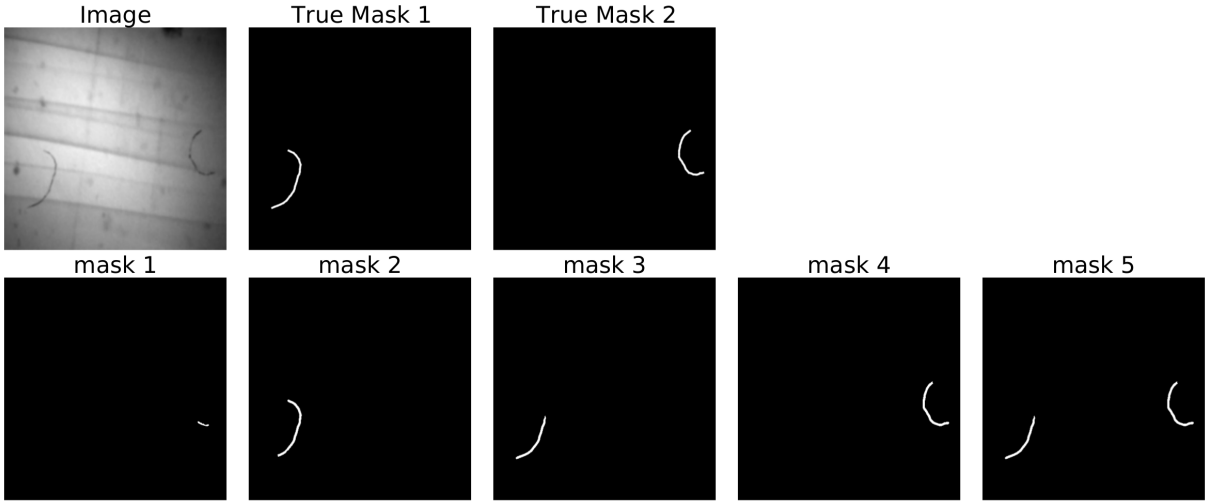


Figure 3.9: An example training data for illustration of the proposed length-based metric for evaluating machine learning model performance. Top row: An image of dislocation microstructure with two dislocations (True Mask 1 and True Mask 2). Bottom row: Predicted masks of dislocations for multi-label segmentation approach.

shown in two masks labeled as true mask 1 and true mask 2. For multi-label segmentation approach, we would expect model to predict a single dislocation in a single mask. This is usually not the case and we might have multiple dislocations predicted in a single mask or a mask predicting only a part of a dislocation. Let us assume that the model is trained to predict a maximum of five dislocations and the five predicted masks for the image are shown in Figure 3.9. Mask 2 and mask 3 are predicting same dislocation but mask 3 predicts only a part of the dislocation and is less accurate. Mask 1 predicts only a very small part of dislocation. Mask 5 is rather interesting where we have one of the dislocation very accurately predicted but a small part of second dislocation is missing. Once we have the predictions of the dislocation masks as shown in the Figure 3.9, we can use the true masks to evaluate metric. Below, we describe the methodology in detail, starting from predicted masks preprocessing to metric calculation.

1. First, we train a model to segment dislocations. Despite the model's capacity to produce multiple masks, the underlying goal is that each predicted mask corresponds to one dislocation. If the model merges multiple dislocations into a single mask (mask 5 in the figure) or breaks a single dislocation into multiple parts, it may negatively impact the final metric score. This requirement reflects the real-world physical expectation that each dislocation line is distinct and continuous.

2. The initial prediction from the models typically comes in the form of probabilities, where each pixel indicates the likelihood of belonging to a dislocation. To simplify further processing, a threshold value of 0.5 is chosen. Pixels exceeding this probability are assigned

a value of 1 (dislocation), and those below are assigned 0 (background). This binarization step makes subsequent operations, such as connected component analysis, more robust by eliminating low-confidence predictions.

3. We now perform an 8-connectivity analysis [112] using the `OpenCV` library [113] on the binarized masks. This process groups neighboring pixels (in all eight directions around a central pixel) into distinct components. Each connected component ideally represents one dislocation. However, it is often observed that a single component may contain multiple small fragments or multiple close dislocations that got merged into one predicted region. Therefore, connected component analysis provides a systematic way to label and separate these distinct regions, allowing us to inspect them individually.

4. In realistic microstructural images, noise or partial artifacts can give rise to very small connected components that do not represent true dislocations. We remove such spurious regions by discarding all components whose pixel count is below a user-defined threshold. The threshold is usually determined empirically, balancing the risk of removing actual (but tiny) dislocations against preserving clean, meaningful structures. By filtering out these tiny regions, we ensure that the subsequent steps focus on more substantive dislocation candidates. Out of the four masks in Figure 3.9, this step would filter our mask 1 and we would be left with the remaining four masks (mask 2 to mask 5) for further steps.

5. Once we isolate the connected components of interest, we apply Lee skeletonization [114], a thinning algorithm that iteratively reduces thick regions to their minimal one-pixel-wide representation without significantly altering the topology. This step is crucial because it converts a two-dimensional blob of pixels (which may represent the cross-section of a dislocation in the image) into a one-dimensional representation. As dislocations are inherently line defects, the skeleton better reflects their true geometry. Each skeleton thus becomes a set of connected pixels that trace the approximate centerline of the dislocation, preserving its overall shape and connectivity.

6. With the dislocation reduced to a skeleton, we fit a spline curve to the skeleton points. The spline provides a smooth, continuous function that captures the meandering path of the dislocation. By treating the skeleton points as control points, the fitting routine ensures that local geometric variations are preserved, yet extraneous noise is mitigated. This step is instrumental in obtaining a stable length measurement: a spline can be parameterized and measured more precisely than a set of discrete points. At this step we perform another filtering based on number of spline curves to get rid of masks which predicts more than one dislocations such as mask 5. For the image, this would give us mask 2, mask 3 and mask 4 to be valid for further steps.

7. Now we have the true masks and the filtered predicted masks where each mask represent only one dislocation. It might happen that two masks such as mask 2 and mask 3 represents

same dislocation. To find the most accurate predicted mask which corresponds to a true mask, we follow the same steps as followed in the loss calculation and find the predicted mask with the highest dDice score for a true mask. We might have a case where none of the predicted masks corresponds to a true dislocation and the Dice score is zero and hence we only consider the cases where the maximum Dice score is at least 0.1. We follow the same steps for the remaining true masks and obtain corresponding predicted masks. For the image in Figure 3.9, predicted mask 2 corresponds to true mask 1 and mask 3 corresponds to true mask 2.

8. Finally, we compare each predicted dislocation spline to the corresponding ground truth spline. The predicted dislocation length L_P and the ground truth dislocation length L_T are measured by summing the arc length of the fitted splines. The metric score for the dislocation is given by:

$$\text{Metric Score for a dislocation} = 1 - \frac{|L_P - L_T|}{L_T}. \tag{3.7}$$

A score of 1.0 represents a perfect length match, indicating that the predicted dislocation aligns well with the ground truth in terms of both location and extent. By definition, any prediction containing multiple dislocations in a single mask automatically receives a score of 0 for that instance, as it violates the one-dislocation-per-mask constraint. This strict penalty evaluates the model's ability to generate predictions that accurately reflect distinct dislocation lines rather than merging or predicting only a part of them. We then calculate the score for all the dislocations and average them to obtain metric score for an image.

This length-based metric offers several advantages over pixel-wise metrics like the Dice score or IoU metric. By focusing on a line defect's fundamental property—its length—we capture the physically salient feature of dislocations. Traditional metrics might award high scores simply because large areas overlap, ignoring cases where the dislocation path is fragmented or inaccurately traced. Nonetheless, one limitation of this metric is its reliance on enforcing a single predicted dislocation per mask. If a model tends to over-segment or under-segment the dislocations, it may be harshly penalized. While additional post-processing (e.g., advanced contour extraction or manual splitting/merging) could refine such masks, these procedures generally require ground truth guidance or heuristic logic, thereby complicating model evaluation. Hence, we prioritize a direct metric, ensuring that any gains in performance truly reflect improvements in how the model handles individual dislocation lines.

Chapter 4 Study Using Synthetic Data

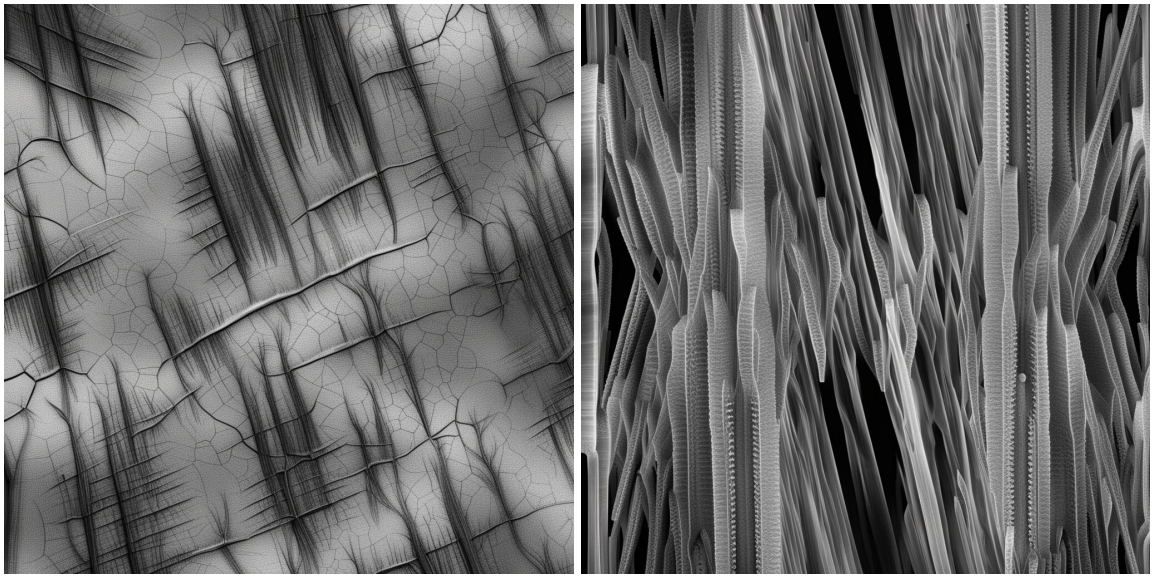
In this chapter we start by exploring generative AI dreambooth approach to train stable diffusion XL by fine-tuning on limited amount of real images to obtain unlabelled synthetic images of dislocation microstructure. This is presented in detail in Section 4.1. We then start exploring the synthetic data generation approach proposed in this work by addressing some of the open research questions which are presented in details in Section 4.2. This is followed by three synthetic datasets that we have used in this work presented in Section 4.3. In the next two following sections we perform a number of machine learning studies on the three datasets using the three machine learning approaches discussed in this work.

4.1 Dreambooth Approach to Generate Unlabelled Synthetic Images.

The parametric synthetic data generation model proposed in this work can be used to generate both the synthetic images of dislocation microstructures as well as the ground truths for various machine learning tasks. Some of the synthetic images generated using the model are shown in Figure 2.13.

Figure 4.1: Dreambooth approach to finetune text to image models to generate personalized images of a subject in different context. Adapted from Ruiz et al. [115]

At first glance, an experimentalist with experience in TEM image data can easily identify if the images are real or not. Even if the synthetic data generation model captures the correct morphology of dislocations but the texture of the image might not be realistic. This is one of the shortcomings of such parametric based models especially when it comes to images where they fail to capture the correct texture of the real images since most of the time the texture is synthetically generated and only approximates the real texture without developing any model for textures.

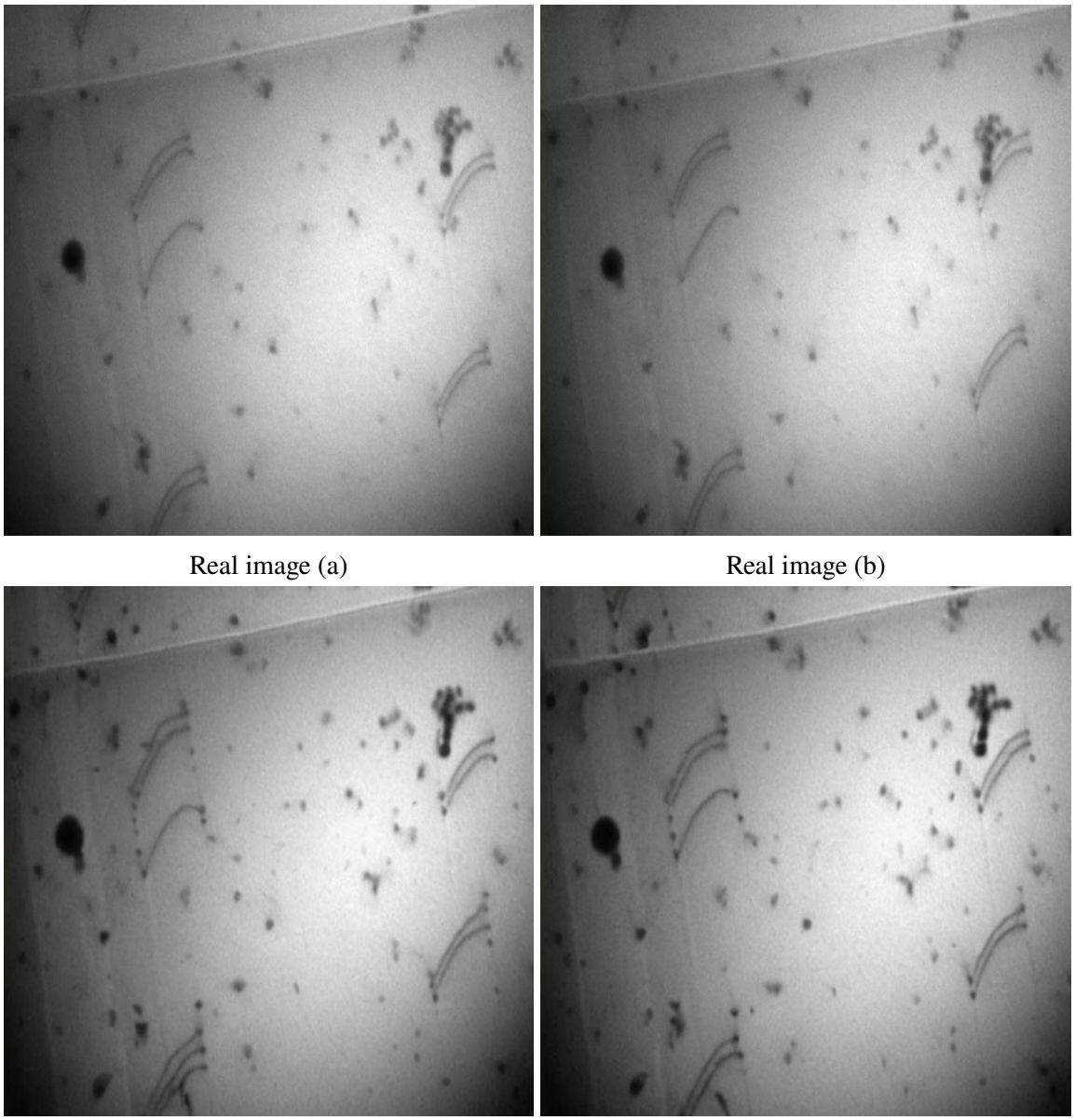


a b

Figure 4.2: Generated images without finetuning stable diffusion model XL model using prompt (a) Greyscale Microscopy image of dislocation microstructure and (b) Greyscale Microscopy image of 4 line like structures.

But the advantage of such approaches is the ability to generate large amount of data along with ground truths. Let us consider an alternate approach based on diffusion models which can be used to generate realistic images of dislocation microstructure but without any labels. Focus is to generate realistic synthetic images. Since we only have limited amount of real images to train the model, we consider a dreambooth approach to finetune the stable diffusion model XL [115, 116]. Even though the model can be used to generate very high resolution realistic images of 1024 pixels, the model lacks the ability to mimic appearance of subjects. If we would like to generate an image using the prompt "Dog playing with a ball in a park", the model would generate different images of a dog and a ball every time we run the model. The dreambooth approach of finetuning stable diffusion model allows us to embed a particular subject of interest in the images so that the same subject can be rendered in different contexts. In the approach, a

unique identifier is used in the prompt, "a sks dog playing with a ball in a park" where the unique identifier sks is used for the dog.



Real image (a) Real image (b)

Synthetic Image (a) Synthetic Image (b)

Figure 4.3: Two Real images from experimental data and synthetic images generated after finetuning using Dreambooth approach. The model lacks the ability to generate images of large variations and only generated images identical to provided real images.

Once the model has been trained and the new identifier has been added to the vocab so that the model can associate the identifier with the subject then we can use it to generate the images of the subject in different context using the identifier. A Prompt like "a sks dog on a mountain top with snowing in the background" would then generate images of the same dog as shown in Figure 4.1.

One of the most attractive feature of this approach is that it requires very little amount of data. Depending of the subject, it might be as small as tens of images only. We adapt this approach to finetune the stable diffusion model XL [117] for dislocation dataset. Let us first try to generate the images of dislocation microstructure using the model without any training. Two of the text prompts used and the generated images are shown in Figure 4.2.

We can see that the model generates images of a microstructure-like structure but they do not represent a dislocation microstructure. We tried several other prompts also but we obtained similar looking images more or less. The model fails to understand what does a dislocation microstructure look like since the underlying foundation model [1] has never been trained on microscopy image of dislocation microstructures.

To start with, we use 10 images from a single experimental data. We used the prompt "Greyscale Microscopy image of sks microstructure" where "sks" is the unique identifier and microstructure is the class. In the finetuning process, we are forcing the model to associate the unique identifier "sks" with the dislocations so that whenever we have the unique identifier in the prompt, it generates dislocations. Two of the images from the data are shown in Figure 4.3. Once the model is finetuned, we use it to generate some synthetic images, two of which are shown. First thing to note is the high quality of the synthetic images generated. The texture and dislocation microstructure look very realistic, some of the images are at par with real data. The dislocation microstructure generated shows very little variation and looks very similar to the microstructure of the real dataset provided for training.

To improve the generalization of the model, we use 230 real images from several different experiments. The aim is to finetune the model so that it can generate a large variety of dislocation microstructures. Predictions of the finetuned model using the same prompt are shown in Figure 4.4. We find that the model is now able to generate dislocation microstructure not present in the training data. The generated images have varying amount of dislocations and dislocation pileups. Although the generated images appear realistic, synthetic data may still have subtle biases compared to real microstructures. We would like to aim for images where we have a cleaner dislocation microstructure such as image c and d. These are the images which might be of more importance to us where we can perform a quantitative study compared to images shown in a and b. But nonetheless the generated synthetic images are of high quality and provides us an opportunity to have large amount of unlabeled dataset for a machine learning task which might

[1] A foundation model refers to a large-scale machine learning model, typically pre-trained on vast amounts of data, that can be fine-tuned for a wide range of downstream tasks. Examples include GPT-4, Stable Diffusion XL, BERT, and CLIP.

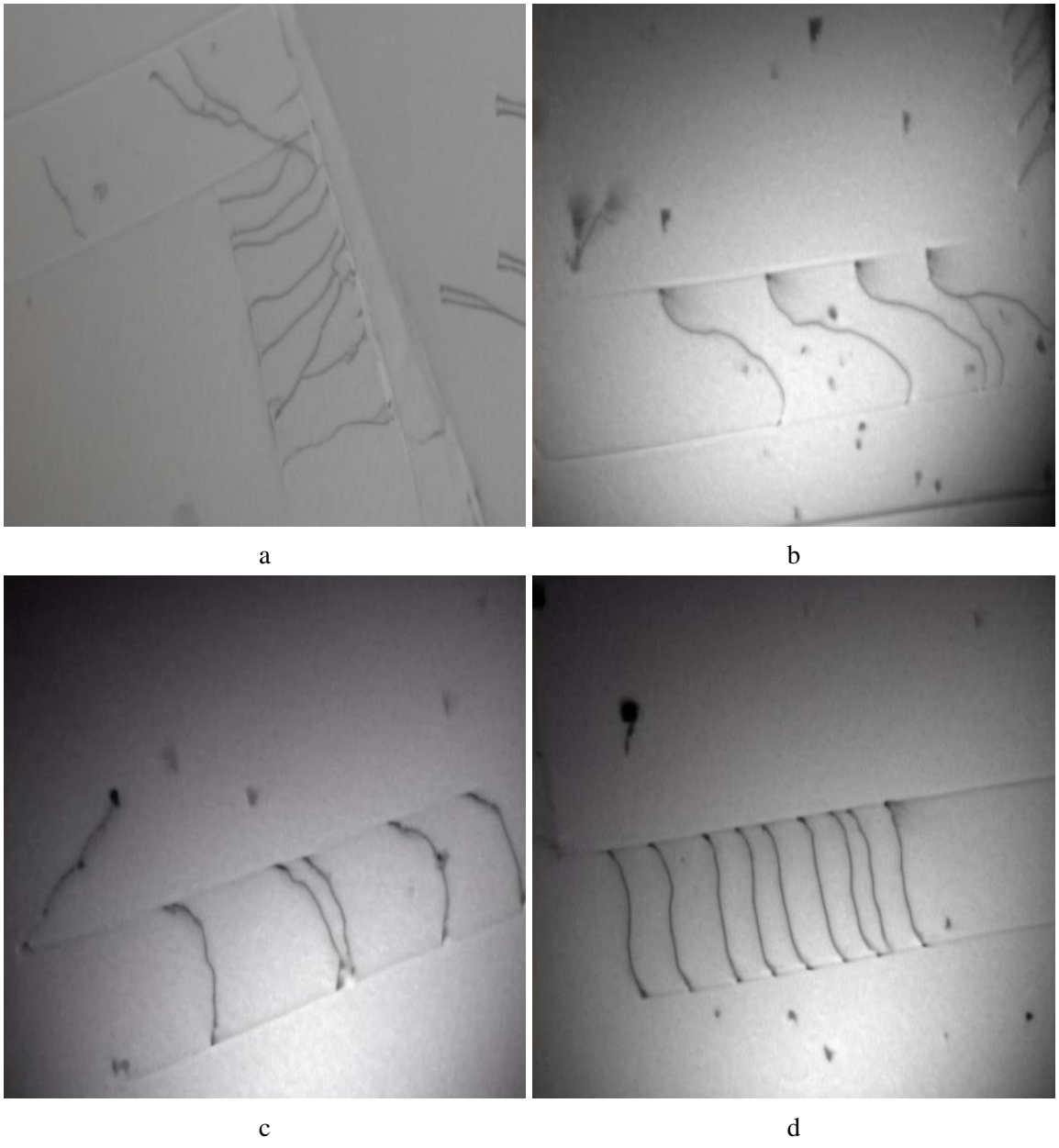


Figure 4.4: Synthetic dislocation microstructure images generated after fine-tuning on general real data. The generated images exhibit substantial variation, presenting distinct structural differences from the original dataset, while maintaining characteristic features of dislocation microstructures.

be useful for active learning and unsupervised approaches.

4.2 Research Questions Related to Synthetic Datasets

The synthetic data generation model proposed in this work provides the capability to create a diverse array of synthetic images, varying not only in dislocation microstructure but also in the background composition. This flexibility allows for the choice between a synthetic background, generated using Perlin noise, and a more realistic background derived from real image patches. Additionally, a wide range of dislocation microstructures can be generated, dependent on specific parameter values. This leads us to several pertinent questions related to the synthetic data which we will address in this particular study:

Q1: Which type of background should be used (synthetic or realistic)?
The choice of background in synthetic datasets for machine learning models is an important factor that directly impacts their performance in real TEM image analysis. This decision influences the model's ability to accurately discern true dislocations from other elements such as spots or secondary phase particles, often present in real TEM images. The type of background, whether synthetic or realistic, plays a crucial role in mimicking various lighting conditions and imaging environments typical of TEM analysis. It's essential to determine which background type more authentically replicates these conditions, thereby enhancing the model's effectiveness in real-world applications. This consideration is particularly relevant in light of our efforts to simulate complex dislocation configurations and microstructural variability,

Q2: How can we generate synthetic dislocation microstructures that are representative of real dislocation microstructures?
A key objective is to produce synthetic dislocation microstructures that closely mirror the characteristics of real dislocation microstructures, particularly when analyzing spatio-temporal data from experimental TEM videos. While training a model on such specialized synthetic data can yield high accuracy on corresponding real data, there is a potential trade-off between specialization and generalization to other real datasets with different characteristics. Achieving this requires careful tuning of the synthetic generation process to match real-world features such as the number of pileups, slip orientations, slip widths, and the shapes and counts of dislocations in each pileup. Integrating domain-specific knowledge into the generation process through the selection of appropriate parameters is essential for this fidelity but is it really required ? Can we have a general microstructure which can also perform well on a real dataset ?

Q3: Are machine learning models trained on synthetic datasets effective and can provide high quality results when applied to real data ?
This issue is central to the research because the primary objective is to create and utilize synthetic datasets for training purposes. The crux of the matter lies in evaluating the performance of these models when they transition from synthetic to real datasets. It is essential to understand the concept of "domain" in the context of synthetic and real images. In machine learning and computer vision, the term "domain" often refers to the source and characteristics of the data.

In this case, the domain represents the intrinsic properties and the environment from which the images originate. Synthetic images are generated using the parameter based model. These images, while potentially diverse and rich in variations, inherently possess characteristics distinct from real images. These characteristics might include textures, lighting conditions, object arrangements, or even subtle nuances that are not typically found in real images. Consequently, these differences establish the "synthetic domain" with its unique set of features and distributions. Real images, on the other hand, are TEM images generated using cameras. They include the randomness, imperfections, and complexities of the sample. This includes variations in lighting, shadows, natural textures, and real-world scenarios that are hard to replicate accurately in synthetic datasets. Thus, the "real domain" is marked by its realistic and often unpredictable nature. The challenge, therefore, lies in domain generalization – the ability of a model trained on one domain (synthetic) to perform well on a different domain (real). This requires the model to learn representations that are not only relevant to the synthetic data it was trained on but also generalize to the real data it will encounter. The question is whether the models can effectively bridge the gap between the synthetic and real domains, especially given that synthetic images can be visually and fundamentally different from real images. To reduce this gap, one of the most widely used methods is to use strong data augmentation methods. Data augmentation methods such as adding random noise, brightness, contrast, image equalization to the synthetic images can be very beneficial in generalizing the trained machine learning model to real images.

Q4: Can we obtain a ML model which is able to generalize to a large number of dislocation microstructures and what are the factors we need to consider to generate a general synthetic dataset?
Synthetic data generation allows us to generate a wide variety of dislocation microstructures. The generation of synthetic data offers the distinct advantage of creating diverse dislocation microstructures, which is pivotal for training robust Machine Learning models. The central question is whether it is feasible to develop an ML model that can generalize across a broad spectrum of dislocation microstructures. Achieving this requires careful consideration of several factors during the synthetic dataset generation process.

To address the above questions, we propose generating three distinct synthetic datasets which differ in their microstructure features as explained in details in following section.

4.3 Three Synthetic Datasets

In this section, we will present three synthetic datasets used for later studies. The section discuss in details the parameters to use to generate synthetic data and ways to generate a generalized synthetic data.

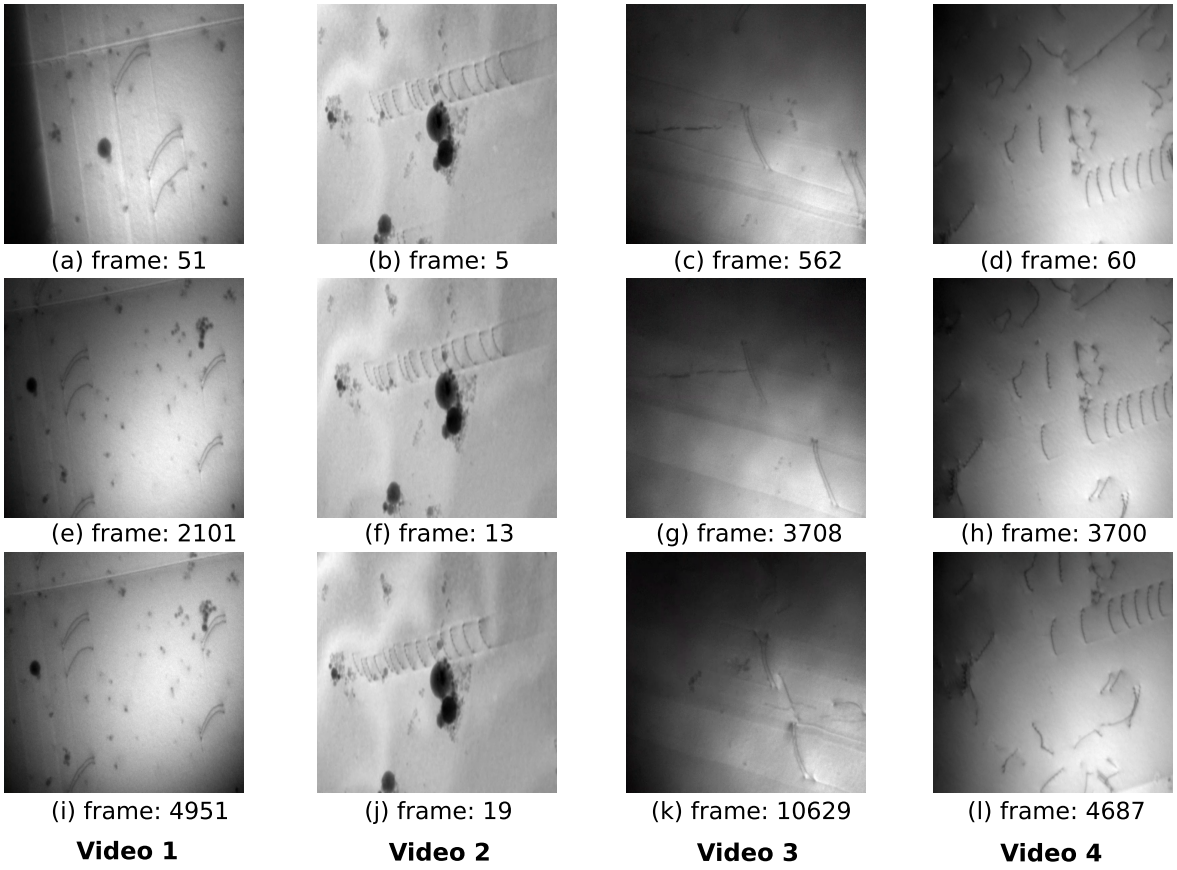


Figure 4.5: Three frames extracted from four different experimental videos, illustrating the variation in the dislocation microstructure across different videos. Frames from the same video highlight microstructure similarities, while those from different videos highlight variations in the observed microstructures.

4.3.1 Synthetic dataset SD1

From the dataset of 23 experimental videos, we have selected a specific video, referred to as *video1* for detailed analysis as shown in Figure 4.5, featuring three representative frames from the complete real data designated as RD1. The observed dislocation microstructure in RD1 comprises numerous dislocations, predominantly in paired configurations. In the video sequence, these pairs demonstrate coordinated motion, characterized by sub-pixel spacing between the dislocations within each pair.

For this real dataset, we generate a corresponding synthetic dataset, designated as SD1. This synthetic dataset is constructed to replicate key features of the dislocation microstructure observed

in the real dataset. To initiate this process, we performed microstructural characterization of the real dataset. This was achieved by manually annotating dislocations in 15 frames, uniformly sampled from the entire video. During this annotation process, we assigned specific identifiers to each dislocation, along with their respective pileup numbers within a given frame which are useful in feature analysis.

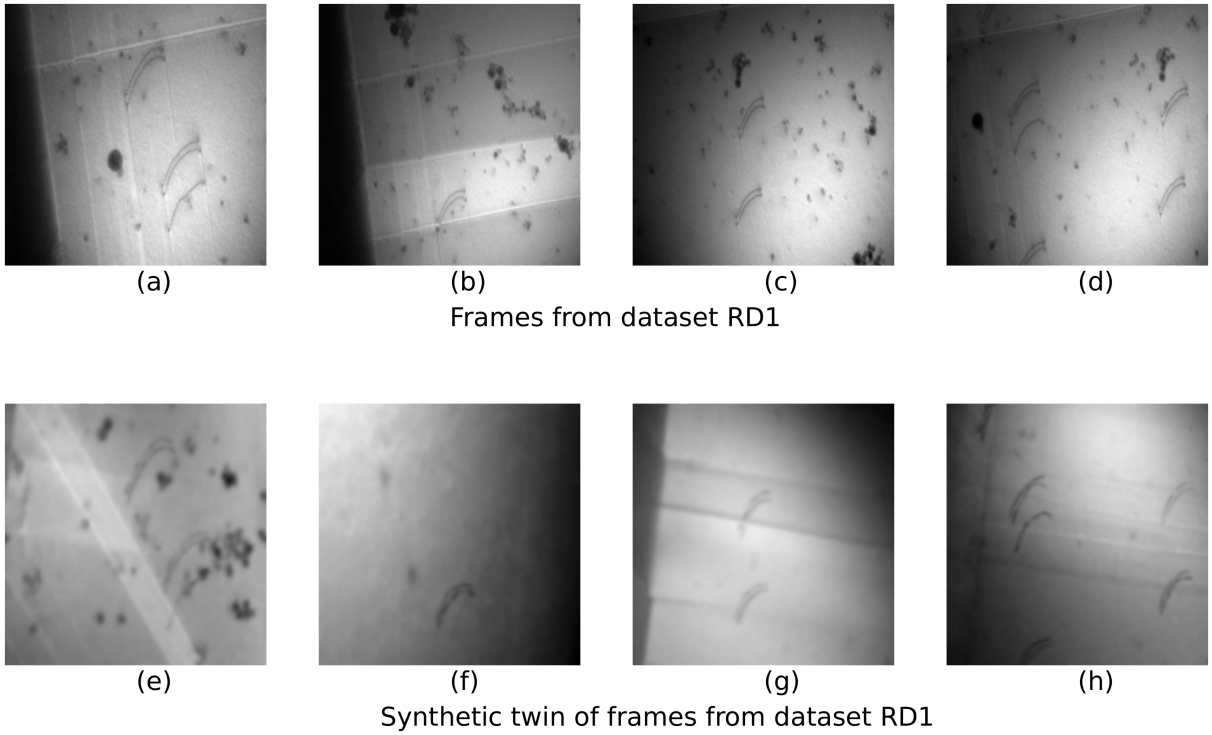


Figure 4.6: Frames from dataset RD1 and corresponding synthetic twins of the dislocation microstructure.

To synthesize a microstructure analogous to that in RD1, two distinct methodologies can be used. The first approach involves a direct replication of the real dislocation microstructure. This replication is based on the coordinates of dislocations obtained from manual annotations of the real images. Utilizing these coordinates, we can create synthetic twins for each of the 15 manually annotated frames, resulting in 15 synthetic microstructures. Some frames from dataset RD1 along with the corresponding synthetic twin microstructure for each frame is shown in Figure 4.6 where we can see that the synthetic microstructure closely reproduce the real microstructure. To introduce variability, modifications can be made in the synthetic microstructures, including selective inclusion of dislocations, as well as variations in dislocation spacings, directions, and slip widths. This approach, while precise in replicating the real microstructure, is limited in its variation.

The second approach in generating the synthetic dataset, SD1, diverges significantly from the direct replication methodology employed in the first approach. Rather than precisely mimicking the exact microstructure of the real dataset, this method prioritizes the replication of fundamental microstructural characteristics. These characteristics include the number of dislocations within pileups, the number of pileups, slip widths, slip directions, and the spatial distance between neighboring dislocations in a pileup. By examining the 15 handlabeled frames, probability distributions for each microstructural characteristic are established. These distributions are then utilized to obtain distribution of the microstructure parameters of the synthetic data generation model for the generation of synthetic microstructures in SD1. In this approach, we still need spline support points for the dislocations. This is obtained from the real dislocation by converting the coordinates from the image frame to the local coordinate system as described in Section 2.2. This methodology ensures that while the synthetic dataset does not exactly replicate the specific microstructures found in the real dataset, it mirrors the overall statistical properties and distributions of key microstructure features of the real dataset RD1.

The primary advantage of this approach lies in its ability to generate a more diverse range of microstructures compared to the previous approach. While the first approach produces synthetic microstructures that are almost exact replicas of the real dataset, it inherently lacks variation. Each synthetic microstructure is tightly bound to the corresponding real structure it imitates, limiting the scope of diversity. Conversely, the second approach, by basing the synthetic generation on statistical distributions of key microstructural features, allows for a broader exploration of possible microstructures. This increased variability is particularly beneficial for deep learning applications, as it provides a richer, more varied dataset. Such diversity is crucial for training robust models capable of generalizing well to new, unseen data. This approach enhances the overall utility of the synthetic dataset, making it more representative of a wider range of potential real-world microstructures, thereby better supporting the development and validation of deep learning models for TEM image analysis.

The probability density distributions of the microstructural features for both synthetic dataset SD1 and real dataset RD1 are presented in Figure 4.7. These distributions illustrate that the synthetic microstructures closely mimic the feature characteristics observed in real microstructures. For instance, a synthetic image in SD1 may contain a single pair of dislocations, similar to the real image shown in Figure 4.6(b). The minimum number of dislocations in a real image is also one, indicating that each image generally includes at least one dislocation pair. Moreover, dislocation pileups with three dislocations are also present, as observed in the real images shown in Figure 4.6(a) and Figure 4.6(d). However, such configurations are relatively rare, as reflected by their lower probability density in the distribution.

The spacing distribution between nearest neighboring dislocations is noteworthy. The distribution exhibits two distinct peaks: one around 7 pixels and the other around 30 pixels. The first peak corresponds to closely spaced dislocation pairs, as seen in Figure 4.6(c), where

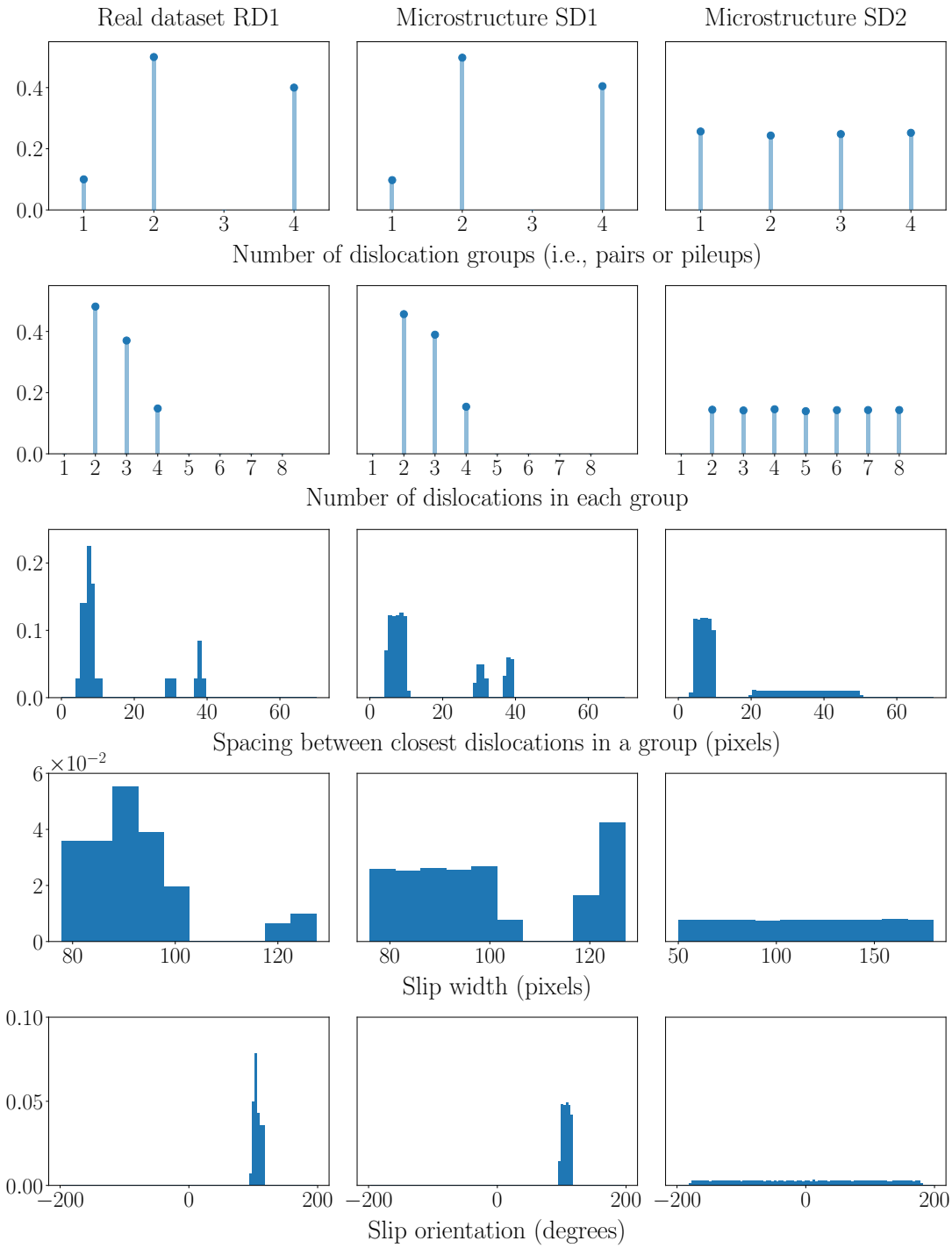


Figure 4.7: Probability density distributions of features of Synthetic microstructure of dataset SD1, SD2 along with the Real dataset RD1. The microstructure of SD1 is modeled based on Real data RD1.

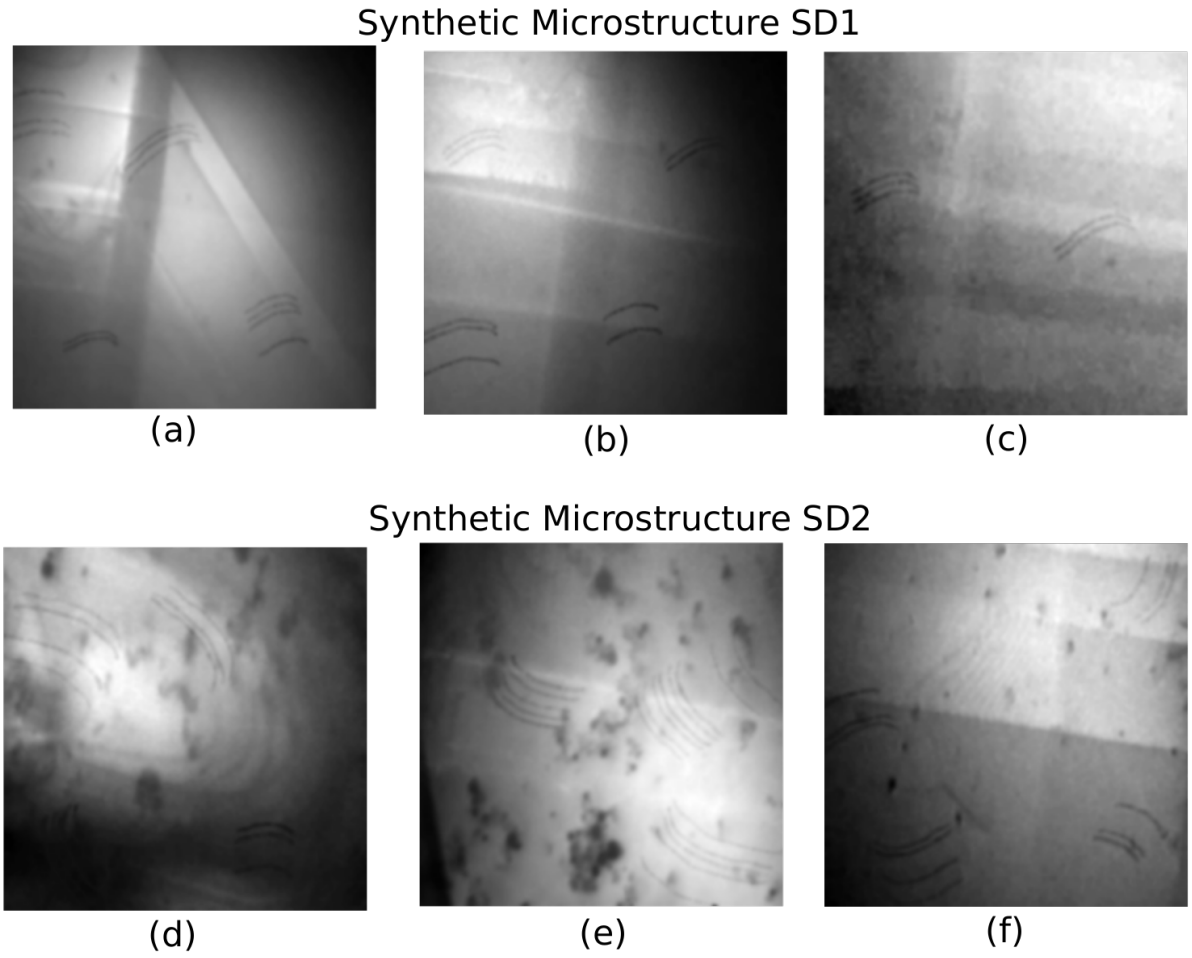


Figure 4.8: Example synthetic microstructure images from datasets SD1 (a-c) and SD2 (d-f), showcasing distinct microstructure patterns and variations representative of each dataset.

two pairs of dislocations are closely positioned. This consistent spacing suggests a potential short-range order in the dislocation microstructure, a feature that warrants further investigation. The second peak arises from configurations where a third dislocation is present in the pileup, positioned at a relatively constant, larger spacing from the other dislocations across frames.

In constructing the synthetic pileups, we ensure that no dislocations overlap or intersect, maintaining a realistic spatial arrangement. While the synthetic microstructures are designed to replicate the statistical distributions observed in real datasets, they do not exactly reproduce the precise configurations of real microstructures. This distinction can be appreciated when comparing real microstructures in Figure 4.6 with synthetic samples in Figure 4.8.

4.3.2 Synthetic dataset SD2

Synthetic dataset SD1 is generated to closely resemble the real dataset, RD1. However, machine learning models trained on such synthetic datasets may lack the ability to generalize effectively across other real microstructures from distinct experiments. Therefore, it is crucial to develop a synthetic dataset with broader generalization, capable of yielding high-quality results on diverse real datasets, not limited to a single example. To this end, we generate a more generalized synthetic dataset, designated as SD2.

The parameters used to generate the microstructures in SD2 are presented in Figure 4.7. In this dataset, the number of dislocation groups or pairs in a microstructure is randomly sampled between 1 and 4, with each group containing between 2 and 8 dislocations. Consequently, a dislocation microstructure can include up to 32 total dislocations. However, for simplicity, we have restricted the generation process to a total of 20 dislocations which would be useful for machine learning approaches like multi-label segmentation. Additionally, in dataset SD1, all dislocation pileups are approximately aligned in the same direction, as illustrated in Figure 4.8. This contrasts with dataset SD2, where the slip orientation for each pileup is uniformly sampled across a full range of orientations, from -180° to $+180^\circ$. Thus, multiple pileups within a single microstructure in SD2 may be aligned along various directions.

This characteristic is demonstrated more clearly in synthetic images (d) and (f) from dataset SD2. Dislocation pileups in SD2 are strategically placed to ensure that no two dislocations overlap or intersect, despite their differing slip directions.

There are several advantages of synthetic dataset SD2 over SD1. The first and most important advantage is the ease to generate the synthetic dataset. Compared to SD1 which requires a detailed study of the real dataset to obtain the distribution of the parameters to use to generate synthetic microstructure, relatively only a few parameter determination is required for SD2. A general estimate of the values of parameters like number of pileups, number of dislocations and slip width of each pileup can be made by just looking at the real microstructure or analyzing only single real image. We can uniformly sample the parameters from a general range as done to generate the whole dataset. There is one parameter which needs careful consideration, the support points for dislocation splines. Dislocations come in different shapes and variety, hence it is important to carefully choose the dislocation shapes in the synthetic microstructure for the machine learning model to learn. To model the dislocation as splines in the synthetic dataset SD2, we use the spline support points from the dislocations in the real dataset RD1. The synthetic dataset SD2 adds variation in the microstructure features which should improve the generalization and robustness of the model.

4.3.3 Synthetic dataset SD3

We have discussed two synthetic datasets so far. Dataset SD1 is modeled using the microstructure features from the real dataset and very closely represent the real dataset RD1 as shown in Figure 4.10. The synthetic dataset SD2 is much more general and provide large variation in terms of dislocation microstructure. The modeling of synthetic dataset SD2 is still dependent on atleast some of the features of the real dataset as discussed above, namely dislocation shapes. There is still a large variety of dislocation microstructure which are not included in SD2. We now generate a much more general synthetic dataset designated as SD3. We considered following while generating this particular synthetic dataset.

1. In Figure 5.1, various dislocation microstructures from different real images are displayed, demonstrating the inherent diversity in dislocation shapes and sizes within these structures. Notably, the scale of dislocations can vary significantly within the microstructural settings, ranging from diminutive sizes to extensive structures that span the entire image. This variation in relative size, or the "size effect," is a critical aspect in the context of dislocation microstructure analysis. It is essential to consider this size effect when generating synthetic microstructures, particularly for the development of machine learning models that are expected to generalize across a wide spectrum of dislocation sizes. While constructing the synthetic dataset SD3, specific attention was paid to the variability in dislocation sizes. To simulate a range of dislocation scales, the slip width was varied, with values uniformly sampled between 50 pixels and 400 pixels. This range was deliberately chosen to include a broad spectrum of possible dislocation sizes, thereby enabling the generation of more complex and diverse dislocation microstructures. This approach ensures that the synthetic dataset is not biased towards a particular size range, but rather provides a comprehensive representation of various dislocation scales. Such diversity in the synthetic dataset is vital for training machine learning models, as it allows these models with the ability to accurately identify and analyze dislocations of varying sizes in real-world TEM images. By incorporating a wide range of dislocation sizes, the synthetic dataset SD3 enhances the robustness and generalization of the resulting machine learning models, ensuring their applicability across a diverse array of microstructures.

2. In synthetic datasets SD1 and SD2, we have so far considered very simple bowed out dislocation shapes which can be seen very clearly in real images shown in Figure 4.8 but this is rarely the case. Figure 5.1 shows real images from several experiments and Figure 2.13 shows some synthetic images where we can see that dislocations can have much more complex shapes. Dislocations can even be long straight lines. A dislocation need not be just only convex or concave over its length. A part of dislocation can have several inflection points and change its change. Depending on several factors such as the presence of impurities, grain boundaries and other defects, dislocations may change its shape or direction. Impurities in a crystal lattice can pin dislocations, a phenomenon

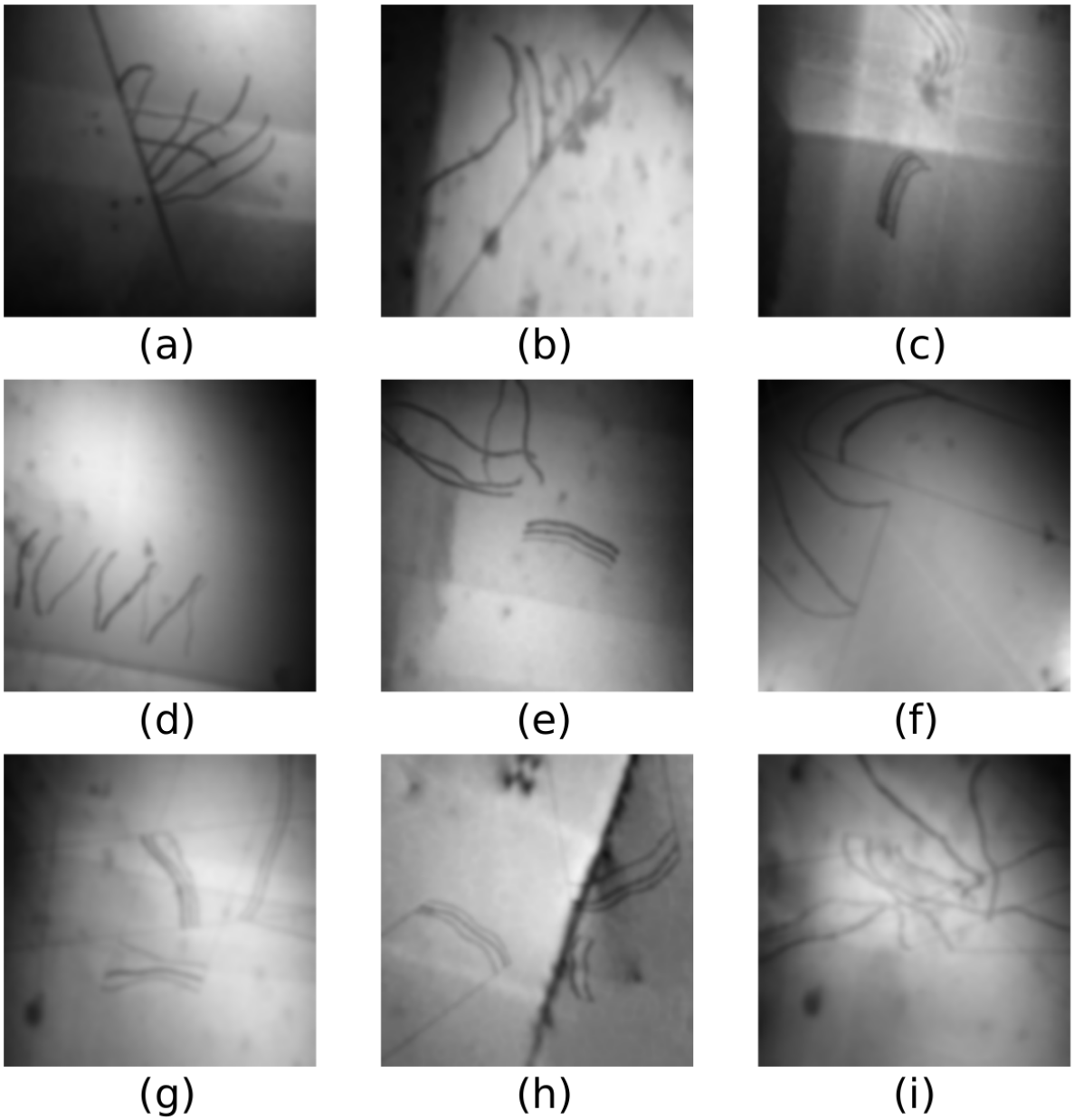


Figure 4.9: Sample synthetic images from synthetic dataset SD3. The dataset is very generalized and microstructure may consists of a number of pileups with grain boundaries, slip trace lines and over lapping dislocations.

known as solute pinning or Cottrell atmosphere. These impurities create a local stress field that can attract or repel dislocations. When a dislocation line interacts with these stress fields, it can bend or change direction. Dislocations can also interact with other types

of defects in the crystal, such as vacancies (missing atoms) and interstitial (extra atoms). These interactions can lead to complex changes in the dislocation shape, as the dislocation line may bend or curve to accommodate or bypass these defects. Hence it is important to include a large variety of dislocation shapes so that the trained machine learning model is robust and can work for any dislocation shape. We meretriciously hand-labeled 70 dislocations of different shapes, some of which are shown in Figure 2.6. Depending on the shape complexity, the number of support points can vary allowing us to carefully model such dislocation shapes. We have included such a large variety of dislocation shapes in synthetic dataset SD3 as shown in Figure 4.9.

3. In the synthetic datasets SD1 and SD2, the focus was predominantly on dislocations within the synthetic microstructures. However, this approach presents a limitation in that it overlooks the presence of other potential defects or artifacts that commonly occur in real TEM images. These additional elements can be easily misinterpreted as dislocations, thereby posing challenges in accurate image analysis. Two notable examples of such artifacts are grain boundaries (or twin boundaries) and slip trace lines. A grain boundary typically manifests as a distinct straight line traversing the entirety of an image. Its interaction with dislocations is of particular interest; dislocations may traverse a grain boundary or, alternatively, accumulate near it due to its impediment effect. On the other hand, slip trace lines, which can closely resemble dislocations, also pose a risk of incorrect identification. These slip trace lines might be erroneously classified as extended dislocations by a machine learning model not trained to differentiate between these two structures. To address these challenges and enhance the representative of the synthetic datasets, we have expanded the scope of dataset SD3. This dataset includes synthetic images that not only feature dislocations but also incorporate grain boundaries and slip trace lines. The inclusion of these structures was executed in a randomized manner, thereby ensuring a diverse and comprehensive range of microstructures within the dataset. By integrating these additional elements into the synthetic dataset SD3, we aim to improve the generalization and accuracy of machine learning models trained on this data. This enhancement allows models to better distinguish between actual dislocations and similar-looking artifacts, such as grain boundaries and slip trace lines, in TEM images. As a result, the models are expected to perform more reliably on real TEM images, where such diverse microstructure elements are frequently encountered.

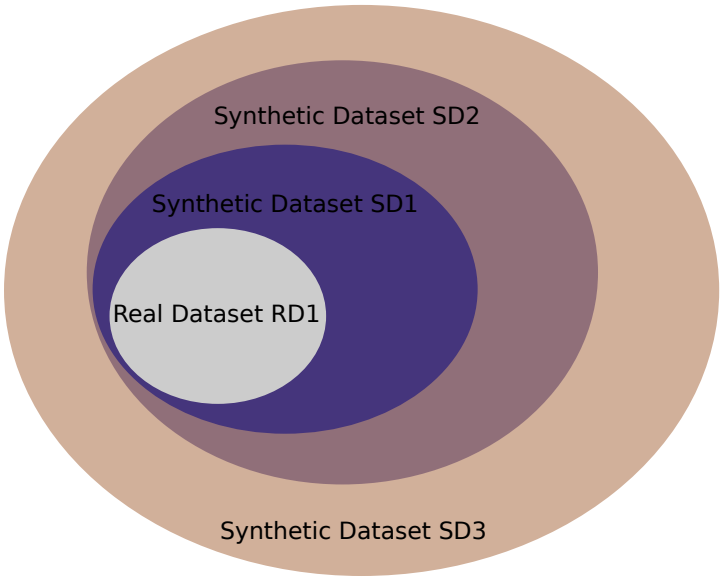


Figure 4.10: Synthetic dataset SD1 closely represent the real dataset RD1. We have more general dataset SD2 where we only take the dislocation shape information from real dataset. Synthetic dataset SD3 is the most generalized version where we use a more generalized range of parameters to generate the synthetic dataset.

4. The synthetic datasets SD1 and SD2 were developed with a specific structural constraint: the designed dislocation microstructures precluded any instances of dislocations intersecting or overlapping each other. This constraint reflects an idealized scenario, often diverging from the complexities encountered in actual TEM images. In TEM imaging, the resultant 2D representations are a product of electron beam interactions with the 3D microstructure of the material. This interaction can lead to apparent overlaps or intersections of dislocations in the 2D plane, despite these dislocations being spatially separated in the actual 3D structure. A representative case is seen in random dislocation microstructures, where multiple dislocation lines appear to overlap or intersect each other. These instances are not artifacts; rather, they are genuine representations of the spatial arrangement and projection effects within the material. The exclusion of such configurations in SD1 and SD2 limits the representational diversity of these datasets and potentially reduces the generalization of machine learning models trained on them. To address this limitation, the synthetic dataset SD3 introduces a critical modification: relaxing the aforementioned structural constraint. By allowing for overlaps and intersections of dislocation lines within the generated microstructures, SD3 better replicates the complexity and variability inherent in real TEM images. The inclusion of complex dislocation configurations in SD3 presents a more challenging and realistic dataset for machine learning training.

The three synthetic datasets presented above vary in complexity as shown in Figure 4.10 which illustrates the hierarchical structure, organized by complexity. The real dataset RD1 serves as the baseline for dislocation microstructure features from a single experiment. Moving outward, each synthetic dataset (SD1, SD2, and SD3) progressively adds complexity and diversity to the dislocation structures. SD1 closely resembles RD1, containing basic dislocation pairs and minimal variability to closely mirror real observations. SD2 introduces additional variations, increasing the range of microstructural features to enhance model generalization. SD3, representing the outermost circle, includes the highest level of complexity with rare and intricate dislocation configurations that may not be commonly observed in RD1. In the following two sections we will use these synthetic datasets to perform machine learning studies.

4.4 Study 1: Multi-label Segmentation Using Dataset SD1 and SD2

In this first study, we use the multi-label segmentation approach which is designed to predict individual dislocations as distinct masks. Each mask represents a single dislocation, ensuring that all pixels within that mask correspond to the same dislocation. Training a machine learning model for segmentation tasks can be challenging, in particular when dislocations are nearly touching each other. Through this approach, we make a comparison of synthetic datasets SD1 and SD2.

4.4.1 Training Details

We perform training on synthetic datasets SD1 and SD2 with 4000 training images and 1000 images for testing. For both the datasets, we use the two background generation methods. The most suitable parameters for the Perlin noise were experimentally obtained and are presented, along with their values in Table 4.1, resulting in altogether four synthetic datasets.

During the training phase, we use the early stopping strategy[2] by saving the model that attains the highest score based on the metric applied to the test data. This approach is a well-established strategy in machine learning, aimed at preventing overfitting, which occurs when a model learns the noise and details of the training data instead of the underlying patterns. By focusing on the model's performance on the test set, we enhance the model's ability to generalize, or perform well on previously unseen data. This not only boosts the overall effectiveness of the model but also contributes to its robustness in practical applications [118].

[2]Early stopping is a regularization technique where training is stopped when the performance on a validation set stops improving, helping to prevent overfitting and save computational resources.

ID	variable name	value range
1	O_1	10.0 . . . 100.0
2	O_2	10.0 . . . 100.0
3	w_{Perlin}	0.2 . . . 0.4
4	w_{white}	0.5 . . . 0.7
5	s_1	1.
6	r_{noise}	0.2 . . . 0.4
7	s_2	0.6 . . . 0.8

Table 4.1: Parameters and their values used for generating background images with Perlin noise.

To further enhance the generalization capability of our model, we employ various image transformation techniques throughout the training process. These methods include the introduction of Gaussian noise, as well as adjustments to brightness and contrast. Additionally, we apply image equalization techniques to improve the contrast in images. These transformations serve to diversify the texture properties of the synthetic images, enabling the model to learn from a wider range of visual inputs. By exposing the model to these variations, we help it to become less sensitive to specific features of the training data that might not be present in real-world scenarios.

The incorporation of these image transformation methods is critical in improving the model's robustness. By simulating different conditions under which images may be encountered, the model learns to recognize patterns and features that are invariant to such changes. This leads to a more adaptable model that is better equipped to handle the complexities and variations found in real images. Ultimately, this strategic approach not only facilitates better performance on synthetic datasets but also paves the way for successful application in real-world tasks, where variability is the norm.

4.4.2 Results and Discussion

The training and test loss curve for the models that were trained on the four synthetic datasets are shown in Figure 4.11. Both the training and test losses decrease during the initial training phase. This is a typical behavior for well-trained models, as the optimization process should lead to a reduction in the loss values during the training process. The loss curves eventually reach a saturation level, at which point the loss of the test curve is lower than that of the training curve and training eventually stops when early stopping is triggered. This observation was consistent across all four synthetic datasets, further strengthening the claim of good generalization on the test synthetic dataset. The diverse nature of the training data allows the models to learn the

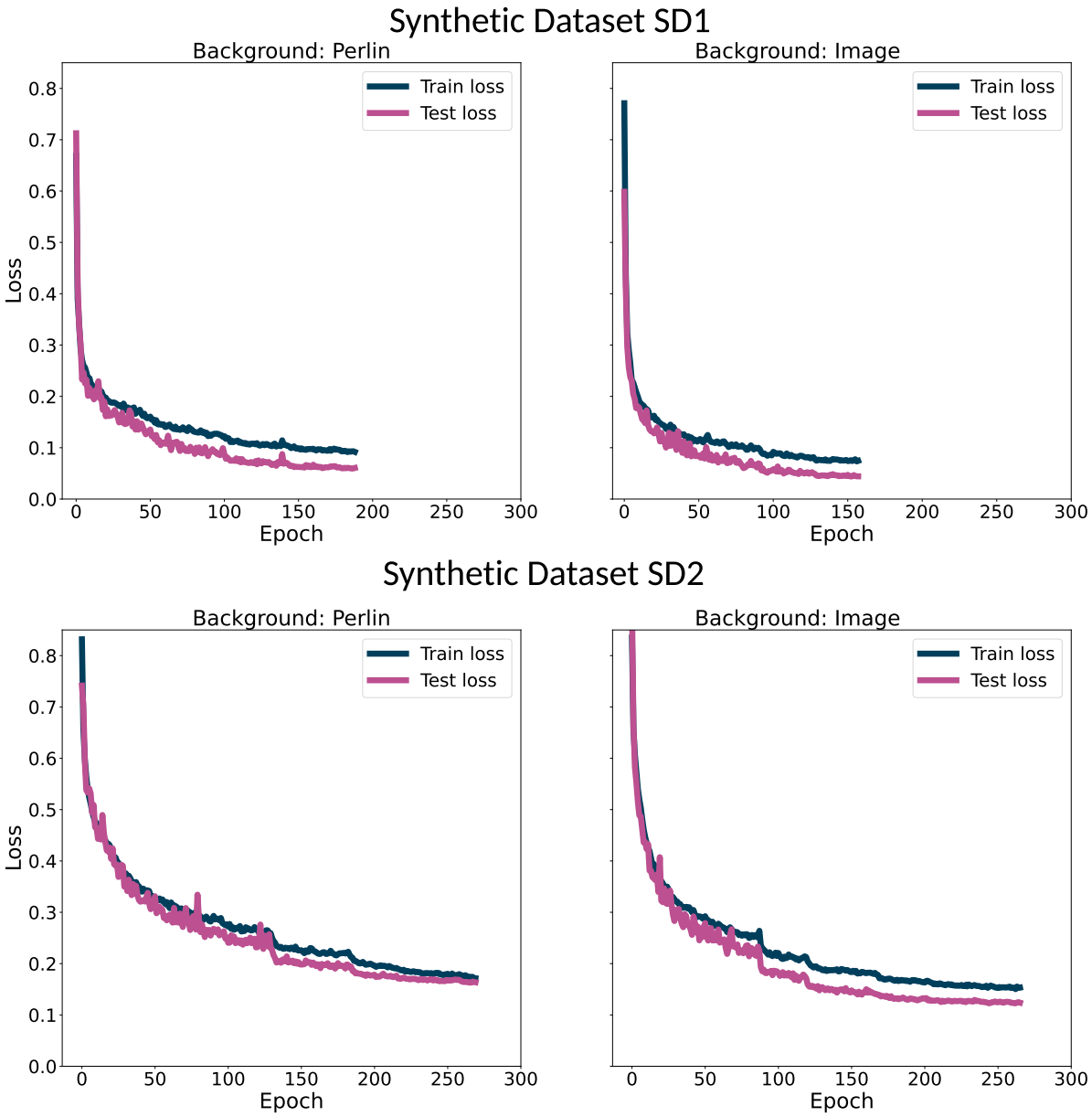


Figure 4.11: Top row shows the loss curves for synthetic dataset SD1 for the two background variations (based on Perlin noise and background image patches from real images) and bottom row for synthetic dataset SD2.

underlying patterns and structure in the images, which in turn makes them more robust and adaptable to variations in the test images. Comparing the loss curves for the two datasets SD1 and SD2, we find that the ML model optimized more easily on microstructure SD1. There, we were able to obtain a test loss even lower than ≈ 0.1. Training on the dataset with the general microstructure, SD2, is more difficult and requires more epochs. To compare the performance of each model trained on the four different synthetic datasets, testing dataset of 2000 images for each of the four datasets were generated and evaluated using the metric as measure. This results in a total of 16 combinations.

- Model trained on **SD1 (Perlin background)**: tested on SD1 (Perlin background), SD1 (Images background), SD2 (Perlin background), and SD2 (Images background),
- Model trained on **SD1 (Images background)**: tested on SD1 (Perlin background), SD1 (Images background), SD2 (Perlin background), and SD2 (Images background),
- Model trained on **SD2 (Perlin background)**: tested on SD1 (Perlin background), SD1 (Images background), SD2 (Perlin background), and SD2 (Images background),
- Model trained on **SD2 (Images background)**: tested on SD1 (Perlin background), SD1 (Images background), SD2 (Perlin background), and SD2 (Images background),

This systematic evaluation enables a comprehensive comparison of the models' performance across diverse dataset conditions. The results are summarized in the left panel of Figure 4.12 which shows the metric score for the models performance.

We now take a look at those combinations, where the same image generation approach was used for the training data as well as for the evaluation of the metric (the framed diagonal values). There, the models trained on synthetic datasets with microstructure based on SD1, result in significantly higher metric values than that of SD2 (> 0.91 for SD1 vs ≤ 0.72 for SD2). Furthermore, models trained on synthetic images with Perlin noise as background do not generalize well to dataset where realistic background images were used, e.g., the model trained on microstructure SD2 with Perlin noise results in a very low score of ≤ 0.22 on data with background images. ML models trained on realistic background images generally perform better: in particular after training with dataset SD2 with realistic background we obtained metric values > 0.6 for all datasets. Thus, using realistic backgrounds is a better choice than using purely synthetic background (keeping in mind that "realistic" means the superposition of *two* real microscopy background images which, from a domain scientist's perspective may look unrealistic).

The metric scores on the right panel of Figure 4.12 represent average scores over images with different numbers of dislocations. To see how the score depends on that, the distribution of the metric score as a function of the number of total dislocations in an image is shown in the box plot in the right panel of Figure 4.12. Achieving a very high metric score of upto 0.9 is feasible even for images containing up to 13 dislocations. As expected, the predictions for images with a

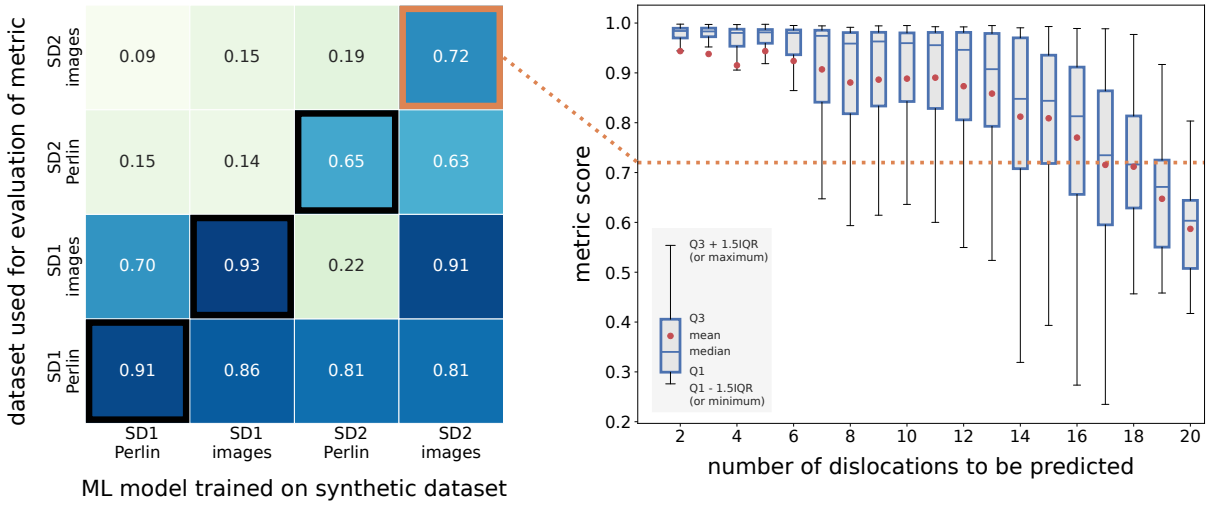


Figure 4.12: Left panel: metric scores for the four models, evaluated using four synthetic datasets. Right panel: Distribution of metric scores of the model trained on dataset with microstructure SD2 with realistic background as a function of the number of dislocations present in the images. The top right value from the left panel (highlighted by the orange frame) represents an average of the right panel.

larger number of dislocations becomes more challenging; specifically, for images with 14 or more dislocations, the average metric scores tend to decrease, accompanied by a greater variability. This decline is not solely attributed to poor segmentation resulting in undetected dislocations; it also results from the inherent difficulty in accurately predicting a single dislocation within a mask.

For brevity, we have excluded empty masks and those containing only a minimal number of "dislocation pixels" or other artifacts that can be readily removed. However, these artifacts remain visible in the aggregate output of all 20 masks. This phenomenon is illustrated clearly in the predictions made on a synthetic image from the test dataset, which comprises 20 dislocations, as shown in Figure 4.13. As a general rule, when multiple dislocations are predicted within a single mask, the corresponding metric value significantly decreases. This aspect has been discussed in detail in Section 3.4.3. Nevertheless, by aggregating predictions from all masks, it is possible to obtain a single mask that encapsulates all the dislocations identified in the image.

Interestingly, when the dislocations from all masks are combined, we observe a marked improvement in results. An alternative approach involves utilizing the comprehensive mask of all predictions and applying contour detection to isolate each individual dislocation. This post-processing step is particularly effective for microstructures where dislocations do not exhibit sub-pixel spacing, thus facilitating the separation of closely situated dislocations. Upon closer examination of the synthetic image presented in Figure 4.13, we observe that the dislocation

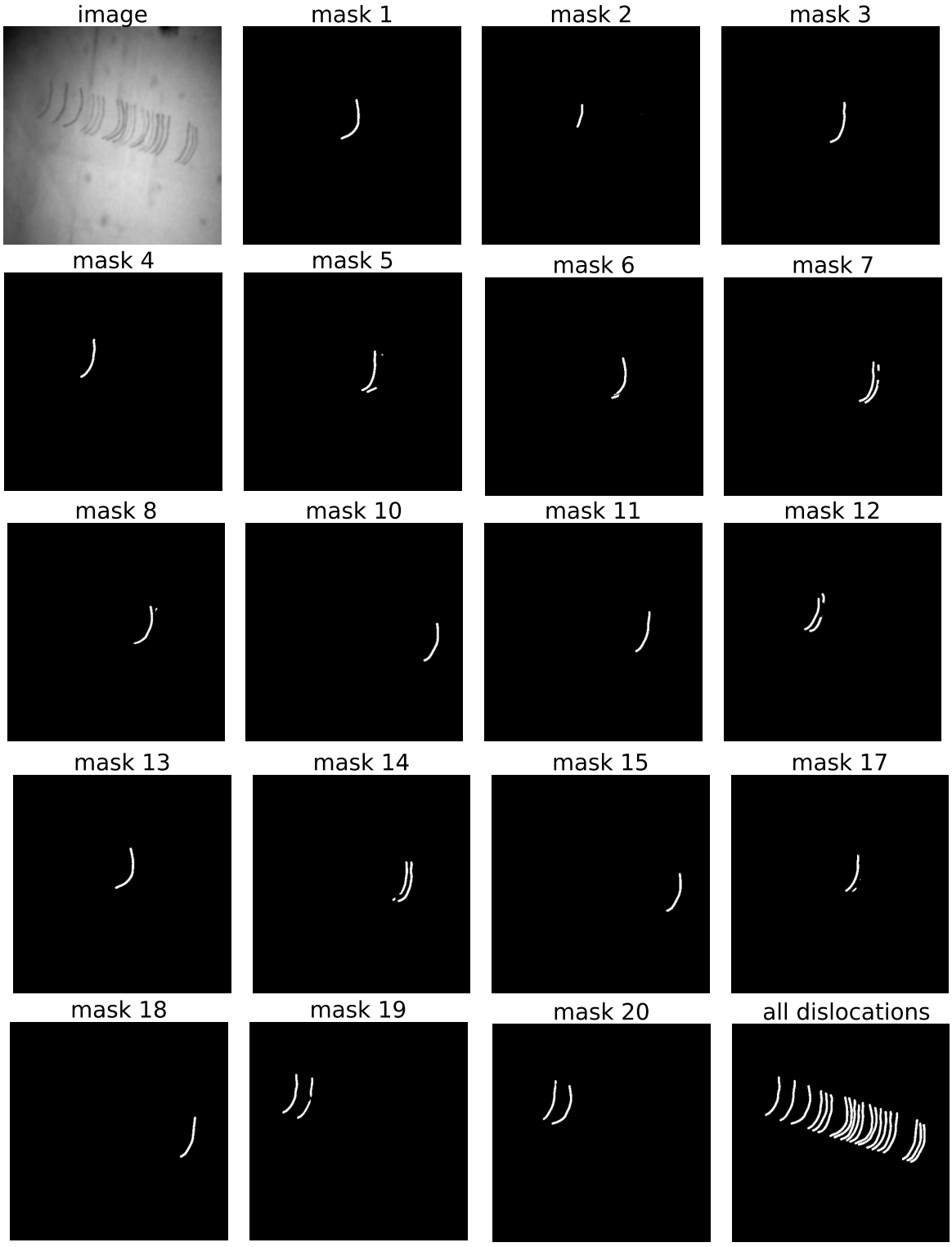


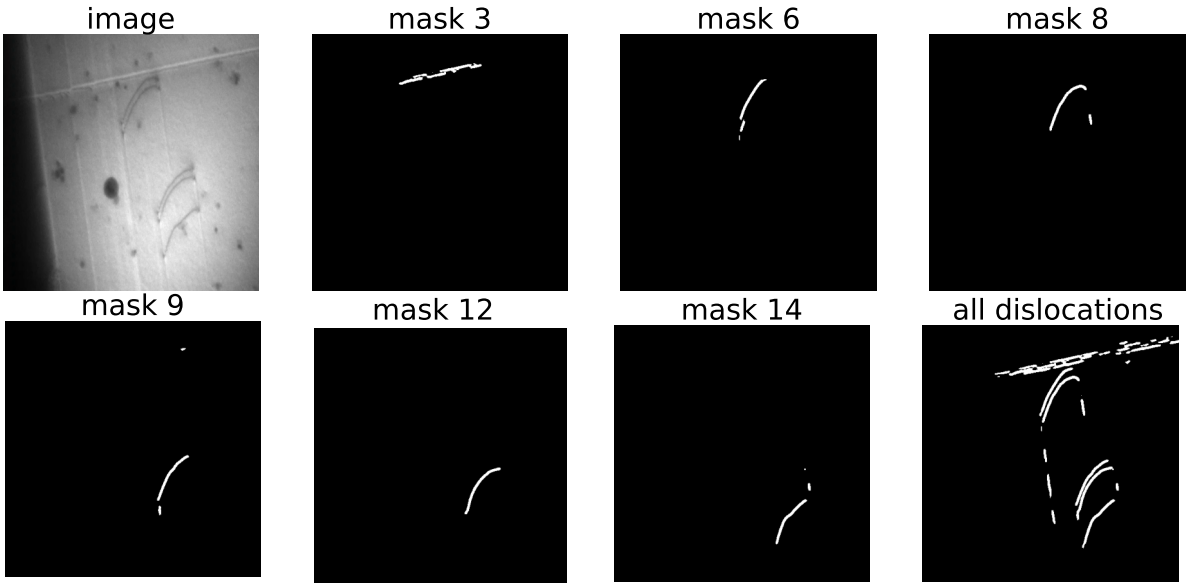
Figure 4.13: Visualization of dislocation segmentation predictions on a synthetic test image containing 20 dislocations, resulting in a low metric score of 0.43. The image illustrates challenges in accurately segmenting densely packed dislocations, highlighting the limitations of the multi-label segmentation approach in handling complex microstructures.

microstructure exhibits a pileup of very close dislocations, characterized by sub-pixel dislocation pixels. The multi-label segmentation approach used in this study successfully identifies single dislocation in separate masks, allowing for subsequent post-processing of each mask to represent dislocations as splines. This capability represents a significant advantage of the approach over traditional binary segmentation approaches.

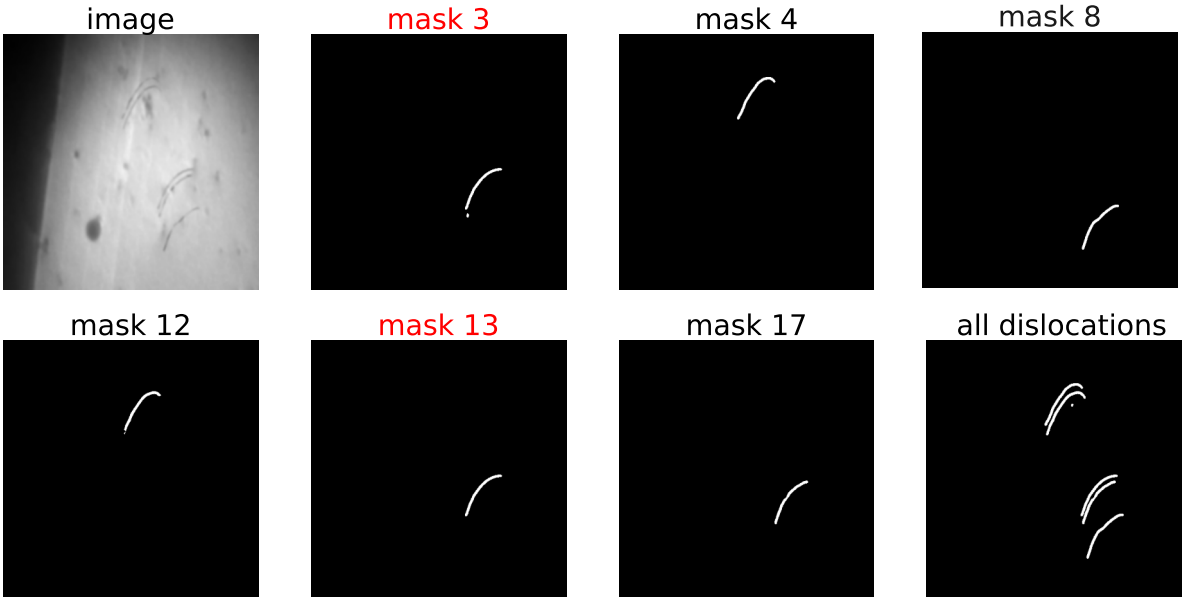
Additionally, we illustrate the model's predictions on a real image from the RD1 dataset, alongside its synthetic twin in Figure 4.14. Notably, the predictions for the synthetic twin demonstrate markedly improved accuracy compared to those for the real image. The image has five dislocations and it predicted all of them accurately in separate masks, facilitating the isolation of incorrect predictions. This is evident in the predictions for the real image, where line-like artifacts—potentially indicative of twin boundaries—are incorrectly predicted as dislocations and are found in a different mask, specifically shown as mask3 for the real image, separate from the actual dislocations. This feature can prove advantageous during the post-processing phase of the predictions, as well as for further analysis of results. For instance, dislocations typically migrate along specific slip directions, which can be leveraged to identify dislocations that contribute to the pileup. It is also noteworthy that certain segments of slip trace lines are misidentified as dislocations. This misclassification of such structures—specifically slip trace lines—as dislocations arises from the characteristics present in the synthetic dataset SD2. In this dataset, our focus was primarily on dislocations, and we excluded any such structures or artifacts that the machine learning model might struggle to differentiate from dislocations.

To enhance predictive performance, several strategies can be considered, including expanding the diversity of the training dataset, employing more sophisticated deep learning architectures, and performing hyperparameter optimization. Additionally, an area of improvement lies in the quality of synthetic images generated. While the dislocation microstructures in synthetic and real images exhibit a high degree of similarity, discrepancies in the background environments remain evident.

A thorough analysis of the pixel intensity distribution surrounding the dislocations reveals substantial differences between the synthetic and real images, as illustrated in Figure 4.15. Specifically, the intensity distribution of the synthetic images appears significantly smoother compared to that of the real images. This smoothing effect can be attributed to the application of a Gaussian filter, which is utilized to refine the drawn dislocation lines in synthetic images. The synthetic image generation used a simplified approach that do not fully capture realistic complexities, leading to an oversimplified intensity distribution.



(a) Predictions of dislocations from a real microscopy image sourced from dataset RD1.



(b) Predictions on the corresponding synthetic twin image.

Figure 4.14: Dislocation predictions on a real microscopy image from dataset RD1 (a) and its corresponding synthetic twin (b). The red labels highlight masks that predict the same dislocation. The synthetic twin demonstrates improved accuracy, with all five dislocations accurately predicted in separate masks, facilitating the isolation of incorrect predictions. In contrast, the real image exhibits line-like artifacts, potentially indicative of twin boundaries, incorrectly predicted as dislocations in a separate mask (mask3).

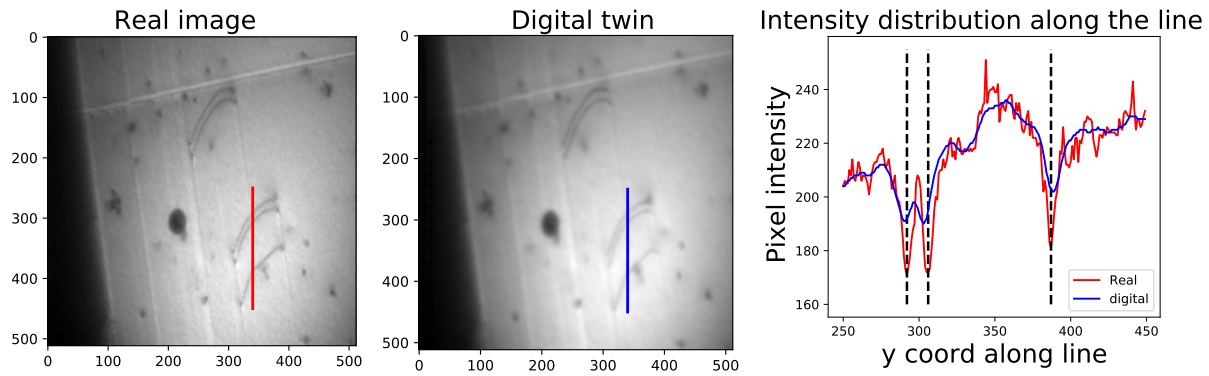


Figure 4.15: Comparison of pixel intensity distributions around a dislocation in a real microscopy image and its synthetic twin. The synthetic image exhibits a smoother intensity profile due to the application of a Gaussian filter during synthetic generation, which refines the drawn dislocation lines. In contrast, the real image shows greater variability in pixel intensities, reflecting more complex microstructural features and background environments.

4.4.3 Conclusion

This study addresses an important question: determining the most effective synthetic dataset for a given real dataset. Two primary approaches explored for generating synthetic datasets. The first approach, exemplified by synthetic dataset SD1, involves deriving parameter values from the real data (RD1) to create a synthetic dataset closely representing the real dataset's microstructural features. This method requires an in-depth analysis of the real dataset to accurately identify and replicate its defining parameters in the synthetic version. The second approach adopts a more generalized methodology, wherein the microstructure parameters for the synthetic dataset are largely independent of the specific characteristics of the real dataset, with the exception of a few parameters, such as dislocation shapes. This approach significantly simplifies the synthetic data generation process, as it eliminates the need for extensive analysis of the real dataset to extract microstructural parameters. As demonstrated in Figure 4.14, this generalized approach can also give satisfactory results when applied to real images comparable to what we would obtain using the specialized synthetic dataset SD1.

The application of the second approach are particularly pronounced in scenarios where access to detailed real data prior to experimentation is limited or unavailable. In such cases, the ability to generate a broadly applicable synthetic dataset becomes invaluable. This approach facilitates the creation of a general-purpose synthetic dataset, designed to be effectively utilized across a diverse range of real datasets. This versatility is crucial in advancing the field of deep learning applications in TEM image analysis, as it enables the development and refinement of models that are robust, adaptable, and capable of delivering high-quality results across various microstructural

scenarios.

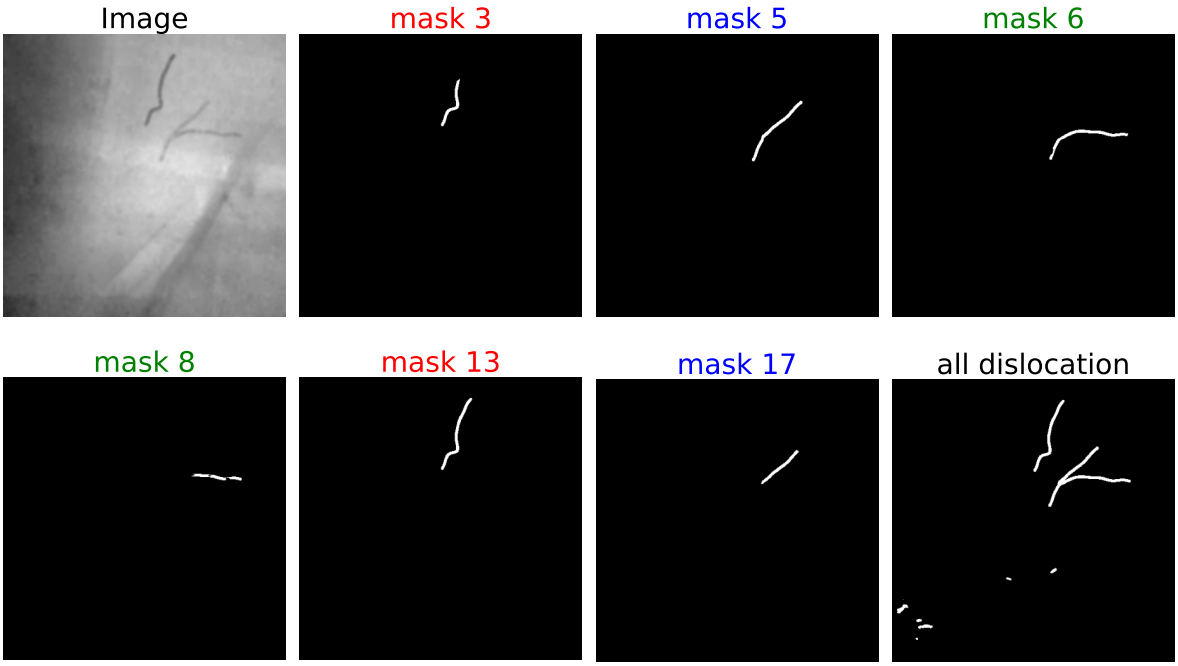
The observed disparity in segmentation quality between synthetic and real images, as highlighted in Figure 4.14, raises critical considerations in the development and training of machine learning models for TEM image analysis. Two primary factors contribute to the lower performance on real images compared to their synthetic counterparts:

1. **Pixel intensity distribution discrepancy:** As illustrated in Figure 4.15, a notable difference exists in the pixel intensity distribution surrounding dislocations between synthetic and real images. This discrepancy presents a significant challenge in accurately segmenting dislocations in real images. However, the implementation of various augmentation methods during training has proven to mitigate this issue to a certain extent. These augmentation techniques have enhanced the model's ability to generate accurate predictions for dislocations in real images, as evidenced by the segmentation results.

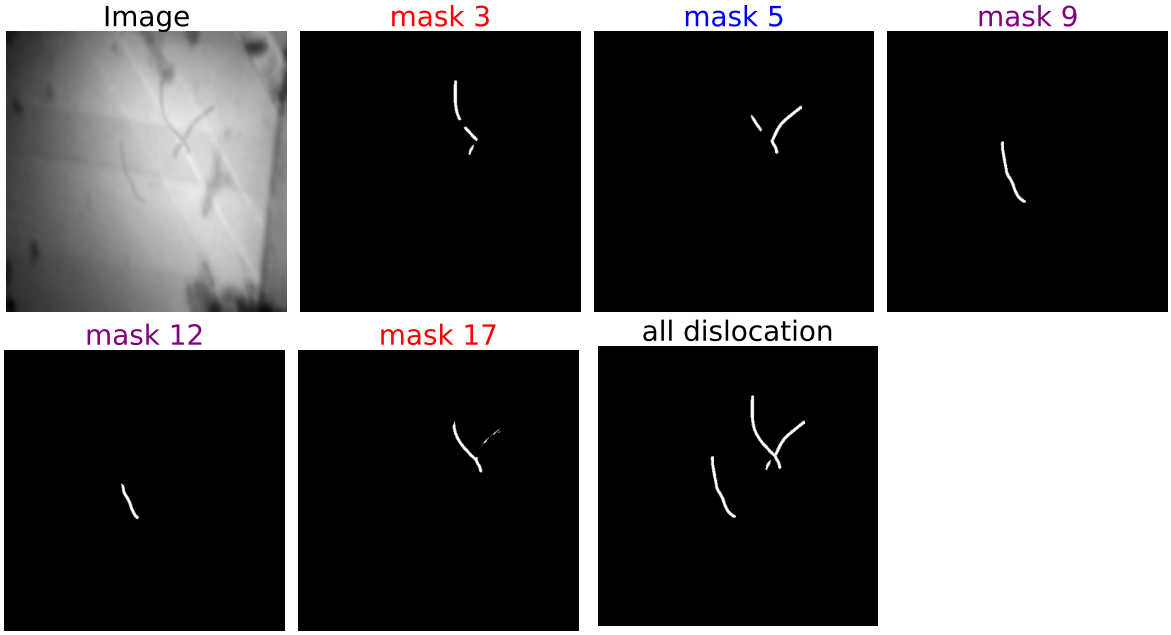
2. **Lack of artifacts in synthetic dataset SD2:** The synthetic dataset SD2, while primarily focused on dislocations, does not includes other microstructural artifacts that frequently appear in real TEM images. These artifacts, such as slip trace lines and twin boundaries, can be mistakenly identified as dislocations by the machine learning model. This misidentification is evident in the real image predictions shown in Figure 4.14, where such artifacts are incorrectly segmented as dislocations. The absence of these elements in SD2 limits the model's ability to distinguish between actual dislocations and other similar microstructural features.

To address these challenges and enhance the model's performance on real images, the development of the more generalized synthetic dataset SD3 was tried. SD3 incorporates a broader range of microstructural elements, including the aforementioned artifacts like slip trace lines and twin boundaries. The inclusion of these elements is intended to provide the machine learning model with a more comprehensive training dataset, thereby improving its ability to accurately segment and differentiate various features in real TEM images. This approach aims to bridge the gap between synthetic and real image analysis, fostering the development of more robust and effective segmentation models for TEM image analysis.

The multi-label segmentation method is found to be very useful in predicting dislocation structures with sub-pixel dislocation spacings but dislocation structures can be much more complex. One such complex structure can be dislocations with overlapping or intersecting dislocations. To test the applicability of the model on such dislocation structure, we generate two synthetic image as shown in Figure 4.16 where we have only 3 dislocations. In Figure 4.16(a) two of the dislocations are very close and touching to each other. The model was able to predict the 3 dislocations in individual masks. For one of the dislocation shown in mask 6, the predictions are slightly wrong as a part of the other dislocation is also predicted. The results are much worse for the other image Figure 4.16(b) where dislocations intersect each other. The model finds it difficult



(a)



(b)

Figure 4.16: Multi-label segmentation predictions on (a) a synthetic image with sub-pixel dislocation spacing .(b) synthetic image with intersecting dislocations. Masks representing same dislocations are color coded.

to correctly assign the pixels to the dislocations as can be seen in mask 5 where a part of other dislocation is incorrectly predicted. One of the reason for poor performance for the model on such complex images might be the training data provided. In the synthetic dataset SD2, we have microstructures with nearly touching dislocations but we do not have images with intersecting dislocations and hence the model might be failing to learn the features of such microstructures.

4.5 Study 2: Comparison of the 3 machine learning approaches using synthetic dataset, SD3

In Study 1, we explored the multi-label image segmentation approach in generating individual dislocation masks for each dislocation from images. However, other methods may offer unique advantages. Study 2 investigates a comprehensive evaluation of three distinct machine learning approaches: multi-label image segmentation, instance segmentation, and dislocation spline support point estimation, utilizing the versatile synthetic dataset SD3. This dataset, designed to mimic a wide array of real-world TEM scenarios, serves as an ideal dataset for this comparative study.

Some of the images from synthetic dataset SD3 are shown in Figure 4.9. As discussed before the dataset is much more general and contains grain boundaries, see Figure 4.9(a) and Figure 4.9(b) as well as slip trace lines, see Figure 4.9(f), Figure 4.9(h) and Figure 4.9(j). The incorrect identification of such structures as dislocations as seen in previous study for a real image in Figure 4.14 motivated us to include such structures in synthetic images to improve the model capabilities in obtaining good results on real data. The dataset not only includes simple dislocation structures Figure 4.9(c) and Figure 4.9(d) which contains dislocation pileups and dislocations do not intersect each other but also some very complex structures similar to shown in Figure 4.9(i) and Figure 4.9(j).

The synthetic dataset SD3, with its diverse range of dislocation sizes, shapes, and configurations, challenges each method's robustness and adaptability. By evaluating these methods on SD3, we aim to uncover their strengths and limitations in handling complex TEM image data of dislocation microstructure. There are two main challenges for a machine learning model to predict dislocations in this particular dataset. The first challenge comes from the diversity of the dislocation microstructure, which is further complicated by the presence of multiple dislocation pileups where dislocations may overlap as shown in Figure 4.9(i). The other is the subpixel dislocation spacing between dislocations which makes it challenging to predict the individual dislocations and challenges the models in accurately predicting such dislocations. The outcome of this study is expected to provide valuable insights into the most effective techniques for TEM image analysis, thereby guiding future research and application in this domain.

4.5.1 Training Data Requirements for Model Convergence

We generated a total of 5000 images for synthetic dataset SD3. This data is split into training and validation using a 80:20 split. During training, we use data augmentation methods as described in previous study. We evaluate the performance of the three approaches using a separate test data of 2000 images which was not part of the training using the physical metric proposed in this work.

We aim to first analyze how the model performance varies with different training data sizes and to determine the minimum data required for model convergence. The objective is to identify the data size at which increasing the dataset no longer yields significant improvements in performance, thereby indicating a state of data sufficiency.

This study is critical because understanding the relationship between dataset size and model performance helps optimize resource allocation in machine learning. Identifying the threshold for data sufficiency enables us to avoid both over-fitting and under-fitting, improve model generalization, and reduce computational costs. This is particularly important when working with limited datasets, such as our dislocation image data, where data is often scarce and hard to obtain. We use the multi-label segmentation approach to train the model using varying amounts of training data, as shown in Figure 4.17. After each training phase, we evaluate the model using the test data of 2000 images to assess its performance.

We begin with just 1% of the available training data, amounting to just 50 images. This represents a realistic scenario where access to only a few dozen images is available. As expected, the initial results are poor, with a mean metric score of only 0.05. Increasing the data size by a factor of five does not lead to a substantial improvement in performance. However, after progressively adding more data, we begin to observe better results. When 70% of the training data is used, the model achieves 100% accuracy on some images. Nonetheless, it is important to note that even at this stage, the model yields a score of 0 for some images, indicating that its performance still fluctuates significantly. This inconsistency, where the model can predict entirely incorrect results, is not ideal for an image segmentation task.

When the full training dataset is used, the model achieves its best performance. However, there is still room for improvement, as some images continue to exhibit suboptimal predictions. Additionally, increasing the dataset further requires significantly more computational resources and training time. For subsequent experiments, a balance is struck by selecting 5,000 images for model training, which offers a compromise between performance and resource efficiency.

This small study shows that using only a few tens of images to train the model is not sufficient and we require images in thousands to get good quality results. This is the main motivation behind the idea of using synthetic data generation model which can allow us to generate large amount of training data which is usually not available.

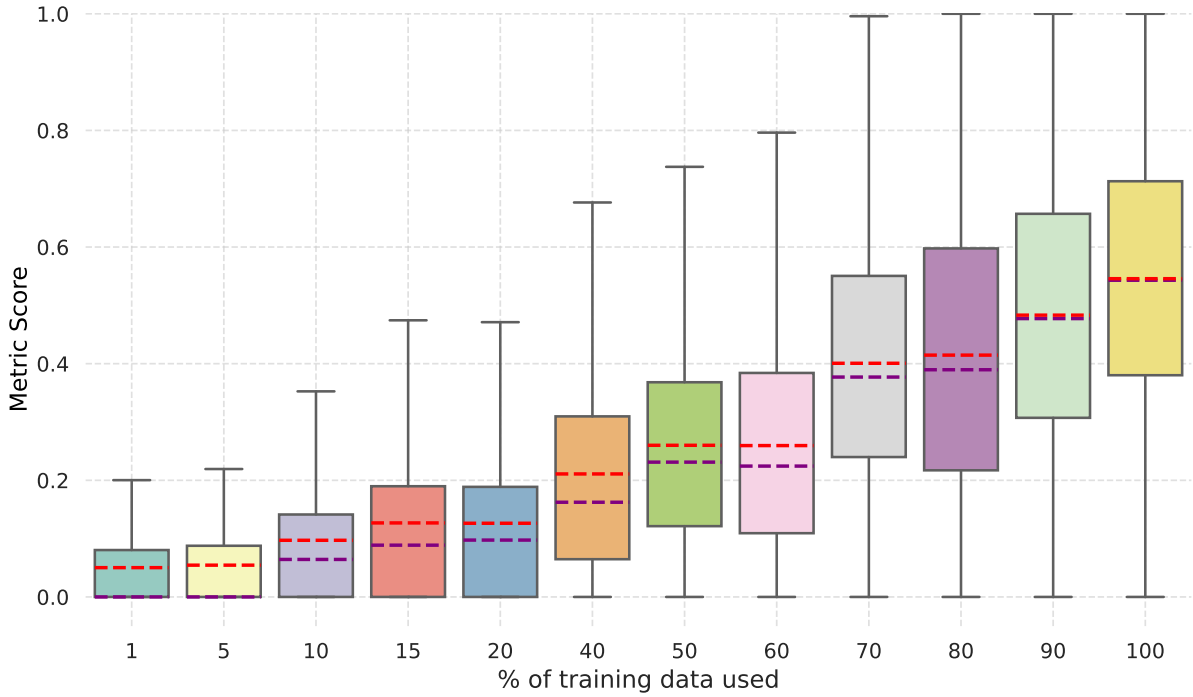


Figure 4.17: Model performance for varying amount of training data. The red line represents the mean metric score, indicating the average prediction accuracy across images, while the purple line represents the median score. the difference between them is a measure of the skewness of the distribution.

4.5.2 Comparison of Machine Learning Approaches

The Mask R-CNN model does not inherently produce segmentation masks at the same resolution as the original input image. Instead, the segmentation head of the model generates outputs at a lower resolution, determined by a predefined output size parameter. During inference, these outputs are upscaled to match the input image size, generating the final segmentation masks. In this study, where accurately representing dislocations as spline contours is essential, we investigated the impact of varying the output size of the segmentation head.

We trained two model variations for the instance segmentation task, setting the segmentation head output sizes to 28 and 56 pixels, respectively. This adjustment affects the resolution of the feature map produced for each RoI, without altering the overall architecture or the number of model parameters. Although the number of layers remains constant, increasing the output size directly influences both computational complexity and memory demands. A larger output size necessitates processing larger feature maps in subsequent layers, requiring additional memory for storage and increased computational resources.

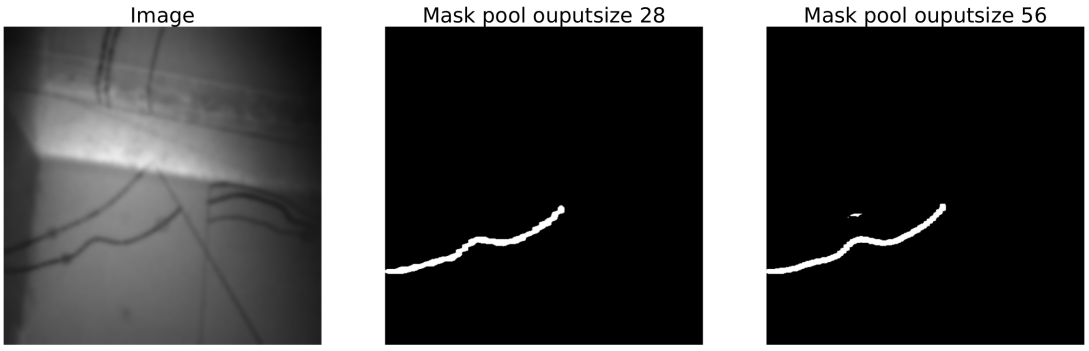


Figure 4.18: Comparison of segmentation masks for a dislocation using Mask R-CNN models with segmentation head output sizes of 28 and 56 pixels.

One advantage of a larger output size is an improvement in segmentation accuracy, at computational cost. As illustrated in Figure 4.18, the instance segmentation masks for dislocations are visually compared between the two output sizes. The model with the larger output size yields smoother and more accurate segmentation contours, minimizing the artifacts introduced by upscaling. However, qualitative analysis suggests that predictions from the model with the larger output size occasionally exhibit inaccuracies, such as the erroneous segmentation of unrelated dislocation instances. This observation requires further investigation to quantify the influence of output size on segmentation precision and instance miss-classification.

4.5.3 Results and Discussion

We train two variations of Mask R-CNN models each of output size 28 and 56 in segmentation head for instance segmentation approach. The metric score averaged over all the test of these the models is shown in Table 4.2.

On comparing the Mask R-CNN approach for the two output mask size, we find that using a lower output size is significantly more useful. As shown above, even though we can get much better dislocation masks at pixel level which can accurately capture the morphology of the dislocations using a high output size, we tend to get poor results in terms of detecting all the dislocations present in an image. Even when we use a default output size of 28, we were able to obtain a metric score of only 0.397 which is significantly lower compared to the perfect score of 1. On comparing the instance segmentation approach with the other 2 approaches, we find that the instance segmentation approach does not perform so well. We can conclude that the multi-label segmentation is the best approach. It is interesting to note that spline support point detection is much better compared to instance segmentation.

To thoroughly evaluate model performance on synthetic datasets, we divided the test dataset

ML approach	**Metric score**
Mask R-CNN, output size 28	0.397
Mask R-CNN, output size 56	0.209
Spline support point	0.503
Multi-label segmentation	0.555

Table 4.2: Average metric scores on the test dataset for different machine learning approaches applied to dislocation microstructures. The results indicate that multi-label segmentation achieves the highest overall performance, followed by spline support point detection. Instance segmentation using Mask R-CNN shows improved results with a lower output size (28) compared to a higher output size (56), highlighting the trade-off between mask resolution and detection of all dislocations in an image.

into two distinct categories: dislocation microstructures with overlapping dislocations and those without. We then assessed the metric scores for four different methods on these two sub-categories. The results, presented in Figure 4.19, show the mean metric score for each method across both categories. This enables a clearer understanding of each method's effectiveness in handling varying dislocation microstructures.

Our analysis reveals that for instance segmentation and spline support point detection, performance is consistent across both categories, indicating that these methods are unbiased and produce comparable results regardless of dislocation overlap. However, the multi-label segmentation approach exhibits a slight preference towards non-overlapping dislocation structures, achieving a mean metric score of 0.59 on non-overlapping structures compared to 0.49 on overlapping structures. Notably, the higher mean metric score observed for the multi-label segmentation approach, as summarized in Table 4.2, is largely due to its superior performance on non-overlapping dislocation structures, where it achieves higher accuracy in segmentation.

The top two approaches to use are spline support point detection and the multi-label segmentation. To look more closely how the models are performing on the images from the two sub-categories, we choose an image from each sub-dataset and use the two models to predict the support points and the masks. The image with non overlapping dislocation microstructure consists of three dislocation pileups with a total of seven dislocations is shown in Figure 4.20. We can see clearly in the image that the image tests the models ability to predict dislocations at sub-pixel spacings and hence the model might consider the two nearby dislocations as one. Even a person has to closely see the structure to understand that there are indeed two dislocations not just one. The spline support detection method seems to fail at this task. The model predicts only five dislocations out of seven dislocations. There are two sets of dislocations which have sub-pixel spacings and the model predicted only one of the two dislocations hence, there are two dislocations the model was not able to predict. Predicted support points for dislocation labeled as four is rather interesting where the model predicted some of the support points on another nearby

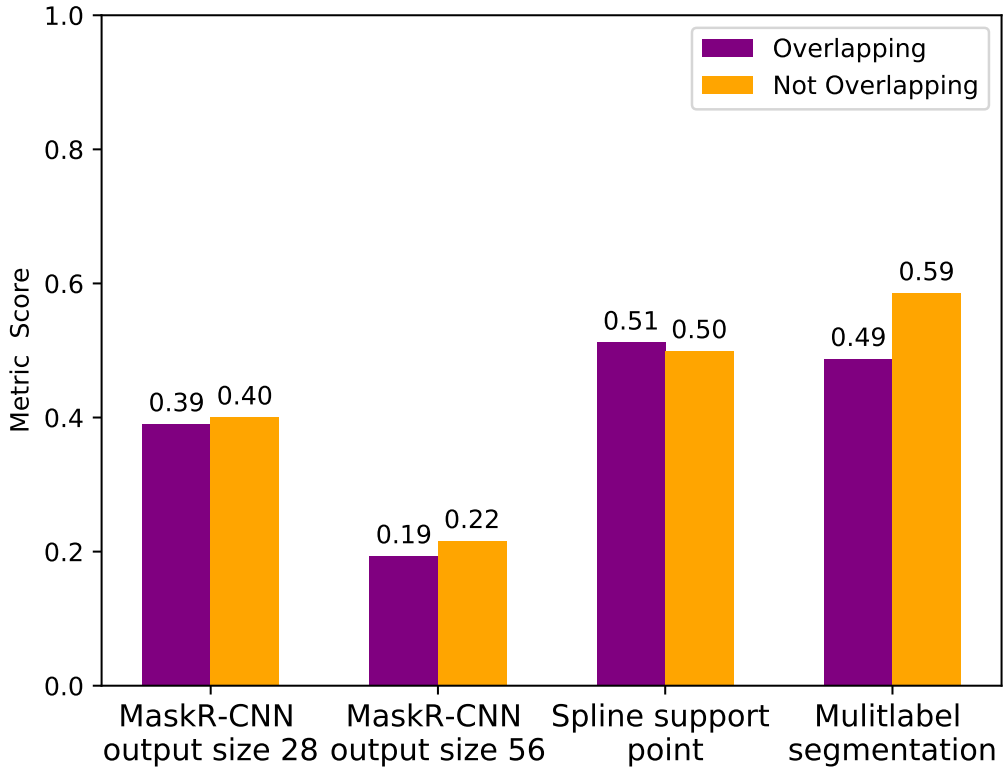


Figure 4.19: Comparison of metric scores for different machine learning approaches on synthetic dislocation microstructures with and without overlapping dislocations. The instance segmentation and spline support point detection maintain consistent performance across both categories, while multi-label segmentation performs slightly better on non-overlapping structures. These results highlight each method's capability to handle varying microstructural complexities, including sub-pixel spacing and overlapping dislocations.

dislocation. The failure of the model to resolve dislocations at sub-pixel spacing can be due to a number of reasons. Mask R-CNN relies on the features extracted by its backbone network.

If the spatial resolution of these features is insufficient to capture the subtleties of closely spaced keypoints, the model may fail to distinguish between them. This limitation is particularly relevant for sub-pixel spacings, where the distinct keypoints may not be adequately represented in the extracted feature maps. The prediction of support points as one-hot encoded masks requires the model to allocate a separate channel for each dislocations. The use of NMS and thresholding for refining support points predictions can further complicate the detection of closely spaced dislocations. NMS, designed to eliminate redundant detections by keeping only the strongest prediction in a vicinity, might mistakenly discard valid keypoints as redundancies when they are too close. In our work, we have tried with lower values of the thresholds and obtained similar

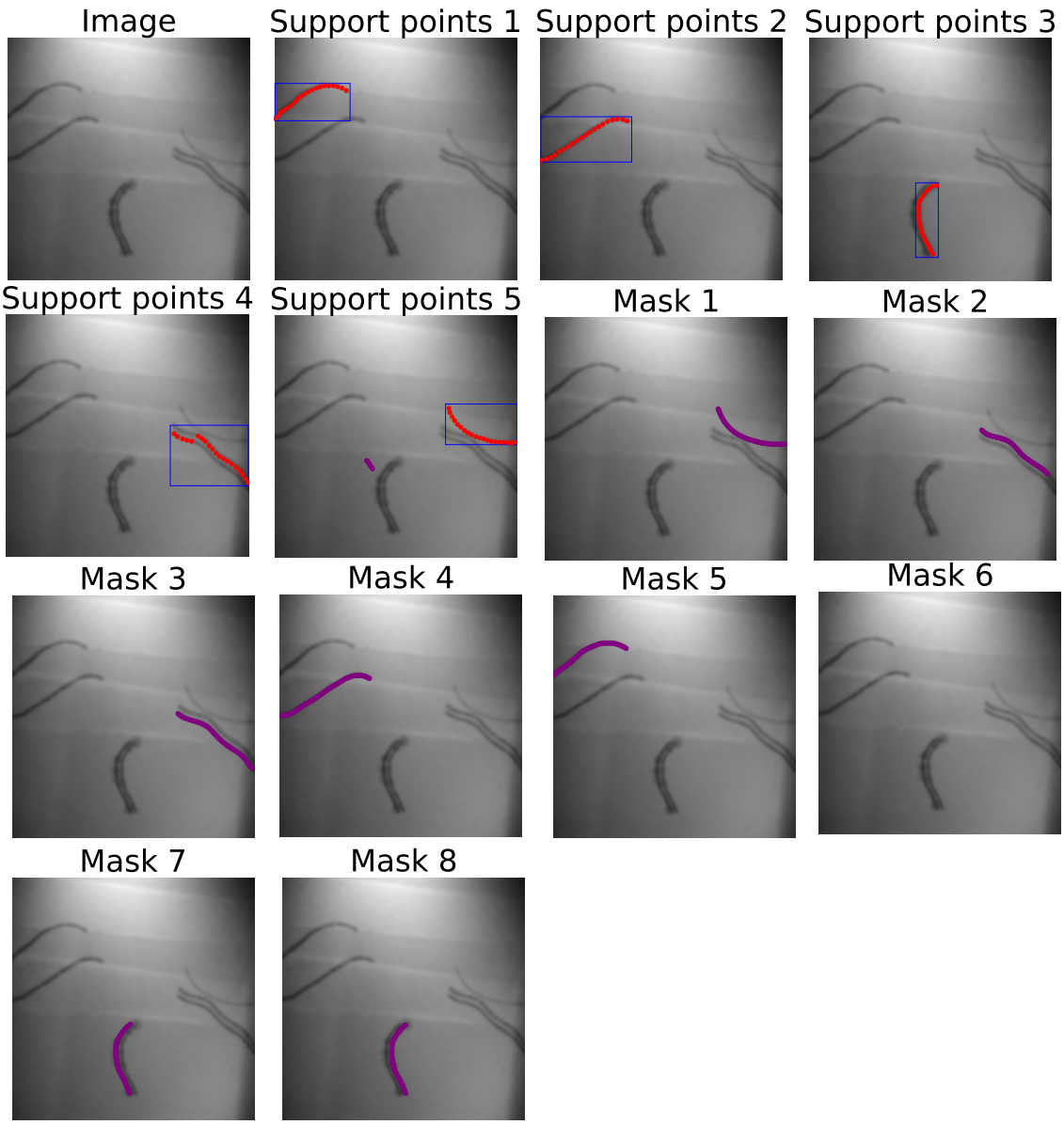


Figure 4.20: Comparison of predictions from spline support point detection and multi-label segmentation on a synthetic image with very closely spaced dislocation microstructure. The image contains seven dislocations across three pileups. The multi-label segmentation approach accurately identifies each dislocation in separate masks, while the spline support point detection method fails to resolve closely spaced dislocations at sub-pixel resolution, predicting only five out of seven dislocations. This highlights the challenges in detecting dislocations with very small inter-dislocation spacing.

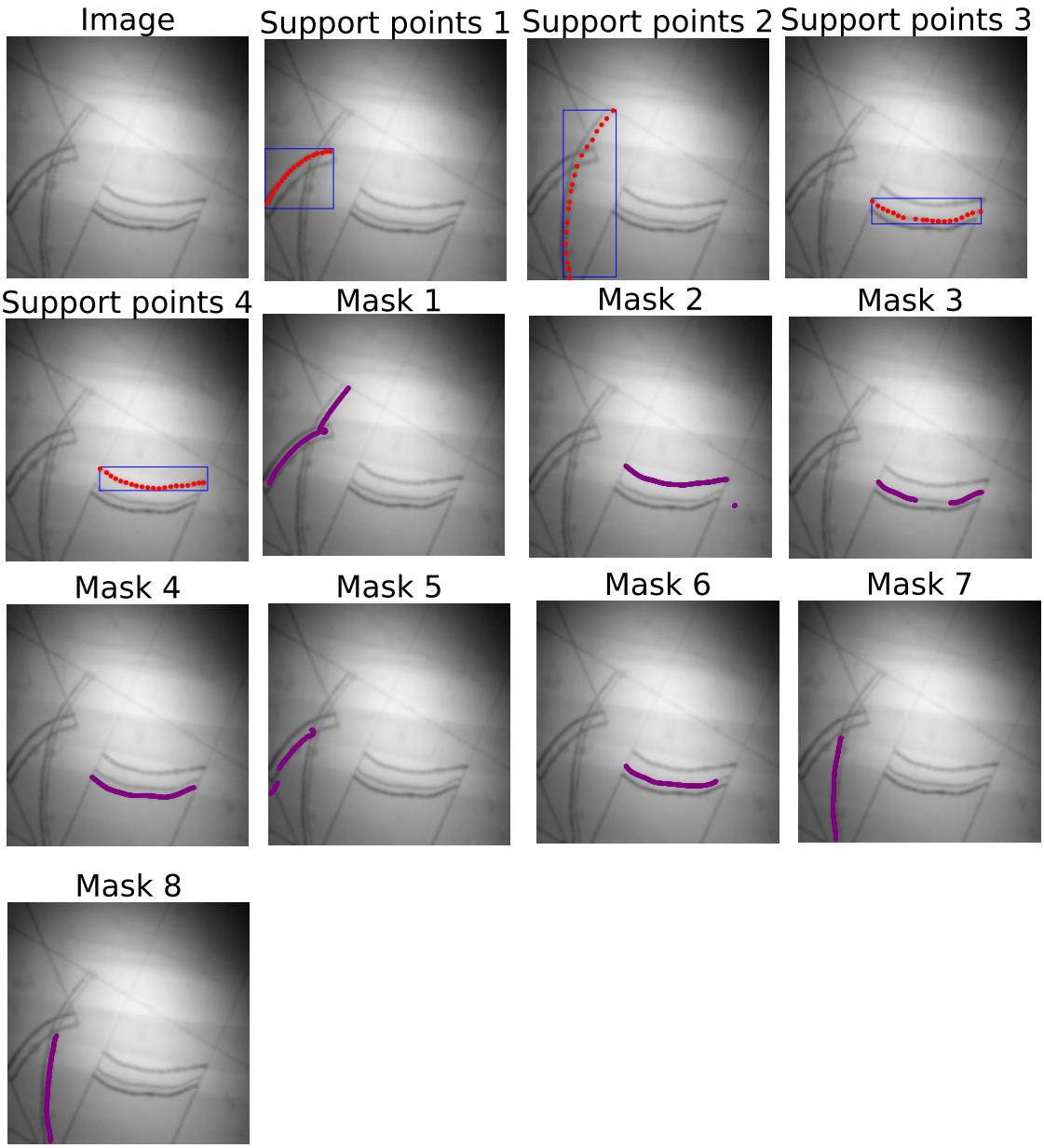


Figure 4.21: Prediction results from spline support point detection and multi-label segmentation on a synthetic image containing overlapping dislocations and slip trace lines. Both models successfully distinguish dislocations from slip trace lines. The spline support point model partially resolves overlapping dislocations but misses some closely spaced ones, predicting four out of seven dislocations. In contrast, multi-label segmentation predicts segments of overlapping dislocations, sometimes merging parts of different dislocations into a single mask, illustrating its limitations for highly overlapping structures.

results which suggests that the above two are the main reasons why the model failed to predict the nearby dislocations accurately.

We also show predictions on another image shown in Figure 4.21 which contains overlapping dislocations in the microstructure as well as slip trace lines. We can see that both models were able to successfully distinguish the dislocations from the slip trace lines. The spline support point model does not perform so well for very close dislocations and predicts only one of them and hence it predicts only four dislocations out of seven. If we focus on the overlapping dislocations, we can see clearly that the model was able to correctly predict the support points on the two overlapping dislocations as shown in support point prediction for dislocation 1. This is not the case for multi-label segmentation. For dislocations which are overlapping, the model predicts only a segment of the dislocations which do not overlap as can be seen for dislocations in mask labeled as 7 and 8. Prediction shown in mask 1 is interesting where the model predicts the segments of two different dislocations. We make similar observations on other such images also where the multi-label segmentation does not perform well for the overlapping dislocation microstructures.

4.5.4 Conclusion

In the study, we find that the instance segmentation approach is the worst approach not only in terms of accuracy measured using the metric based on dislocation length but also it does not capture the morphology of the dislocations accurately. The best choices are support point detection and multi-label segmentation but the choice of the method depends of the dislocation microstructure of interest. The support point detection method does not perform well for dislocations with sub-pixel spacing and tends to predict only one of the two dislocations but is a good choice dislocations overlapping with other dislocations in the microstructure. The multi-label segmentation approach provides very good results and can successfully segment the sub-pixel spaced dislocations but does not work so well for overlapping dislocations. The results tend to be very poor using the multi-label segmentation approach where it predicts only a part of the overlapped dislocation not the complete dislocations. These results can also be used as a recommendation which method to use, once it is known which dislocation structure is (approximately) to be expected.

Chapter 5 Study Using Synthetic and Real Data: Transfer Learning Approach

In the previous chapter we focused on synthetic data generation and comparison of the three machine learning approaches. One of the central question of interest is the generalization of the machine learning methods to new unseen datasets. In this chapter we focus on this aspect and introduce real dataset used in further studies in Section 5.1 followed by machine learning studies in Section 5.2 where we train one model only on real dataset from scratch and use transfer learning approach for the second. To benchmark the two models, we prepared a test dataset from experimental data which was not accessible during the training process. This allow us to evaluate the generalization capabilities of the two approaches.

5.1 Real dataset

Generating real TEM data of dislocation microstructures requires specialized expertise, sample preparation, and access to high-quality experimental facilities. Conducting a single TEM experiment typically involves weeks of planning and careful preparation. During these experiments, researchers scan various regions of the specimen and capture the evolution of dislocation microstructures within regions of interest in video format.

The TEM videos obtained are often as long as an hour, during which the focus may shift across different regions of the sample. To isolate consistent observations of a single microstructure region, we performed a comprehensive analysis of each experimental video, segmenting and trimming sequences where the electron microscope remains focused on the same region. This yielded a set of 23 trimmed video sequences, ranging from a few seconds to several minutes in length, wherein the evolution of identical dislocation microstructures can be observed over time.

We extracted frames from these video sequences at a rate of 30 frames per second, producing thousands of frames for certain videos. Figure 5.1 illustrates three individual frames from four of the video sequences. In Video Sequence one, frames 2101 and 4951 depict similar dislocation

microstructures, with variations only in dislocation positions due to movement. In contrast, frame 51 displays a distinctly different microstructure, as the microscope's focus has shifted slightly to the right. This shift reveals additional dislocations in the newly focused region, and the field of view has been zoomed out to capture the entire structure. This frequent adjustment of focus and zoom by experimentalists reflects their adaptation of microscope settings to capture specific details of interest throughout the experiments.

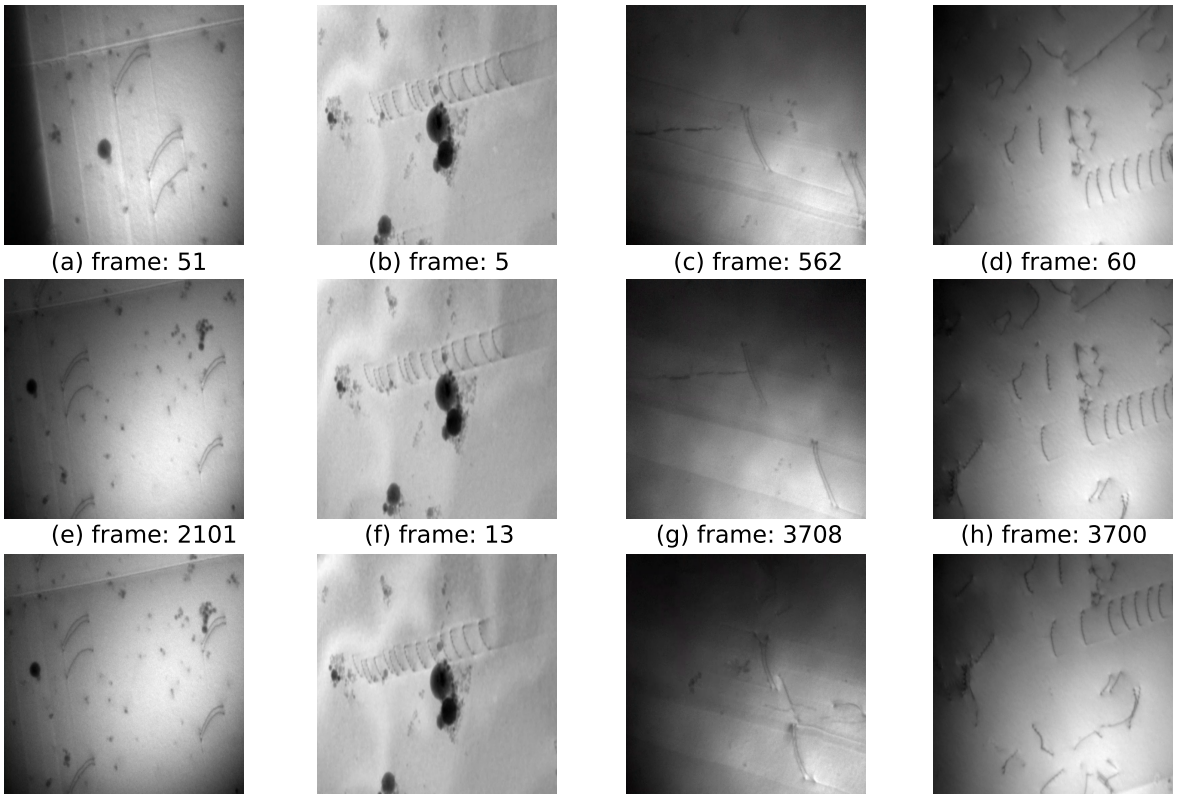


Figure 5.1: Three frames from four different experimental videos showing the variation in the dislocation microstructure from different videos and similarities from same video.

We have shown only the first few starting frames for video sequence two where we can observe that the dislocation microstructure is very similar and hardly changes. This is usually the case where dislocations do not move or change over several frames and hence there is only very little variation in the frames. It is only after several hundred frames that we see any noticeable difference in the microstructure as shown for video sequences three and four. This is one of the main reason why even though hundreds of frames are extracted from each video sequence,

frames with significant differences in terms of microstructure are quite few. We go through all the frames of a video sequence and manually selected a few frames to prepare a generalized labelled real dataset. The number of selected frames varies for each video sequence depending on the variance in the frames. For each video sequence, we selected a minimum of five frames for labelling. These frames are then labelled by selecting tens of points on dislocations using *labelme* and the labelled file containing the data of the dislocation points is saved as JSON file. The distribution of the number of frames labeled for each of the 23 video sequences is shown in Figure 5.2 where we have further split the total data into training and testing. This gives us a total of 239 images for training and 52 images for testing. The train-test split is done on the basis of video sequence so that all the frames of a video sequence are either part of training or testing data but not both. This ensures that there is no data leakage and we can properly evaluate the performance of any machine learning model trained on the training data. This is especially important for the dislocation dataset since any two images from the same video sequence might be very similar and hence there would be data leakage during the training if the split is not based on video sequences. The test data is not part of training in any way and kept aside only to test the performance of the trained machine learning models. The object is to evaluate the performance of the trained models on unseen data from different experiments.

We have obtained the TEM images from a large number of experimental data. It is important to learn more about the distribution of the data. There are a number of approaches we can use to visualize the distribution ranging from simple linear dimensionality reductions methods like Principle Component Analysis (PCA) which focuses on providing a representation of the high dimensional data by maximizing the variance of the dataset along each direction and providing a compressed representation of data in a low dimensional space. Cohn and Holm [119] utilized the VGG16 model to extract feature representations from micro-graphs of material microstructures, resulting in 4096-dimensional vectors. These high-dimensional representations were reduced to lower dimensions using PCA, enabling unsupervised methods such as K-means clustering to classify micro-graphs without the requirement for labeled data. This method is simpler and easier to use but may not be very useful to the datasets such as chemical structure and dislocation microstructure where it is also important to focus on the local features since in such datasets there are several but related low dimensional manifolds present within the high dimensional data. These relationships requires non linear methods such as Tree-based Mapper (TMAP)[120], t-distributed stochastic neighbor embedding (t-SNE) [121], or uniform manifold approximation and projection (UMAP) [122]. These methods have gained significant popularity for visualizing high-dimensional data, offering enhanced interoperability of results.

Various deep convolutional models are available for generating image feature representations. In this work, we employ the Self-Distillation with No Labels (DINO) model. DINO utilizes a self-supervised distillation approach, where a student model learns to mimic a teacher model using unlabeled data. This framework effectively enhances the feature extraction process, enabling the model to capture intricate patterns and rich representations within the data. DINO is designed to

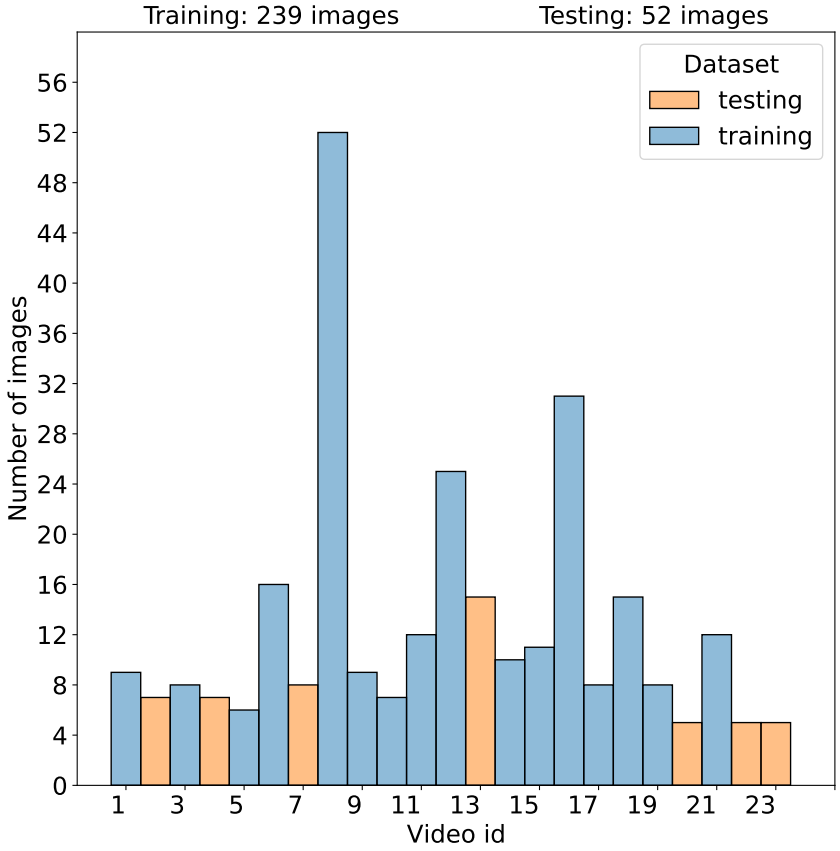


Figure 5.2: Distribution of number of images labelled from videos sequence along with the test and train split of the whole data. The dataset is split based on video sequences to prevent data leakage, ensuring that frames from the same video are not present in both the training and testing sets. This allows for a robust evaluation of model performance, as it better simulates real-world scenarios where models encounter unseen video sequences during testing.

learn invariant features, making it particularly effective for various downstream tasks, including clustering, classification, and segmentation.

In our study, we initiate a simple example where we utilize the DINO model to generate feature vectors for our training dataset comprising 239 dislocation microstructure images. Subsequently, we select a query image, as illustrated in Figure 5.3, and perform image indexing with the objective of retrieving "similar" images from the dataset. This process involves normalizing the feature vector of the query image and calculating the distance between the vectors using the L2 norm of the vector differences. The resulting scores facilitate the ranking of images where the image with the lowest score is the closest match, and the sorted results for the 15 closest matches

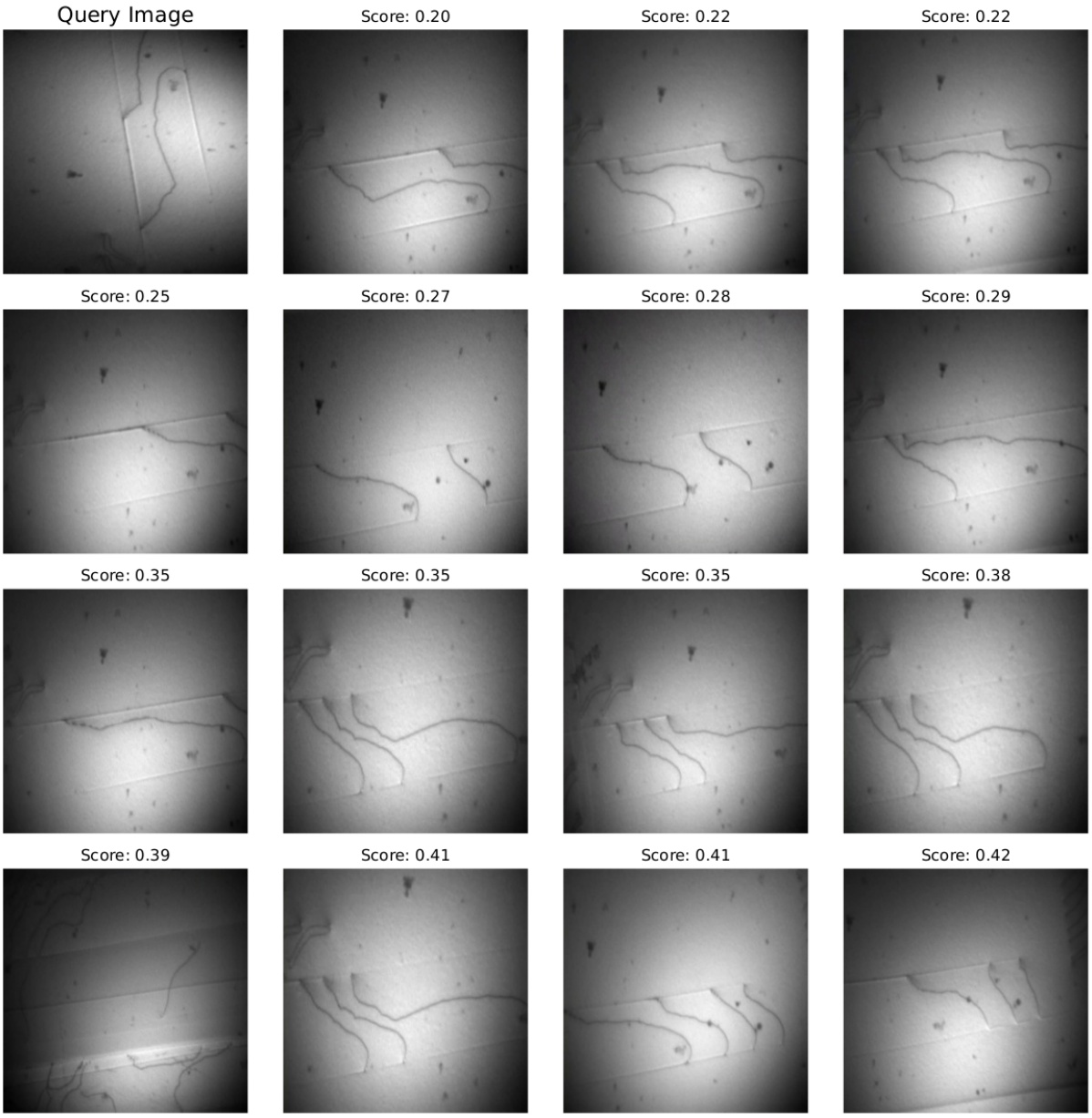


Figure 5.3: Query-based image retrieval using feature representations extracted from the DINO model. A single query image is indexed against a real dataset of 230 dislocation microstructure images, and the 15 closest matched images are shown. The associated scores indicate the similarity to the query image, with lower scores representing closer matches. The results demonstrate the model's ability to capture microstructural features and rotational invariance, as exemplified by a closely matched image that is a rotated version of the query.

are presented in Figure 5.3.

Upon examining the closest matched image with a score of 0.2, we observe that it is merely a rotated version of the query image. This observation highlights DINO's capability for rotational invariance, a crucial attribute to obtain high quality features. Such invariance ensures that the model recognizes similar structures despite variations in orientation, enhancing its robustness.

Furthermore, the majority of the matched images correspond to the same experimental conditions as the query image, indicating the model's effectiveness in grouping similar microstructures. This suggests that DINO not only captures fine-grained features but also retains the contextual relationships among images. The ability to identify similar structures effectively aids in material characterization, where understanding microstructural similarities can inform research and development processes. Even though the model has not been trained on microscopy image data of material microstructures, the model shows promising results and adaptability to our dislocation dataset where it was able to extract important useful features.

To analyze the data distribution, we utilized eight experimental videos, from which feature vectors were extracted for 50 randomly selected images per video using the specified feature extraction method. These feature vectors were then applied to obtain a TMAP visualization, as illustrated in Figure 5.4. In the TMAP visualization, images from individual experimental videos are predominantly clustered along distinct branches. However, some images appear in branches corresponding to different videos, indicating a degree of similarity across certain features. For instance, the image pair labeled as (b) consists of images from two separate videos, yet they share similar structural characteristics, such as a two-dislocation pileup configuration. Similarly, other locations within the TMAP also exhibit these cross-video similarities, as seen in (c). Additionally, transitional regions exist within the TMAP where branches from different experimental videos converge. At these junctions, as demonstrated in pair (b), the images situated at the transition points possess shared features, including comparable pileup configurations and dislocation shapes. This indicates that specific microstructural characteristics are common across multiple videos, contributing to these visual proximities in the TMAP representation.

The TMAP distribution provides a very nice distribution of the data and is very interesting from a machine learning point of view. As discussed above, the experimental data share some features but most of the them lie on their own branches. In such a case, it becomes very important to understand the robustness and quality of machine learning results where the new unseen data might lie on its own branch and has different domain distribution that the data used for training. This is discussed in details in the following section where we will evaluate the performance of the trained models on test data which is obtained from different experimental videos.

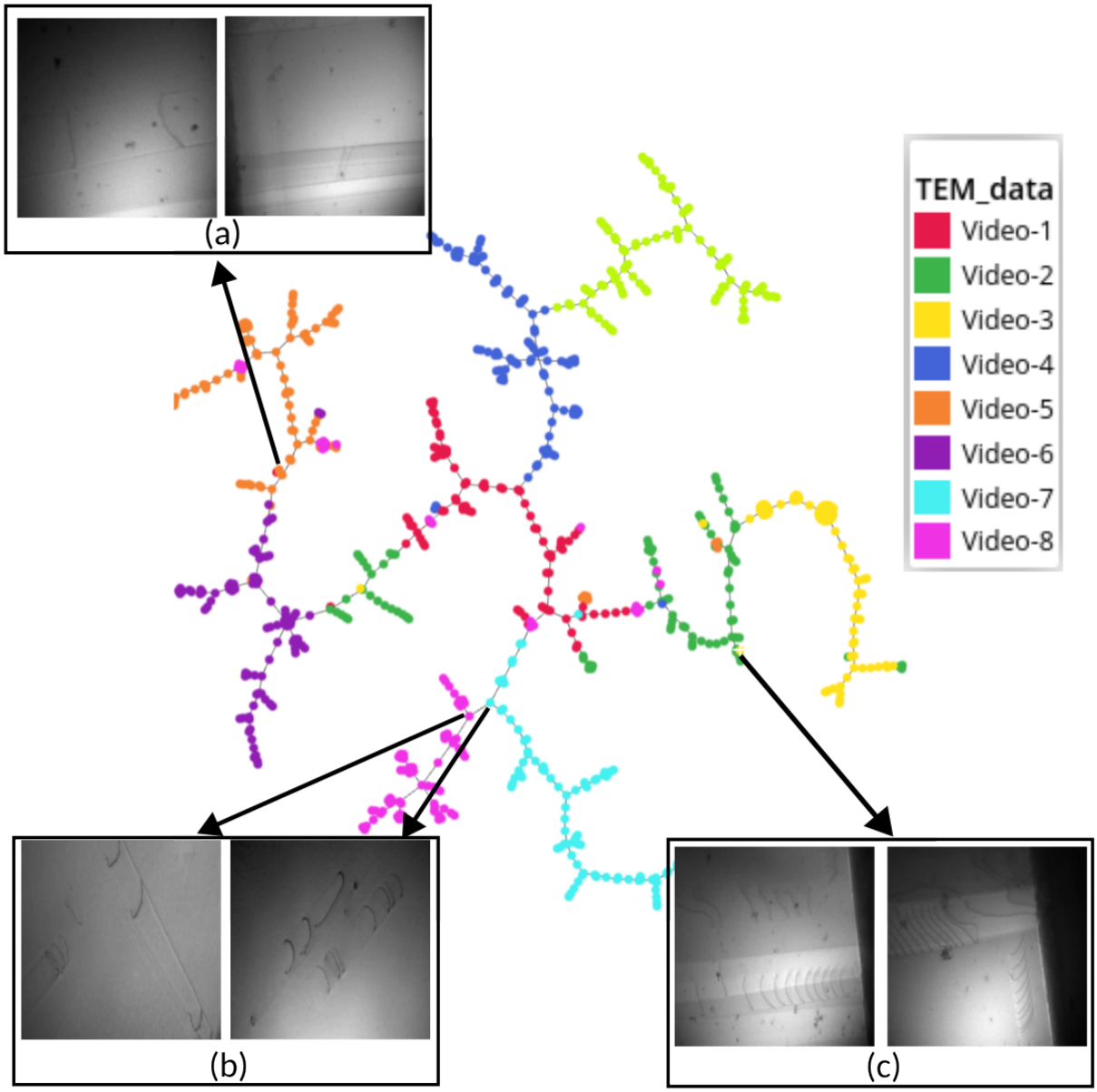


Figure 5.4: TMAP visualization of feature vectors extracted from 50 randomly selected images from each of eight experimental TEM videos. Images from the same experimental video predominantly cluster along distinct branches, illustrating the model's capability to capture video-specific microstructural patterns. Selected image pairs highlight structural relationships across videos: (a) and (c) show images from different videos appearing on the same branch, while (b) corresponds to images at branch transition points, sharing similar microstructural features such as two-dislocation pileups. This visualization demonstrates both intra-video consistency and inter-video microstructural similarities.

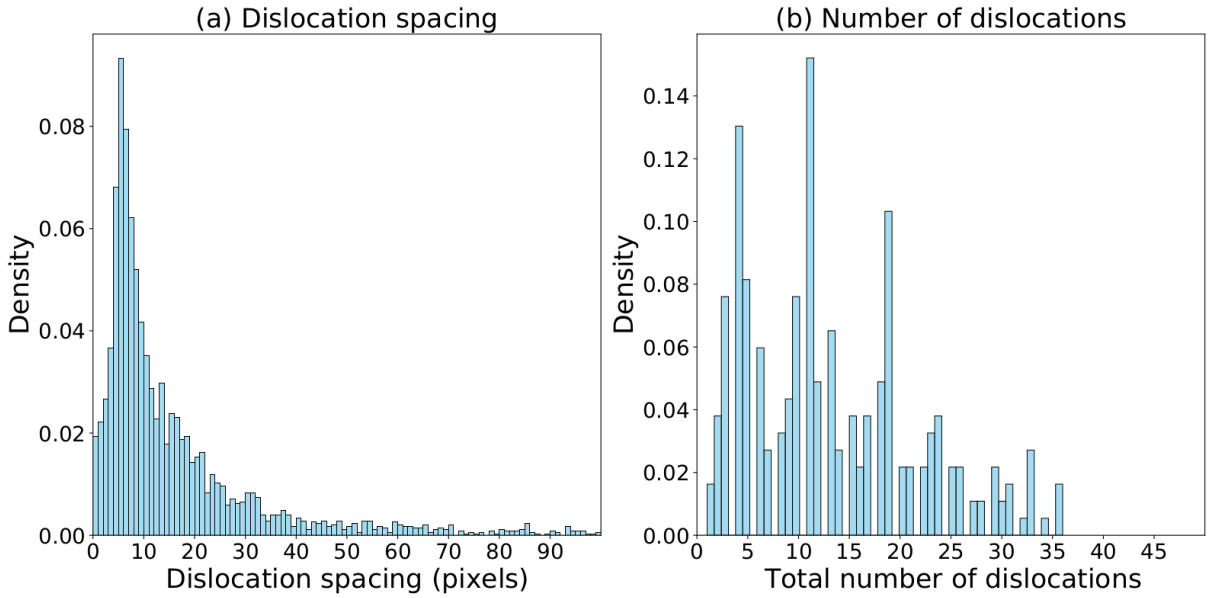


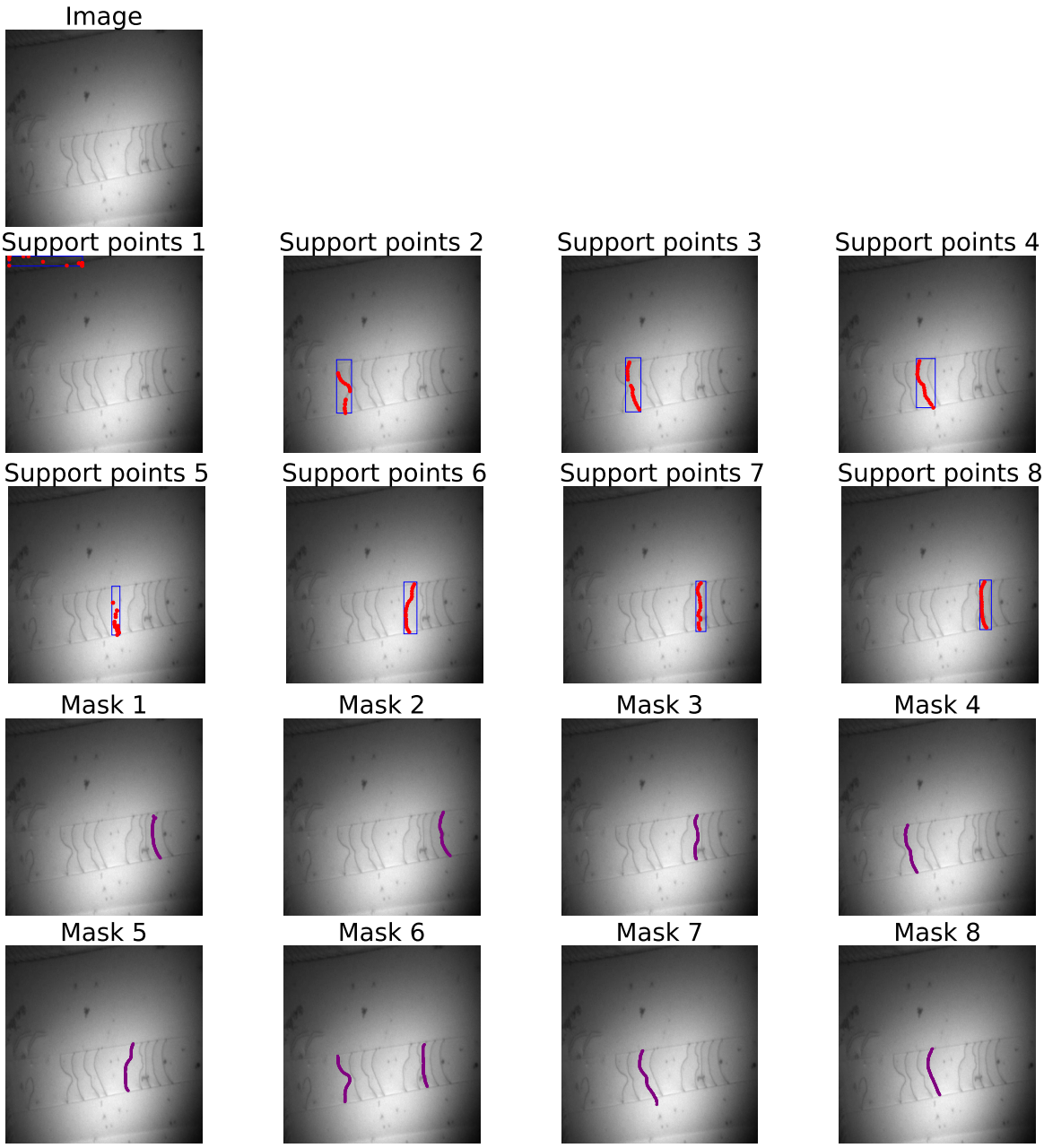
Figure 5.5: Probability density distribution of (a) the dislocation spacing between two nearest dislocations and (b) total number of dislocations in a real image of training dataset.

5.2 Results and Discussion

We have seen in Section 4.5 that both the machine learning approaches, multi-label segmentation and spline support point detection are very useful. For a dislocation microstructure where we can have a large number of dislocations (i.e., greater than 8) the multi-label segmentation method might fail to segment dislocations as individual dislocations in a single mask. However the method is very useful for structures where we have less dislocations that are very nearby. The method is able to capture the details at the pixel level and accurately predict such pair of very close dislocations as individual dislocations.

The probability density distribution of the dislocation spacing between two nearest dislocations and the total number of dislocations in an image of the real dataset is shown in Figure 5.5. We can observe from the distribution that most of the dislocations have distances of only 7 pixels and we can have as many as 35 dislocations in just one image. It is not necessarily the case that all the dislocations are important and are of interest. The moving dislocations are usually of interest and need to be studied. Since the data has been collected from a large number of experimental videos, the effectiveness of an approach may vary between different datasets.

To gain deeper insights, we apply the trained weights from the previous study—where models were exclusively trained on synthetic data—to predict two real images from the real dataset, as illustrated in Figure 5.6 and Figure 5.7. The figure also displays the predictions for these images



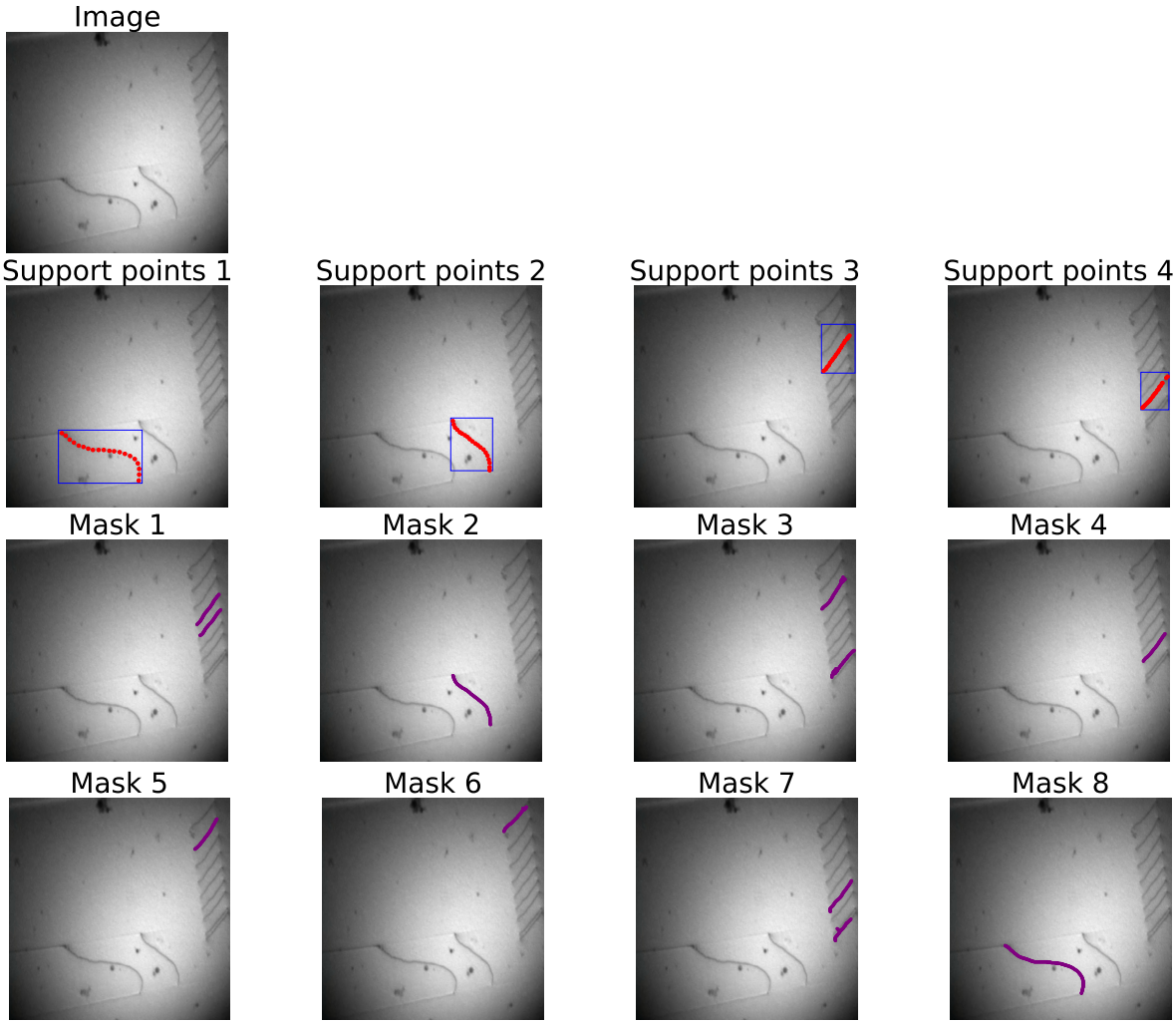
(a)

Figure 5.6: Predictions on a real TEM image using models trained exclusively on synthetic datasets. The figure compares the spline support point detection approach and the multi-label segmentation approach. Multi-label segmentation captures dislocations more accurately, particularly for closely spaced or complex-shaped dislocations, whereas the spline support point detection method misses several dislocations and mispredicts support points for some structures.

using both the spline support point detection and the multi-label segmentation approaches.

On comparing the results of two approaches, we can see that the multi-label segmentation approach provides much better results compared to the spline support point detection method for both the images. This is especially the case for real image in Figure 5.7 where the spline support point detection method only detects 4 dislocations out of 11 dislocations. For real image in Figure 5.6, the spline support point detection method fails to accurately capture the complex shapes of the dislocations, as demonstrated by the predictions shown as support points 2 and 3. Prediction shown as support points 5 is rather interesting where the model predicted a dislocation incorrectly. The results can be improved by increasing the number of support points, as this would allow for a finer resolution of the dislocations. However, optimizing the model parameters for a larger number of support points becomes challenging, as it introduces more complexity into the model and optimization of loss function. The architecture of Mask R-CNN requires the prediction of support points as one-hot encoded masks, allocating a separate channel for each dislocation. When dislocations are within sub-pixel proximity or we try to predict large number of support points, the models find it challenging to assign them to distinct channels and calculate the required loss in the positions of the points. This is one of the main reason why the support point detection method failed for the real image in Figure 5.7. The results are also affected by the use of NMS threshold for refining support points predictions, which, while designed to eliminate redundant detections, might inadvertently discard valid support points as redundancies when they are too close together. Multi-label segmentation results are much more precise and able to predict the dislocations at sub-pixel spacing accurately. It is interesting to note that even though the multi-label segmentation model was trained to predict a maximum of 9 dislocations it tries to predict more than 8 dislocations by predicting more than one instance in a single mask as shown for the two real images. In such cases, it could be beneficial to further post-process the masks using methods like contour detection to separate instances of dislocations. For example, two dislocations predicted in mask 6 of image in Figure 5.6 can be split into two distinct instances using such techniques.

To understand the effect of NMS in more details, let us again consider the real image in Figure 5.7 and post-process the spline support points predictions using different NMS threshold value as shown in Figure 5.8. We use a detection score of 0.9 so that the model is very confident in the detection of the bounding boxes and hence does not affect the post-processed results after NMS due to bounding box detection itself. We start with a very low value NMS of 0.01, which is much lower compared to the normally considered value of 0.25 (which is used earlier). The NMS algorithm begins with the bounding box that has the highest detection score. It then compares this box to all other boxes, calculating the IoU score between each pair of boxes. If the IoU value exceeds a predefined NMS threshold, the box with the lower detection score is suppressed (removed) because it is considered to be overlapping with the box of higher confidence. We can see that for the threshold value of 0.01, there are 6 dislocations which are suppressed since even if the IOU score of any two boxes is as small as 0.01 (slightly overlapping only), the bounding



(a)

Figure 5.7: Predictions on a real TEM image using models trained only on synthetic datasets. Multi-label segmentation provides superior detection of dislocations, successfully identifying nearly all instances, whereas spline support point detection fails to resolve several dislocations, particularly those in close proximity or with complex geometries. These results highlight the limitations of support point detection for real images and the effectiveness of multi-label segmentation in capturing sub-pixel dislocation structures.

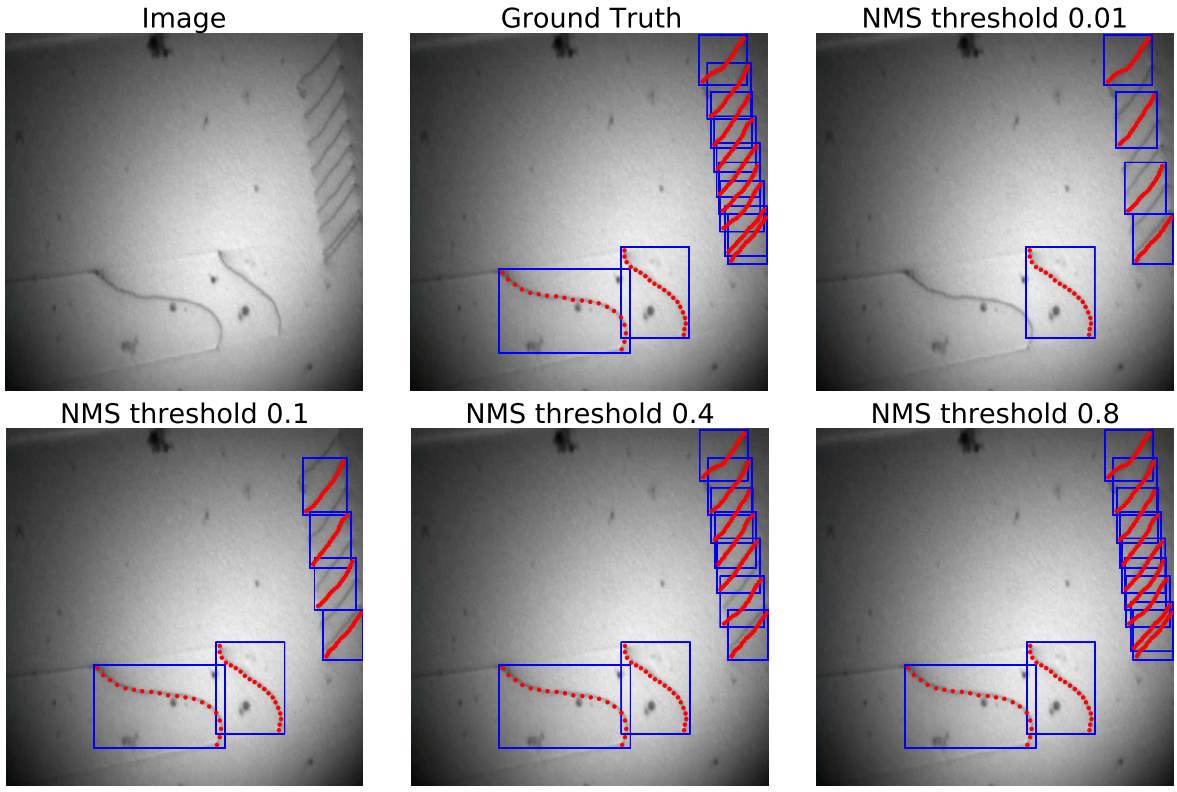


Figure 5.8: Effect of NMS threshold on the detection of closely spaced dislocations in a real image. The ground truth is shown alongside predictions made using different NMS threshold values, ranging from very low value of 0.1 to very high value of 0.8.

boxes will be suppressed. As we start to increase the threshold to 0.1, we start to see more dislocations. It is very interesting to note that now we start to get nearby dislocations of the pileup which we did not get previously with lower value since the overlap between them might be atleast 0.01. Even for a very high threshold of 0.4, we do not get all the dislocations and one of the two nearby dislocations are suppressed. It is only at very high threshold of 0.8, that we start to get all the predictions. A low NMS threshold is useful when it is important to avoid multiple detections of the same dislocation, even if this results in missing some dislocations. This helps in maintaining cleaner, less cluttered results. On the other hand, higher threshold values are beneficial when it is more important to ensure that no dislocation is missed, even if this leads to multiple detections of the same dislocation. While this approach increases recall (the number of detected dislocations), it may decrease precision because of the presence of duplicate bounding boxes for the same dislocation.

Dislocation detection in real crystal images presents significant challenges, particularly when models are trained solely on synthetic data. Although purely synthetic datasets can be generated in large volumes and with controlled dislocation properties, domain gaps—such as differences in contrast, noise characteristics, and spatial patterning—often hinder these models from performing optimally on real images. To address this issue, we build upon our previous study and propose an enhanced training methodology that leverages a multi-label segmentation approach with transfer learning. Our goal is to adapt a model originally trained on a purely synthetic dataset, referred to as *SD3*, so that it can more accurately detect dislocations in real images and we can get better results.

In our analysis of real dataset, we observed that some images can contain as many as 35 dislocations as illustrated in Figure 5.5, which poses a challenge for multi-label segmentation approach. For simplicity, we constrain our model to a maximum of 20 dislocations. By design, this decision does not substantially degrade performance on most images, as the distribution of total dislocations per image is skewed and the majority of images exhibit fewer than 20 dislocations. Specifically, we note that out of a set of 239 real images in the training dataset, 209 of them contain 20 or fewer dislocations. We consequently focus on these 209 images, creating a specialized subset of the real dataset that is more tractable for our chosen multi-label segmentation approach.

From these 209 images, an 80:20 split is used, yielding 167 training images and 42 validation images. We utilize transfer learning from the model that was previously trained on the large synthetic dataset, *SD3*, where the presence, shapes, and layouts of dislocations were systematically varied to give the model a robust initial understanding of dislocation-like features. Our synthetic dataset comprises thousands of generated images, allowing the original model to learn a diverse range of edge cases and spatial configurations. We then modify the final layer of this pretrained network so that it produces 20 output channels, one for each dislocation instance in our multi-label segmentation approach.

Following this architectural adjustment, we fine-tune the network on the real images by exposing it to noise characteristics, complex backgrounds, and possibly subtle dislocation patterns that were not encountered in the synthetic data. This fine-tuning step serves a critical role: it updates the model's parameters to better align with the statistical and visual properties inherent in real TEM images. In order to evaluate how much of an advantage we gain by leveraging a large synthetic dataset, we also train a separate model from scratch using only the 167 real training images. This additional experiment highlights the value of pretraining: whereas the scratch-trained model has to learn dislocation patterns and low-level features directly from limited real data, the transfer learning model already possesses a foundational understanding of dislocation-like structures, enabling it to adapt more rapidly.

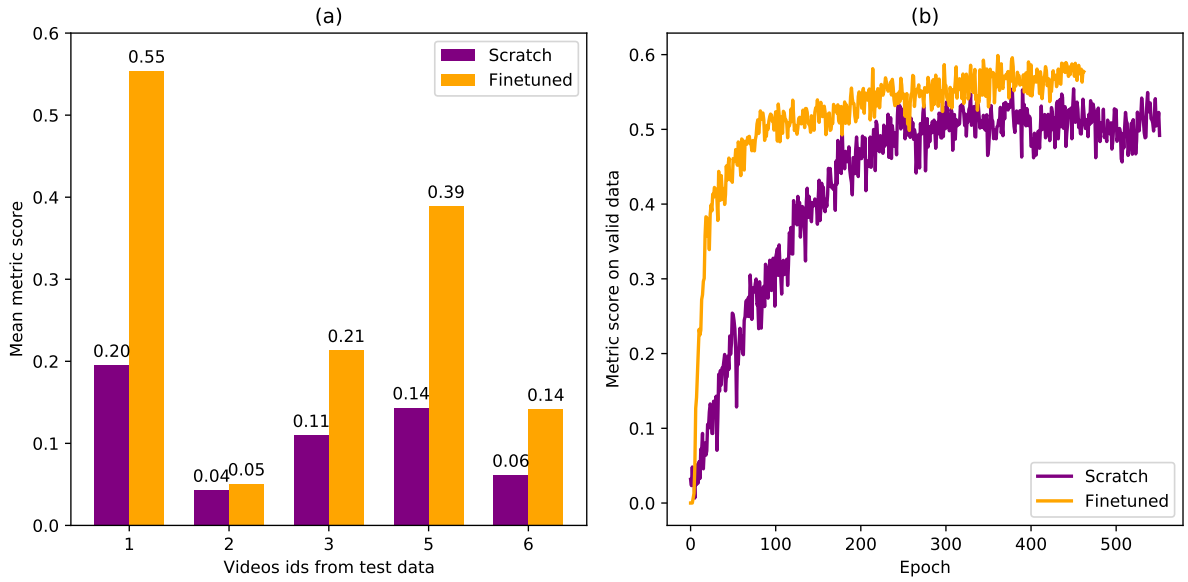


Figure 5.9: Comparison of two multi-label segmentation models evaluated on real TEM images. One model is trained exclusively on the real training dataset, while the other model is pretrained on the generalized synthetic dataset (SD3) and subsequently fine-tuned on the real dataset. (a) Metric scores for each test video, illustrating improved performance of the transfer learning model across diverse experimental conditions. (b) Evolution of the metric score on the validation dataset during training, showing faster convergence and higher final performance for the model pretrained on synthetic data.

During the training of the two models, the metric score on the validation dataset obtained from the data split is presented in Figure 5.9(b). It is observed that the model pretrained on the synthetic dataset, when finetuned on the real dataset, not only converges faster but also achieves a better performance compared to training a model from scratch. Specifically, a metric score of 0.59 was obtained using the transfer learning approach, highlighting the benefits of leveraging synthetic data as an initial training stage.

Subsequently, these trained models were evaluated on a separate test dataset that remained inaccessible during the training phase. The metric score was computed individually for each video in the test dataset, as illustrated in Figure 5.9(a) for both the models. The evaluation on the test data reveals that the metric scores are notably low. For certain videos, such as video 2, the mean metric score reaches as low as 0.04, indicating that the model exhibits minimal effectiveness for specific data instances. Addressing this issue forms one of the primary objectives of our work.

Dislocation image data exhibits substantial diversity, with significant variations in terms of

dislocation shapes, structures, and the number of dislocations. This diversity poses a considerable challenge in developing a generalized machine learning model capable of consistently delivering high-quality results across all dislocation datasets. The synthetic data approach thus becomes highly valuable, as it allows the generation of a diverse set of training examples, thereby enhancing the model's ability to generalize. As shown in the results for all videos in the test dataset, the model that utilized transfer learning from synthetic data consistently outperforms the model trained solely on real data.

5.3 Conclusion

In this study, we have addressed the challenge of generalization in machine learning models applied to real data, with the goal of developing a model that can effectively perform on unseen or new experimental datasets. Our objective is to identify the primary limitations that hinder the generalization capability of these models. A major factor that restricts the generalization of machine learning approaches is the availability of a comprehensive, generalized real dataset. As shown in Figure 5.9(a), the model, even after fine-tuning on real data, demonstrates limited performance on unseen experimental data. The highest metric score achieved for an experimental dataset is 0.55, which remains insufficient for accurately representing dislocations as splines.

In certain situations when we only have very limited amount of training data, applying transfer learning directly to the dataset of interest can be both practical and efficient. For instance, consider a scenario where, out of all the dislocation present in the images, only a subset of dislocations actively undergoes motion that is relevant to the particular experiment. Rather than annotating every dislocation in every frame, we can greatly streamline our labeling effort by focusing only on these actively moving dislocations. A small number of frames may be selected from the complete dataset, and each relevant dislocation within those frames can then annotated. These annotations form the real training set, which can be used for fine-tuning a model that was initially trained on a larger synthetic dataset.

An additional time-saving strategy involves generating what can be called *partial labels*. With this approach, a model that has already been trained (either on synthetic data or on a different subset of real data) is used to automatically label frames in the dataset. These automatically generated labels—partial because they may not be fully accurate or comprehensive—are then manually refined by an expert. This refinement step requires far less labor than creating every label from scratch, as the human annotator only needs to correct errors, add missing labels, or remove irrelevant features. In doing so, the overall volume of high-quality real labels for the moving dislocations expands quickly and at a lower cost of annotation effort.

Once this partially labeled and corrected dataset is assembled, the model is fine-tuned on

these meticulously curated labels. Although this targeted approach can lead to a model that is over-fitted to a specific experiment's conditions—due to limited coverage and potential bias in the labeled frames—it nevertheless produces a highly specialized tool. This specialized model can then be reliably applied to the entire suite of experimental frames, yielding accurate and efficient detection of only the dislocations of primary interest. Such a strategy is particularly advantageous in time-sensitive or resource-limited settings, where the precise tracking of a select group of actively moving dislocations is more critical than capturing every single structural defect in the sample.

Bibliography

[1] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. "LabelMe: a database and web-based tool for image annotation". *International journal of computer vision*, 77(1):157–173, 2008.

[2] Gerhard Hildebrandt. "The Discovery of the Diffraction of X-rays in Crystals—A Historical Review". *Crystal Research and Technology*, 28(6):747–766, 1993.

[3] Michael Eckert. "Max von Laue and the discovery of X-ray diffraction in 1912". *Annalen der Physik*, 524:83–A85, 05 2012.

[4] David Phillips. "William Lawrence Bragg. 31 March 1890-1 July 1971". *Biographical Memoirs of Fellows of the Royal Society*, 25:75–143, 1979. ISSN 00804606.

[5] James Alfred Ewing and Walter Rosenhain. "Viii. the crystalline structure of metals.(second paper.)". *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 195(262-273):279–301, 1900.

[6] Henry Clifton Sorby. "On the microscopical, structure of crystals, indicating the origin of minerals and rocks". *Quarterly Journal of the Geological Society*, 14(1-2):453–500, 1858.

[7] MF Osmond and William Chandler Roberts-Austen. "XI. On the structure of metals, its origin and changes". *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, (187):417–432, 1896.

[8] James Alfred Ewing and Walter Rosenhain. "Experiments in micro-metallurgy:-Effects of strain. Preliminary notice". *Proceedings of the Royal Society of London*, 65(413-422): 85–90, 1900.

[9] Henry Cort Harold Carpenter and Constance F Elam. "The production of single crystals of aluminium and their tensile properties". *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 100(704):329–353, 1921.

[10] G. I. Taylor and C. F. Elam. "Bakerian Lecture. The Distortion of an Aluminium Crystal during a Tensile Test". *Proceedings of the Royal Society of London. Series A, Containing*

Papers of a Mathematical and Physical Character, 102(719):643–667, 1923. ISSN 09501207.

[11] Geoffrey Ingram Taylor and William Scott Farren. "The distortion of crystals of aluminium under compression.—Part I". *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 111(759):529–551, 1926.

[12] JA Frenkel. "On the theory of the elastic limit and the strength of crystalline bodies". *journal for physics*, 37(7-8):572–609, 1926.

[13] Vito Volterra. "Sur l'équilibre des corps élastiques multiplement connexes". *Annales scientifiques de l'École Normale Supérieure*, 24:401–517, 1907.

[14] Geoffrey Ingram Taylor. "The mechanism of plastic deformation of crystals. Part I.—Theoretical". *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 145(855):362–387, 1934.

[15] E. Orowan. "Zur Kristallplastizität. III". *Zeitschrift fur Physik*, 89(9-10):634–659, September 1934.

[16] Michael Polanyi. "Über eine Art Gitterstörung, die einen Kristall plastisch machen könnte". *Zeitschrift für Physik*, 89:660–664, 1934.

[17] Nevill Francis Mott. "Bakerian lecture: dislocations, plastic flow and creep". *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 220(1140): 1–14, 1953.

[18] Alan H Cottrell, A Seeger, and JL Amorós. "Dislocations in crystals". In *Deformation and Flow of Solids/Verformung und Fliessen des Festkörpers: Colloquium Madrid September 26–30, 1955/Kolloquium Madrid 26. bis 30. September 1955*, pages 33–52. Springer, 1956.

[19] David B Williams, C Barry Carter, David B Williams, and C Barry Carter. *The transmission electron microscope*. Springer, 1996.

[20] Ernst Abbe. "Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung". *Archiv für mikroskopische Anatomie*, 9(1):413–468, 1873.

[21] PB Hirsch, RW Horne, and MJ Whelan. "Direct observations of the arrangement and motion of dislocations in aluminium". *Philosophical Magazine*, 86(29-31):4553–4572, 2006.

[22] P.B. Hirsch, R.W. Horne, and M.J. Whelan. "Transmission Electron Microscopy Observations of Dislocations". *Proceedings of the Royal Society of London. Series A. Mathematical*

and Physical Sciences, 235:464–477, 1956.

[23] Mike L Jenkins and Mark A Kirk. *Characterisation of radiation damage by transmission electron microscopy*. CRC Press, 2000.

[24] Jiwon Park, Reza Darvishi Kamachali, Sung-Dae Kim, Su-Hyeon Kim, Chang-Seok Oh, Christian Schwarze, and Ingo Steinbach. "First evidence for mechanism of inverse ripening from in-situ TEM and phase-field study of δ precipitation in an Al-Li alloy". *Scientific reports*, 9(1):3981, 2019.

[25] P.B. Hirsch, A. Howie, and M.J. Whelan. "Weak-Beam Dark-Field Microscopy: A New Technique for Dislocation Imaging". *Journal of Applied Crystallography*, 3:319–324, 1960.

[26] Hao Zhou, Chongxiang Huang, Xuechao Sha, Lirong Xiao, Xiaolong Ma, Heinz Werner Höppel, Mathias Göken, Xiaolei Wu, Kei Ameyama, Xiaodong Han, et al. "In-situ observation of dislocation dynamics near heterostructured interfaces". *Materials Research Letters*, 7(9):376–382, 2019.

[27] Sung-Dae Kim, Jun Young Park, Seong-Jun Park, Joonoh Moon, Heon-Young Ha, Chang-Hoon Lee, Jun-Yun Kang, Jong-Ho Shin, Tae-Ho Lee, et al. "Direct observation of dislocation plasticity in high-Mn lightweight steel by in-situ TEM". *Scientific Reports*, 9 (1):1–13, 2019.

[28] Dominik Steinberger, Inas Issa, Rachel Strobl, Peter J Imrich, Daniel Kiener, and Stefan Sandfeld. "Data-mining of in-situ TEM experiments: Towards understanding nanoscale fracture". *Computational Materials Science*, 216:111830, 2023.

[29] K Nogiwa, T Yamamoto, K Fukumoto, H Matsui, Yasuyoshi Nagai, Kunio Yubuta, and M Hasegawa. "In situ TEM observation of dislocation movement through the ultrafine obstacles in an Fe alloy". *Journal of nuclear materials*, 307:946–950, 2002.

[30] Vahid Samaee, Riccardo Gatti, Benoit Devincre, Thomas Pardoen, Dominique Schryvers, and Hosni Idrissi. "Dislocation driven nanosample plasticity: new insights from quantitative in-situ TEM tensile testing". *Scientific reports*, 8(1):12012, 2018.

[31] M. Legros. "In situ mechanical TEM: Seeing and measuring under stress with electrons". *Comptes Rendus Physique*, 15, 2014.

[32] Subin Lee, María Jazmin Duarte, Michael Feuerbacher, Rafael Soler, Christoph Kirchlechner, Christian H Liebscher, Sang Ho Oh, and Gerhard Dehm. "Dislocation plasticity in FeCoCrMnNi high-entropy alloy: quantitative insights from in situ transmission electron microscopy deformation". *Materials Research Letters*, 8(6):216–224, 2020.

[33] Daniel Utt, Subin Lee, Yaolong Xing, Hyejin Jeong, Alexander Stukowski, Sang Ho Oh, Gerhard Dehm, and Karsten Albe. "The origin of jerky dislocation motion in high-entropy alloys". *Nature communications*, 13(1):4777, 2022.

[34] Lisa Y Chen, Mo-rigen He, Jungho Shin, Gunther Richter, and Daniel S Gianola. "Measuring surface dislocation nucleation in defect-scarce nanostructures". *Nature materials*, 14(7):707–713, 2015.

[35] Chen Zhang, Hengxu Song, Daniela Oliveros, Anna Fraczkiewicz, Marc Legros, and Stefan Sandfeld. "Data-mining of in-situ TEM experiments: On the dynamics of dislocations in CoCrFeMnNi alloys". *Acta Materialia*, 241:118394, 2022.

[36] Hengxu Song, Binh Duong Nguyen, Kishan Govind, Dénes Berta, Péter Dusán Ispánovity, Marc Legros, and Stefan Sandfeld. "Enabling quantitative analysis of in situ TEM experiments: A high-throughput, deep learning-based approach tailored to the dynamics of dislocations". *Acta Materialia*, 282:120455, 2025.

[37] Emad Oveisi, Antoine Letouzey, Sandro De Zanet, Guillaume Lucas, Marco Cantoni, Pascal Fua, and Cecile Hebert. "Stereo-vision three-dimensional reconstruction of curvilinear structures imaged with a TEM". *Ultramicroscopy*, 184:116–124, 2018.

[38] Leonardo Agudo Jácome, Kai Pöthkow, Olaf Paetsch, and Hans-Christian Hege. "Three-dimensional reconstruction and quantification of dislocation substructures from transmission electron microscopy stereo pairs". *Ultramicroscopy*, 195:157–170, 2018.

[39] Ryan B Sills and Douglas L Medlin. "Semi-Automated, Object-Based Tomography of Dislocation Structures". *Microscopy and Microanalysis*, 28(3):633–645, 2022.

[40] Okan Altingövde, Anastasiia Mishchuk, Gulnaz Ganeeva, Emad Oveisi, Cecile Hebert, and Pascal Fua. "3D reconstruction of curvilinear structures with stereo matching deep convolutional neural networks". *Ultramicroscopy*, 234:113460, 2022.

[41] Arindrajit Seal, Arunava Das, and Prasad Sen. "Watershed: an image segmentation approach". *International Journal of Computer Science and Information Technologies*, 6 (3):2295–2297, 2015.

[42] Pascal Getreuer. "Chan-Vese Segmentation". *Image Process. Line*, 2:214–224, 2012.

[43] Jaakko Sauvola, Tapio Seppänen, Sami Haapakoski, and Matti Pietikäinen. "Adaptive Document Binarization.". volume 33, pages 147–152 vol.1, 09 1997. ISBN 0-8186-7898-4.

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[45] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. "Deepglobe 2018: A challenge to parse the earth through satellite images". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 172–181, 2018.

[46] Lei He, Guanghui Wang, and Zhanyi Hu. "Learning depth from single images with deep neural network embedding focal length". *IEEE Transactions on Image Processing*, 27(9): 4676–4689, 2018.

[47] Ying Li, Haokui Zhang, Xizhe Xue, Yenan Jiang, and Qiang Shen. "Deep learning for remote sensing image classification: A survey". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6):e1264, 2018.

[48] Yabo Fu, Yang Lei, Tonghe Wang, Walter J Curran, Tian Liu, and Xiaofeng Yang. "Deep learning in medical image registration: a review". *Physics in Medicine & Biology*, 65(20): 20TR01, 2020.

[49] Geert Litjens, Francesco Ciompi, Jelmer M Wolterink, Bob D de Vos, Tim Leiner, Jonas Teuwen, and Ivana Išgum. "State-of-the-art deep learning in cardiovascular image analysis". *JACC: Cardiovascular Imaging*, 12(8 Part 1):1549–1565, 2019.

[50] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. "Deep Leaf Segmentation Using Synthetic Data". In *British Machine Vision Conference*, 2018.

[51] Leonid Mill, David Wolff, Nele Gerrits, Patrick Philipp, Lasse Kling, Florian Vollnhals, Andrew Ignatenko, Christian Jaremenko, Yixing Huang, Olivier De Castro, et al. "Synthetic image rendering solves annotation problem in deep learning nanoparticle segmentation". *Small Methods*, 5(7):2100223, 2021.

[52] Graham Roberts, Simon Y Haile, Rajat Sainju, Danny J Edwards, Brian Hutchinson, and Yuanyuan Zhu. "Deep learning for semantic segmentation of defects in advanced STEM images of steels". *Scientific reports*, 9(1):1–12, 2019.

[53] Mingren Shen, Guanzhao Li, Dongxia Wu, Yuhan Liu, Jacob RC Greaves, Wei Hao, Nathaniel J Krakauer, Leah Krudy, Jacob Perez, Varun Sreenivasan, et al. "Multi defect detection and analysis of electron microscopy images with deep learning". *Computational Materials Science*, 199:110576, 2021.

[54] K Sasaki, M Muramatsu, K Hirayama, K Endo, and M Murayama. "Nanoscale defect evaluation framework combining real-time transmission electron microscopy and integrated machine learning-particle filter estimation". *Scientific reports*, 12(1):1–10, 2022.

[55] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. "Mask R-CNN.". In *ICCV*, pages 2980–2988. IEEE Computer Society, 2017. ISBN 978-1-5386-1032-9.

[56] Shoieb Ahmed Chowdhury, MFN Taufique, Jing Wang, Marissa Masden, Madison Wenzlick, Ram Devanathan, Alan L Schemer-Kohrn, and Keerti S Kappagantula. "Automated Grain Boundary (GB) Segmentation and Microstructural Analysis in 347H Stainless Steel Using Deep Learning and Multimodal Microscopy". *Integrating Materials and Manufacturing Innovation*, 13(1):244–256, 2024.

[57] Wei-Ying Chen, Hangyu Li, Zhi-Gang Mei, Logan Ward, Matthew J Olszta, Jacob Haag, Isabella van Rooyen, Derek Hopkins, and Kevin Fielder. "Computer vision models and advanced TEM imaging for microstructures of irradiated AM316 stainless steels". Technical report, Argonne National Laboratory (ANL), Argonne, IL (United States), 2024.

[58] Md Shamim Hossain, Leisa J Armstrong, Jumana Abu-Khalaf, and David M Cook. "The segmentation of nuclei from histopathology images with synthetic data". *Signal, Image and Video Processing*, pages 1–9, 2023.

[59] Kexin Ding, Mu Zhou, He Wang, Olivier Gevaert, Dimitris Metaxas, and Shaoting Zhang. "A large-scale synthetic pathological dataset for deep learning-enabled segmentation of breast cancer". *Scientific Data*, 10(1):231, 2023.

[60] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. "Synthetic Data–what, why and how?". *arXiv preprint arXiv:2205.03257*, 2022.

[61] Vajira Thambawita, Pegah Salehi, Sajad Amouei Sheshkal, Steven A. Hicks, Hugo L. Hammer, Sravanthi Parasa, Thomas de Lange, Pål Halvorsen, and Michael A. Riegler. "SinGAN-Seg: Synthetic training data generation for medical image segmentation". *PLOS ONE*, 17(5):1–24, 05 2022.

[62] Patrick Trampert, Dmitri Rubinstein, Faysal Boughorbel, Christian Schlinkmann, Maria Luschkova, Philipp Slusallek, Tim Dahmen, and Stefan Sandfeld. "Deep Neural Networks for Analysis of Microscopy Images—Synthetic Data Generation and Adaptive Sampling". *Crystals*, 11(3):258, 2021.

[63] Gowtham Nimmal Haribabu, Jeyapriya Thimukonda Jegadeesan, Chiranjib Bhattacharya, and Bikramjit Basu. "A deep adversarial approach for the generation of synthetic titanium alloy microstructures with limited training data". *Computational Materials Science*, 230: 112512, 2023. ISSN 0927-0256.

[64] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. "Training deep networks with synthetic data: Bridging the reality gap by domain randomization". In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.

[65] Sehyun Chun, Sidhartha Roy, Yen Nguyen, Joseph Choi, H. Udaykumar, and Stephen Baek. "Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials". *Scientific Reports*, 10, 08 2020.

[66] Gowtham Nimmal Haribabu, Jeyapriya Thimukonda Jegadeesan, Chiranjib Bhattacharya, and Bikramjit Basu. "A deep adversarial approach for the generation of synthetic titanium alloy microstructures with limited training data". *Computational Materials Science*, 230: 112512, 2023. ISSN 0927-0256.

[67] Anh Tran Khoi Nguyen Quang Ho Nguyen, Truong Vu. "Dataset Diffusion: Diffusion-based Synthetic Dataset Generation for Pixel-Level Semantic Segmentation". In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[68] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. "Guiding Text-to-Image Diffusion Model Towards Grounded Generation". *ArXiv*, abs/2301.05221, 2023.

[69] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. " DiffuMask: Synthesizing Images with Pixel-level Annotations for Semantic Segmentation Using Diffusion Models ". In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1206–1217, Los Alamitos, CA, USA, October 2023. IEEE Computer Society.

[70] Yang Song and Stefano Ermon. *Generative modeling by estimating gradients of the data distribution*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[71] Youssef Kossale, Mohammed Airaj, and Aziz Darouichi. "Mode Collapse in Generative Adversarial Networks: An Overview". In *2022 8th International Conference on Optimization and Applications (ICOA)*, pages 1–6. IEEE, 2022.

[72] O Blender. "Blender—A 3D modelling and rendering package". *Retrieved. represents the sequence of Constructs1 to*, 4, 2018.

[73] Antón Cid-Mejías, Raúl Alonso-Calvo, Helena Gavilán, José Crespo, and Víctor Maojo. "A deep learning approach using synthetic images for segmenting and estimating 3D orientation of nanoparticles in EM images". *Computer Methods and Programs in Biomedicine*, 202:105958, 2021.

[74] Yosuke Toda, Fumio Okura, Jun Ito, Satoshi Okada, Toshinori Kinoshita, Hiroyuki Tsuji, and Daisuke Saisho. "Training instance segmentation neural network with synthetic datasets for crop seed phenotyping". *Communications biology*, 3(1):173, 2020.

[75] Tim Dahmen, Patrick Trampert, Faysal Boughorbel, Janis Sprenger, Matthias Klusch, Klaus Fischer, Christian Kübel, and Philipp Slusallek. "Digital reality: a model-based

approach to supervised learning from synthetic data". *AI Perspectives*, 1(1):1–12, 2019.

[76] Ole Schmedemann, Melvin Baaß, Daniel Schoepflin, and Thorsten Schüppstuhl. "Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection". *Procedia CIRP*, 107:1101–1106, 2022.

[77] Qinqin Yang, Zi Wang, Kunyuan Guo, Congbo Cai, and Xiaobo Qu. "Physics-driven synthetic data learning for biomedical magnetic resonance: The imaging physics-based data synthesis paradigm for artificial intelligence". *IEEE Signal Processing Magazine*, 40 (2):129–140, 2023.

[78] Taher Hajilounezhad, Rina Bao, Kannappan Palaniappan, Filiz Bunyak, Prasad Calyam, and Matthew R Maschmann. "Predicting carbon nanotube forest attributes and mechanical properties using simulated images and deep learning". *npj Computational Materials*, 7(1): 1–11, 2021.

[79] Davis Unruh, Venkata Surya Chaitanya Kolluru, Arun Baskaran, Yiming Chen, and Maria KY Chan. "Theory+ AI/ML for microscopy and spectroscopy: Challenges and opportunities". *MRS Bulletin*, 47(10):1024–1035, 2022.

[80] PJ Phillips, MC Brandes, MJ Mills, and M De Graef. "Diffraction contrast STEM of dislocations: Imaging and simulations". *Ultramicroscopy*, 111(9-10):1483–1487, 2011.

[81] Peter Humble. "The Analysis of Defects Using Computer Simulated Images". In *Introduction to Analytical Electron Microscopy*, pages 551–574. Springer, 1979.

[82] R Schäublin and P Stadelmann. "A method for simulating electron microscope dislocation images". *Materials Science and Engineering: A*, 164(1-2):373–378, 1993.

[83] W Wu and Robin Schäublin. "TEM diffraction contrast images simulation of dislocations". *Journal of microscopy*, 275(1):11–23, 2019.

[84] Lionel C Gontard, Joaquín Pizarro, Ángel Ruiz-Zafra, and Jesús Hernández-Saz. "Simulation of transmission electron microscopy images using a generalized single-slice approach: The case of self-assembled quantum dots". *Materials Characterization*, 164:110312, 2020.

[85] Patrick G Callahan, Jean-Charles Stinville, Eric R Yao, McLean P Echlin, Michael S Titus, Marc De Graef, Daniel S Gianola, and Tresa M Pollock. "Transmission scanning electron microscopy: Defect observations and image simulations". *Ultramicroscopy*, 186:49–61, 2018.

[86] KAZUO Ishizuka and NATSU Uyeda. "A new theoretical and practical approach to the multislice method". *Acta Crystallographica Section A: Crystal Physics, Diffraction,*

Theoretical and General Crystallography, 33(5):740–749, 1977.

[87] Kazuo Ishizuka. "A practical approach for STEM image simulation based on the FFT multislice method". *Ultramicroscopy*, 90(2-3):71–83, 2002.

[88] M De Graef. "Image Simulations of Defect Images in STEM Observation Mode". *Microscopy and Microanalysis*, 16(S2):246–247, 2010.

[89] Zhongfu Zhou, ML Jenkins, Sergei L Dudarev, AP Sutton, and MA Kirk. "Simulations of weak-beam diffraction contrast images of dislocation loops by the many-beam Howie–Basinski equations". *Philosophical Magazine*, 86(29-31):4851–4881, 2006.

[90] Junyu Dong, Jun Liu, Kang Yao, Mike Chantler, Lin Qi, Hui Yu, and Muwei Jian. "Survey of Procedural Methods for Two-Dimensional Texture Generation". *Sensors*, 20(4), 2020. ISSN 1424-8220.

[91] Nakamasa Inoue, Eisuke Yamagata, and Hirokatsu Kataoka. "Initialization Using Perlin Noise for Training Networks with a Limited Amount of Data". In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1023–1028. IEEE, 2021.

[92] Andrei Tatarinov. "Perlin noise in real-time computer graphics". In *GraphiCon*, pages 177–183, 2008.

[93] Casey Duncan. "Perlin Noise Library for Python". `https://github.com/caseman/noise`, Accessed: 2014. GitHub repository.

[94] Kouichi Tougou, Maoto Fukui, Ken-ichi Fukumoto, Ryoya Ishigami, and Kiyohiro Yabuuchi. "Dynamic interaction between the dislocations and cavities in tungsten during tensile test". *Nuclear Materials and Energy*, 30:101130, 2022.

[95] Gulnaz Ganeeva, Okan Altingövde, Quy Ong Khac, Francesco Stellacci, Pascal Fua, Emad Oveisi, and Cécile Hébert. "Automatic 3D Reconstruction by Deep Learning Neural Networks Using Images Acquired via 4D-STEM Stereo Imaging". *Microscopy and Microanalysis*, 28(S1):218–220, 2022.

[96] Lu Wang, Yanming Zhang, Hou Yi Chia, and Wentao Yan. "Mechanism of keyhole pore formation in metal additive manufacturing". *npj Computational Materials*, 8(1):22, 2022.

[97] Kathleen Bates, Kim N Le, and Hang Lu. "Deep learning for robust and flexible tracking in behavioral studies for C. elegans". *PLOS Computational Biology*, 18(4):e1009942, 2022.

[98] Shoubhik Chandan Banerjee, Khursheed Ahmad Khan, and Rati Sharma. "Deep-worm-tracker: Deep learning methods for accurate detection and tracking for behavioral studies

in C. elegans". *Applied Animal Behaviour Science*, 266:106024, 2023.

[99] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. "A survey on instance segmentation: state of the art". *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

[100] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[101] Pedro F Felzenszwalb and Daniel P Huttenlocher. "Efficient graph-based image segmentation". *International journal of computer vision*, 59:167–181, 2004.

[102] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks". *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[103] Luisa M Zintgraf, Taco S Cohen, and Max Welling. "A new method to visualize deep neural networks". *arXiv preprint arXiv:1603.02518*, 2016.

[104] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[105] Muhammad Hussain. "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection". *Machines*, 11(7): 677, 2023.

[106] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. "Yolact: Real-time instance segmentation". In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9157–9166, 2019.

[107] Sasha Targ, Diogo Almeida, and Kevin Lyman. "Resnet in resnet: Generalizing residual architectures". *arXiv preprint arXiv:1603.08029*, 2016.

[108] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". *arXiv preprint arXiv:1409.1556*, 2014.

[109] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[110] Pavel Iakubovskii. "Segmentation Models Pytorch". `https://github.com/qubvel/`

`segmentation_models.pytorch`, Accessed: 2019.

[111] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations". In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017.

[112] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. "The connected-component labeling problem: A review of state-of-the-art algorithms". *Pattern Recognition*, 70:25–43, 2017.

[113] Gary Bradski. "The OpenCV Library.". *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[114] Oscar Kin-Chung Au and Chiew-Lan Tai and Hung-Kuo Chu and Daniel Cohen-Or and Tong-Yee Lee. "Skeleton extraction by mesh contraction". *ACM Trans. Graph.*, 27(3): 1–10, August 2008. ISSN 0730-0301.

[115] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation". In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[116] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. "Photorealistic text-to-image diffusion models with deep language understanding". *Advances in neural information processing systems*, 35:36479–36494, 2022.

[117] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis". *ArXiv*, abs/2307.01952, 2023.

[118] Xue Ying. "An overview of overfitting and its solutions". In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[119] Ryan Cohn and Elizabeth Holm. "Unsupervised machine learning via transfer learning and k-means clustering to classify materials image data". *Integrating Materials and Manufacturing Innovation*, 10(2):231–244, 2021.

[120] Daniel Probst and Jean-Louis Reymond. "Visualization of very large high-dimensional data sets as minimum spanning trees". *Journal of Cheminformatics*, 12, 02 2020.

[121] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE.". *Journal of machine learning research*, 9(11), 2008.

[122] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. "UMAP: Uniform Manifold Approximation and Projection". *The Journal of Open Source Software*, 3(29): 861, 2018.