

A Defect Dataset for Electrode Coating Manufacturing

Received: 10 June 2025

Accepted: 2 December 2025

Cite this article as: Sampath, V., Lee, A.S., Miller, S.D. *et al.* A Defect Dataset for Electrode Coating Manufacturing. *Sci Data* (2026). <https://doi.org/10.1038/s41597-025-06419-1>

Vignesh Sampath, Andrew S. Lee, Samuel David Miller, Noah H. Paulson, Yuepeng Zhang & Logan Ward

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

A Defect Dataset for Electrode Coating Manufacturing

Vignesh Sampath, Data Science and Learning Division, Argonne National Laboratory, Lemont, IL, USA

Andrew S. Lee, Applied Materials Division, Argonne National Laboratory, Lemont, IL, USA

Samuel David Miller, Applied Materials Division, Argonne National Laboratory, Lemont, IL, USA

Noah H. Paulson, Data Science and Learning Division, Argonne National Laboratory, Lemont, IL, USA

Yuepeng Zhang, Applied Materials Division, Argonne National Laboratory, Lemont, IL, USA

Logan Ward, Data Science and Learning Division, Argonne National Laboratory, Lemont, IL, USA

Abstract:

Electrode is a key component of many energy storage and energy conversion devices such as batteries and fuel cells. Defects in electrodes can significantly influence device performance and reliability and thus need to be monitored and eliminated during the electrode manufacturing process. Advancements in in-line metrology, computer vision, and machine learning have enabled the development of integrated hardware-software systems for automated defect detection and diagnostics. While several manufacturing domains have published defect datasets to support such efforts, publicly available datasets specific to electrode coating processes are not available. To fill this gap and support research on defect detection for automated coating processes, we present CoatingVision, a comprehensive dataset of slot-die coating images with labeled defect types. This dataset supports a diverse range of image recognition tasks, including defect segmentation, defect detection, and multi-label classification. It includes high-resolution images with associated labels for common defects such as surface cracks, delamination cracks, pinholes, and unclassified defects. To facilitate benchmarking and reproducible research, CoatingVision is packaged with an open-source codebase that enables comparative evaluation of AI models and hyperparameter configurations. The dataset has been meticulously curated to ensure high quality and consistency, providing researchers with reliable data for training and evaluating computer vision models. With over 2,200 image samples under various processing conditions, CoatingVision offers a robust foundation for developing automated defect detection systems. It promotes deeper insights into defect formation in coating manufacturing processes, which can be used to advance various coating-related applications including batteries and fuel cells.

Background & Summary:

In the fields of energy storage and advanced manufacturing, the fabrication of high-quality electrode coatings is crucial for ensuring the performance, reliability, and longevity of energy conversion and storage systems [1]. As the demand for efficient and sustainable energy solutions rises, the quality of the electrode coating process has become a key factor in achieving optimal energy storage performance [2]. Defects in the electrode coatings, whether occurring during production or application, can significantly reduce the efficiency, capacity, and lifespan of these systems [3]. Consequently, maintaining stringent quality control throughout the manufacturing process is essential to safeguard the performance and durability of the final product.

To meet the growing demand for high-performance energy storage systems, the development of automated quality assurance technologies for electrode coating has become increasingly important [4]. By integrating advanced computer vision and machine learning methods into manufacturing pipelines, real-time monitoring and precise evaluation of electrode coatings have become feasible [5]. These technologies enable the rapid detection and classification of defects, reducing reliance on manual inspection, enhancing scalability, and ensuring the production of high-quality coatings suited for large-scale industrial applications.

The coating process is inherently complex, with multiple parameters influencing the final electrode quality [1]. In general, electrode manufacturing involves mixing the active material [42], binder [43], and conductive agent to prepare a slurry [44], which is subsequently coated onto a substrate [45]. After drying to remove the solvent, an electrode sheet is obtained. The coating is then compacted and densified before the sheet is cut or slit to the desired dimensions (see figure 1). Variations in material properties, processing conditions, and environmental factors can introduce defects that degrade the performance and lifespan of energy storage devices.

Defects in electrode coatings, such as surface cracks, delaminated cracks, pinholes, and other irregularities, often arise due to the interactions between these various factors [6]. Due to the complex nature of electrode coating, understanding the fundamental mechanisms governing film formation and stability remains an active area of research. The interplay between thermodynamics and hydrodynamics plays a crucial role in governing the behavior of coating slurries during drying, compaction, and densification. However, the precise mechanisms leading to defect formation are still not fully understood, as they depend on a combination of material properties, processing conditions, and environmental factors.

Previous studies have suggested that during the drying process, solvent evaporation causes a reduction in particle–particle distances, leading to capillary pressure that drives particle cohesion. As the film continues to shrink, a porous network forms, and liquid-filled pores begin to empty, weakening cohesion between affected particles. This phenomenon results in surface cracks, which typically originate at the top layer and propagate toward the substrate [7],[8],[9]. The likelihood of cracking increases with larger particle sizes, wider particle size distributions, and thicker coatings. High drying temperatures tend to promote crack formation, whereas humidified drying conditions may help mitigate cracking [10],[11],[12],[13],[14].

At the interface between the coating and the substrate, the loss of cohesion can lead to delamination cracks. Delamination occurs when the adhesion between film particles and the substrate is insufficient to counteract the internal stresses generated during drying. The tendency for delamination depends on the adhesive properties of the coating material, the roughness of the substrate, and the overall mechanical integrity of the film [15].

From an electrochemical perspective, defects in electrode coatings can influence ion transport, electronic conductivity, and overall cell performance. While some studies [16] suggest that minor defects such as pinholes and small agglomerates may not significantly degrade electrochemical performance, the precise correlation between defect type, size, and distribution remains an open question. Research is still ongoing to determine the optimal defect tolerance thresholds and their cumulative impact on cycle life, capacity retention, and rate capability.

By employing deep learning models, the identification and characterization of coating defects can be significantly accelerated, providing an opportunity for more efficient and consistent quality control [46], [47], [48]. Such advancements are crucial for enhancing manufacturing processes, particularly in fields like electrode production, where defects can lead to costly failures in energy storage systems. The success of these models largely depends on access to high-quality, annotated datasets that support a variety of computer vision tasks. These datasets are crucial for training models that can effectively identify and classify defects under diverse production conditions. To fill this gap, we present a comprehensive dataset designed to accelerate research in automated defect detection systems for electrode fabrication quality control. Our dataset encompasses a broad spectrum of image recognition tasks, including defect segmentation, detection, and multi-label classification. It includes high-resolution images with labels for common defects such as surface cracks, delamination cracks, pinholes, and other defects, providing a solid foundation for advancing research in both battery and fuel cell production technologies.

Figure 1 goes here.

Literature Review:

Defect detection has been widely explored in various industrial domains, with multiple publicly available datasets facilitating advancements in computer vision-based inspection. These datasets can be broadly grouped based on their application domains, defect types, and data characteristics.

Construction and Infrastructure: Datasets in this domain primarily focus on crack detection in materials such as masonry, road pavement, concrete, and ceramic tiles. The Masonry dataset [17] includes images of masonry walls, featuring a wide range of crack sizes, from small patches affecting a few bricks to larger-scale cracks spanning entire walls. These images were sourced from both the internet and actual masonry buildings in Groningen, Netherlands, offering a diverse collection of real-world data. In a similar vein, the CFD dataset [18] captures urban road surface conditions in Beijing, China, with detailed, hand-labeled ground truth contours for cracks, providing high-quality data for road maintenance and inspection. Another well-established dataset, CrackTree200 [19], includes images of cracks collected from pavements and buildings,

with challenges such as varying lighting conditions, shadows, occlusions, and noise that complicate the detection process. In a complementary context, the Volker [20] and DeepCrack [21] datasets also focus on cracks in building structures, addressing issues such as low contrast and noisy environments, which are common in real-world data and present significant challenges for automated detection systems. The Ceramic Cracks Database [22], curated by students at the University of Pernambuco, expands the scope by including images of cracks on ceramic building facades, capturing defects across various textures and colors, further enriching the dataset diversity. The SDNET2018 [23] dataset offers images from concrete bridge decks, walls, and pavements, including cracks that range from narrow fissures as small as 0.06 mm to significantly wider gaps of 25 mm. The dataset provides real-world scenarios that include obstructions like shadows, surface roughness, scaling, and background debris, which add complexity to automated crack detection systems. Likewise, the Rissbilder dataset [20] contains a wide range of architectural crack types, showcasing the diversity in crack patterns that may appear in different structural contexts. Crack500 [24], collected at Temple University, further adds to the dataset landscape by using smartphone sensors to capture cracks, reflecting the growing trend of utilizing consumer-grade devices for field data collection. The GAPS dataset [25], collected under dry and warm conditions in 2015, provides images of various cracks found in urban environments, offering another valuable resource for the detection of road and pavement defects.

Electronics Manufacturing: Expanding into other industries, the KolektorSDD [26] dataset focuses on defective electronic converters, capturing small cracks in the plastic casing of the electronic components. With 52 defective and 347 non-defective samples, this dataset addresses a niche application in electronic defect detection. Yang et al. [27] introduced a Heat Sink Surface Defect dataset, which contains 1000 surface images of gold-plated tungsten copper alloy heat sinks, specifically aimed at detecting defects in high-precision components used in electronics and cooling systems.

Steel and Wood Industry: The NEU-DET dataset [28], released by Northeast University, includes gray-scale images depicting surface defects in hot-rolled steel strips. This dataset categorizes defects into three main types: patches, inclusions, and scratches, with 300 images for each defect, offering high-quality data for the steel industry. The Severstal dataset [29], one of the largest industrial datasets, features over 12,000 high-resolution grayscale images of steel sheet defects, annotated with pixel-wise masks by technical experts, covering various defect classes like cracks and scratches. In the wood industry, Pavel et al. [30] introduced a large-scale dataset of sawn timber surface defects, containing more than 43,000 labeled images of surface defects across 10 different types, including splits, knots, and holes. This extensive dataset is pivotal for quality control in the timber industry, where surface integrity is crucial for determining material usability in manufacturing and construction. The concise overview of the various datasets used for defect detection across multiple industries is presented in **Table 1**.

Dataset	Industry	Material	Size	Resolution	Tasks
NEU [28]	Steel	Steel	1800	200 x 200	Object detection, Segmentation
Severstal [29]	Steel	Steel	12568	1600x256	Segmentation
Pavel et al. [30]	Wood	Wood	4000	2800 x 1024	Object detection
Volker [20]	Construction	Concrete/stone	427	512 x 512	Segmentation
DeepCrack [21]	Construction	Concrete	443	544 x 384	Segmentation
CFD [18]	Road Pavement	Road	118	480 x 320	Segmentation
CrackTree200 [19]	Road Pavement	Road	175	800 x 600	Segmentation
Masonry [17]	Construction	Concrete	240	224 x 224	Classification, Segmentation
Ceramic [22]	Construction	Ceramic tiles	167	256 x 256	Segmentation
SDNET2018 [23]	Construction	Concrete	1411	4032 x 3024	Classification
Rissbilder [20]	Construction	Concrete	2736	512 x 512	Object detection, Segmentation
Crack500 [24]	Road Pavement	Road	3126	2000 x 1500	Segmentation
GAPS 384 [25]	Road Pavement	Road	383	640 x 540	Segmentation
KolektorSDD [26]	Electronics	Plastic casing	399	500 x ~1250	Segmentation
Heat Sink [27]	Electronics	Gold-plated tungsten-copper alloy	1000	320 x 320	Segmentation

Table 1: Summary of Open-Source Defect Detection Datasets Across Industries

To the best of our knowledge, no defect dataset specifically related to electrode coatings has been published. Our study introduces the first such dataset, filling the gap. By utilizing the slot die coating defect datasets, manufacturers can build robust deep learning models that enable automated defect diagnostics and autonomous decision-making. These models could optimize production parameters to reduce defects or even halt production in the presence of critical anomalies, improving production safety and efficiency, minimizing material waste, and enhancing sustainability in electrode manufacturing. These advancements lay the groundwork for smart factories, where data-driven insights help optimize every step of the production cycle, ensuring that high-performing energy storage technologies are consistently delivered.

Methods:

To create a CoatingVision dataset for defect detection in slot-die-coated films, we varied experimental parameters to generate coated samples with different quality levels. These parameters were carefully controlled to simulate real-world manufacturing conditions and help identify trends in defect formation.

Coating Process: The coating process was carried out using a Benchtop slot-die coater (VectorSC, FOM Technologies) with a constant coating speed of 0.5 m/min. The coatings were deposited on

a polytetrafluoroethylene (PTFE) substrate under different coating gaps (i.e., the distance between die head and substrate), which varied between 600 μm and 1100 μm with a 100 μm increment. The coating width and distance is 25 mm and 75 mm, respectively. After coating, the samples were dried in air for 40 minutes, followed by an additional 4 minutes drying at 80°C in an oven. The air-dry step was introduced to avoid extraneous defect generation during sample transfer from coater to oven. All samples were handled in a consistent manner to ensure consistency.

Ink Coating Parameters: The coating inks contain Vulcan carbon particles, mixed water and isopropyl alcohol (IPA) solvents, and Nafion binders. To observe how ink parameters influence defect formation in coating and drying, we varied the solvent properties of the inks such as dielectric constant, evaporation rate, and solubility by modifying the ratio of water and IPA. This parameter, alongside the varied coating gap, contributed to the observed coating quality differences between samples. Initial trends suggest that defects increased with larger coating gaps and higher water/alcohol ratios, which helped establish the ink/coating parameters suitable for the development of experimental protocols.

Data Collection Methods: Surface defects of the coating samples were captured using a Pixelink PL-D753CU area camera paired with a Navitar 12x zoom lens. The camera was positioned perpendicular to the sample surface, with the lens set to its minimum focusing distance to capture high-resolution optical images of the defects (see figure 2). To ensure optimal lighting and minimize shadows, LED lamps were placed on both sides of the camera. Due to the light-absorption nature of the carbon blacks in the coatings, the camera shutter speed was adjusted to allow for sufficient exposure, and the camera's sensitivity was set to its highest setting to ensure adequately bright images.

Figure 2 goes here.

Data Annotation Methods: The annotation process for the CoatingVision dataset follows an iterative pipeline aimed at continuously improving the quality and accuracy of the segmentation models (see figure 3). This process is designed to generate high-quality labeled data for the development and evaluation of automated defect detection systems in electrode coatings. Below is the detailed workflow:

Motion Blur Simulation and Image Extraction: To emulate the real-time production process and capture the effect of motion during the coating process, the data was recorded as video with a degree of motion blur. This motion blur simulated the dynamic nature of the manufacturing environment, which is often characterized by high-speed motion. From the recorded video

footage, individual frames were extracted, which were then converted into high-resolution images for analysis. This allowed the model to analyze data that mirrors the conditions found in actual production environments, where motion blur is often present due to the high-speed coating process.

Duplicate Removal and Hashmap Filtering: To ensure the uniqueness of the dataset and avoid redundant data, a hashmap-based filtering method was applied. This technique helped identify and remove duplicate images from the extracted frames, ensuring that each image in the dataset represented a distinct instance, free from repetition. By utilizing this filtering method, the dataset quality was improved, and potential biases from duplicate data were eliminated, ensuring more accurate training and validation of segmentation models.

Step 1: Preprocessing and Initial Feature Extraction

The raw images collected from the slot-die-coated electrode samples are preprocessed using traditional OpenCV-based techniques. These include morphological operations, thresholding, and contour analysis. These preprocessing steps are applied to enhance and extract key features from the images, such as surface cracks, delamination cracks, pinholes, and other miscellaneous defects. This step ensures that the images are adequately prepared for the segmentation task, highlighting relevant structures while eliminating unnecessary background noise.

Step 2: Initial Segmentation Model Training

With the preprocessed dataset, three different segmentation models are trained to learn to distinguish between defective and non-defective regions in the images. These models are trained using a combination of pixel-wise labeling and multi-label classification techniques to identify and segment various types of defects, including cracks, delamination, pinholes, and other defects common in slot-die coatings. The output of each model is a set of segmentation masks that highlight the predicted regions of interest, which include both the location and type of defects.

Step 3: Generation of Pseudo-Labels

After training, the three segmentation models (U-Net, FPN and Linknet) are used to infer defect locations in new, unannotated data. The resulting segmentation masks from these models are treated as pseudo-labels. These pseudo-labels serve as the initial predictions of the segmentation models on the unannotated images, representing the models' best attempt at identifying defects. The ensemble of these models' predictions helps to provide a more reliable first-pass estimate for the defect locations, benefiting from the combined strengths of multiple model architectures and hyperparameters.

Step 4: Manual Error Correction

The generated pseudo-labels are then reviewed and manually corrected by experienced annotators using annotation quality control (QC) tools (see figure 4). This step aims to address any misclassifications or boundary errors in the segmentation masks. Manual corrections may include refining the edges, adjusting the size of detected defects, or correcting misidentified regions. This step is crucial for ensuring the accuracy of the annotations, especially where the automated models may have struggled. The annotators in this step are skilled in identifying coating defects, with expertise in electrode production and coatings.

Step 5: Iterative Refinement and Model Retraining

Once the manual corrections are made, the corrected annotations are incorporated back into the dataset as ground truth labels. These refined annotations are then used to retrain the segmentation models. During retraining, an ensemble approach is employed where the outputs of all three models are combined to produce a more robust prediction. Each model in the ensemble is trained with different architectures or hyperparameters, helping to reduce bias and variance. The ensemble method benefits from the strengths of each model, leading to better overall performance and more accurate defect detection. After retraining, the updated segmentation models are used to generate new segmentation masks, which are again subjected to manual correction, forming a continuous loop of improvement.

Through this iterative process, the segmentation models gradually become more accurate and robust. With each iteration of manual refinement and model retraining, the models' ability to detect and segment defects in electrode coatings improves, ensuring better accuracy over time. To minimize missed defects, we employed a multi-pass manual verification protocol involving domain experts to carefully review annotations. This rigorous process significantly reduces the likelihood of unannotated subtle defects.

Step 6: Final Verification and Model Evaluation

Once the segmentation model has undergone sufficient iterations of training and refinement, the final set of annotations is reviewed by a team of expert annotators. This final verification ensures the accuracy and consistency of the ground truth data. The dataset is then divided into training and testing subsets for cross-validation and performance evaluation of the final segmentation model. Cross-validation ensures that the model is robust and can generalize well to unseen data, and performance evaluation metrics such as precision, recall, and F1-score are used to assess the model's efficacy.

Figure 3 goes here.

Figure 4 goes here.

Data Records:

The CoatingVision dataset is available at Figshare [41]. It consists of 2,227 images, each with a resolution of 480 by 640 pixels and 3 color channels. It includes annotations for three main types of defects: Surface Crack, Delamination, and Pinhole. Among these, Surface Crack is the most prevalent defect, appearing in 1,947 images, which accounts for approximately 87.43% of the dataset. Delamination is relatively rare, with only 203 images (9.12%) showing this defect. Pinhole defects (see figure 6) are present in 519 images, representing about 23.30% of the total. Also, the dataset contains some unclassified defects, such as foreign materials and ambiguous irregularities, which are not explicitly categorized but may affect analysis. By mirroring the operational challenges faced in industrial coating inspections, CoatingVision serves as a highly relevant benchmark for developing and evaluating computer vision algorithms in manufacturing quality control. The dataset is structured in a way that supports a diverse range of image recognition tasks, including defect segmentation, defect detection, and multi-label classification.

Defect Segmentation:

The segmentation masks in the CoatingVision dataset are provided in PNG format. Each image in the dataset is paired with a corresponding mask image that contains pixel-wise annotations for the various defects on the electrode surface (see figure 5). These masks are stored as multi-channel PNG images, where each defect type is encoded in a separate channel. Each segmentation mask is a 4-channel PNG image, where each channel represents a specific defect type. Each channel contains a binary mask where the value of each pixel indicates whether a specific defect is present at that location in the image. For example, in a 4-channel mask, the presence of a surface crack in a particular pixel is marked with a 1 in Channel 0, and similarly for delamination, pinholes, and unclassified defects in the corresponding channels. Each channel operates independently, enabling precise delineation of each defect type without overlap. The use of separate channels for each defect type ensures that multiple defects can be simultaneously identified at a given pixel location.

Figure 5 goes here.

Figure 6 goes here.

The dataset is structured such that each image in the CoatingVision dataset is paired with a corresponding mask in the *masks/* directory (see figure 7). The images and masks are organized into the following subdirectories:

Figure 7 goes here.

For each pinhole defect in the dataset, we measured the roundness to quantify how closely the shape of the defect resembles a perfect circle. The roundness is computed using the area and perimeter of the defect's boundary, as defined in the following equation:

$$\text{Roundness} = \frac{4\pi A}{p^2} \quad (1)$$

Where, A is the area of the pinhole defect, which is the space enclosed by the defect's boundary and p is the perimeter. To better understand the distribution of defect shapes in the dataset, we plotted a histogram of roundness values. The histogram (shown in figure 8) illustrates the frequency of different roundness values across all pinhole defects in the dataset. Most defects exhibit significant deviations from a roundness value of 1, indicating irregular shapes, while only a few are relatively circular.

Figure 8 goes here.

The defects are defined using visual characteristics and established geometric criteria based on shape, scale, and depth. Surface cracks, delamination, and pinholes are distinguished by their unique geometric features. Surface cracks appear as linear fractures, delamination as coating separation, and pinholes as small voids.

Defect Classification:

The CoatingVision dataset also includes labels for the multi-label classification task, where each image is annotated with one or more defect types. The defects are classified into four categories: surface cracks, delamination cracks, pinholes, and unclassified defects. The classification task requires determining the presence or absence of these defects in each image, with the possibility of multiple defects being present simultaneously. The labels for the multi-label classification task are provided in CSV format, where each row corresponds to a single image in the dataset (figure 9). Each image has a corresponding label indicating the presence or absence of each defect type.

Figure 9 goes here.

Pinhole Detection:

Each image in the CoatingVision dataset is accompanied by a corresponding annotation file in YOLO text format (figure 10). The annotation files follow a specific structure, where each line corresponds to an object (i.e., defect) detected in the image. The format of the annotation file is as follows:

- **Class ID:** An integer indicating the defect type (e.g., 0 for pinholes).
- **X_center:** The normalized x-coordinate of the center of the bounding box, relative to the image width (a value between 0 and 1).
- **Y_center:** The normalized y-coordinate of the center of the bounding box, relative to the image height (a value between 0 and 1).
- **Width:** The normalized width of the bounding box, relative to the image width (a value between 0 and 1).

- **Height:** The normalized height of the bounding box, relative to the image height (a value between 0 and 1).

An example of an annotation file for an image with pinholes and unclassified defects is shown in Table 2.

Class_id	x	y	w	h
0	0.356	0.489	0.132	0.081
1	0.580	0.312	0.104	0.060

Table 2: Example of an annotation file for an image containing pinholes and unclassified defects, with normalized coordinates (x , y) and dimensions (w , h) for bounding boxes.

This means that the first defect (class 0, pinhole) is located with its bounding box centered at coordinates (0.356, 0.489) and spans a width of 0.132 and height of 0.081, normalized to the image dimensions. The second defect (class 1, unclassified defect) is located similarly, with its own bounding box dimensions. figure 11 and 12 illustrate the directory structure of the CoatingVision detection dataset and examples of annotated pinhole defects, respectively.

Figure 10 goes here.

Figure 11 goes here.

Figure 12 goes here.

Technical Validation:

To evaluate the technical effectiveness and statistical quality of the CoatingVision dataset, we conducted a series of supervised experiments on defect segmentation, object detection, and multi-label classification. These experiments aimed to establish benchmark performance metrics for future research and practical applications, while also validating the utility of the dataset across various defect detection tasks. The following sections detail the methods employed and the results obtained for each task.

Defect Segmentation: The segmentation task focuses on pixel-wise detection of defects on the electrode surface. We employed multiple distinct segmentation models for this task: a ResNet-based U-Net [31], Feature Pyramid Networks (FPN) [32], and LinkNet [33]. These models are well-established in the field of image segmentation and have demonstrated strong performance in similar defect detection applications [35],[36].

For the segmentation experiments, the dataset was divided into training and testing sets, with 80% of the images used for training and the remaining 20% reserved for testing. To assess baseline segmentation performance, two commonly used metrics were utilized: the Dice Similarity Coefficient (DSC) and Intersection over Union (IoU).

Dice Similarity Coefficient (DSC): The DSC is a widely used metric that quantifies the overlap between two sets. It is defined as:

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \quad (2)$$

where A and B represent the predicted and ground truth segmentation masks, respectively, and $|A|$ and $|B|$ denote the number of pixels in each mask. The DSC ranges from 0 (no overlap) to 1 (perfect overlap).

Intersection over Union (IoU): The IOU is another widely used metric that measures the overlap between the predicted and ground truth segmentation masks (figure 13). It is defined as:

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

where $|A \cap B|$ is the number of pixels in the intersection of the predicted and ground truth masks, and $|A \cup B|$ is the number of pixels in their union. Similar to DSC, IoU ranges from 0 (no overlap) to 1 (perfect overlap), but it tends to penalize false positives more heavily than the DSC.

Figure 13 goes here.

Both the DSC and IoU are essential for evaluating the quality and reliability of the CoatingVision dataset for defect segmentation tasks. These metrics assess the alignment between the predicted segmentation masks and the ground truth annotations, providing insights into the dataset's accuracy in capturing defect characteristics. The supplementary table presents the evaluation results of the six segmentation model architectures, and figure 14 shows the training and validation loss across the different models.

Figure 14 goes here.

Defect Detection: For object detection, we utilized the YOLOv11 (You Only Look Once) framework [38], a widely adopted real-time object detection algorithm known for its speed and accuracy. The object detection task in the CoatingVision dataset aims to localize defects within the images using bounding boxes. The model was trained to predict bounding boxes around defects in the electrode surface, where each bounding box includes the defect location and its class label. To evaluate the performance of the YOLO model, we used the following commonly adopted object detection metrics: mean Average Precision (mAP), Precision, and Recall.

Mean Average Precision (mAP): mAP is the primary metric used for evaluating the overall performance of object detection models. It provides a summary of how well the model detects and localizes defects by calculating the average of average precision (AP) scores across different defect classes. The AP score is obtained by averaging precision values at various recall levels,

considering different confidence thresholds for detection. The mAP value ranges from 0 to 1, with higher values indicating better overall performance of the model in terms of both localization and classification of defects.

It is defined as:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

Where N is the number of defect classes (e.g., different types of defects) and AP_i is the Average Precision for the i -th class. AP itself is computed as the area under the Precision-Recall curve for each class, reflecting the model's precision at different levels of recall. A high mAP value indicates that the model has achieved both high precision and recall across all classes. Table 3 presents the evaluation results of the YOLOv11 and Faster-RCNN object detection models.

Model	$mAP @ 0.5$	$mAP @ [0.5:0.95]$	Precision	Recall
YOLOv11	0.7956	0.5208	0.7588	0.7621
Faster-RCNN	0.7480	0.4420	0.7785	0.6705

Table 3: Performance Metrics Object Detection Models

Figure 15 illustrates the Precision-Recall curve for YOLOv11, showing the model's ability to balance precision and recall across various defect types. The curve illustrates how well the model performs at different levels of confidence and recall, with the area under the curve (AUC) serving as a reflection of overall performance.

Figure 15 goes here.

Defect Classification: The multi-label classification task involves predicting multiple defect types simultaneously for each image, where each defect class can be independently predicted as present or absent. This is particularly relevant in the context of the CoatingVision dataset, where a single image can contain multiple types of defects on the electrode surface. To address this task, we employed two well-known deep learning models: ResNet [37], ConvNext [39], and EfficientNet [40]. These models are recognized for their strong performance in classification tasks, leveraging advanced architectures that efficiently capture both local and global features within the images. The performance of the models was evaluated using two standard metrics: F1 Score Precision, Recall and Accuracy (Table 4).

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both the precision (the ability to avoid false positives) and recall (the ability to detect all true positives). It is useful in multi-label classification, where multiple defect classes may be predicted for each image, and achieving a balance between precision and recall is critical.

The F1 score for a single label i is defined as:

$$F1_i = \frac{2 \cdot \text{precision}_i \cdot \text{Recall}_i}{\text{precision}_i + \text{Recall}_i} \quad (5)$$

The macro-averaged F1 score is computed by taking the average of the F1 scores across all classes:

$$F1_{macro} = \frac{1}{N} \sum_{i=1}^i F1_i \quad (6)$$

Where N is the total number of defect classes. A higher F1 score indicates a more balanced performance in terms of precision and recall for each defect class.

Architecture	Surface Crack				Delamination				Pinholes			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
ResNet [37]	0.9121	0.8736	0.8924	0.9074	0.7973	0.7266	0.7600	0.7963	0.7848	0.6902	0.7344	0.7768
ConvNext [39]	0.9374	0.8910	0.9135	0.9245	0.8740	0.6204	0.7257	0.7921	0.8684	0.6680	0.7551	0.8106
EfficientNet [40]	0.9383	0.8886	0.9127	0.9244	0.8777	0.6601	0.7535	0.8086	0.9229	0.7506	0.8276	0.8604

Table 4: Comparative Performance Analysis of three Different classifications model Architectures

Code availability:

The code for data annotation, model training, and evaluation is available on GitHub at <https://github.com/vigsam-coder/CoatingVision>. The dataset has been uploaded to Figshare and can be accessed at <https://figshare.com/articles/dataset/29260121>.

Data availability:

The dataset is available at <https://doi.org/10.6084/m9.figshare.29260121.v1>

Competing interest:

The authors declare no competing interests.

Author contribution:

Conceptualization: V.S, A.L, N.P, Y.Z. Data curation: V.S. Methodology: V.S, A.L, N.P, Y.Z. Formal analysis: V.S. Data Acquisition: S.M, A.L, V.S. Y.Z. Validation: V.S, L.W. Software: V.S., L.W. Investigation: V.S, A.L, N.P, Y.Z. Manuscript preparation: V.S. Supervision: N.P, Y.Z.

Acknowledgement:

This work was supported by the U.S. Department of Energy's Hydrogen and Fuel Cell Technologies Office and the Laboratory Directed Research and Development (LDRD) program at Argonne National Laboratory.

References

- [1] C. D. Reynolds, P. R. Slater, S. D. Hare, M. J. H. Simmons, and E. Kendrick, "A review of metrology in lithium-ion electrode coating processes," *Mater. Des.*, vol. 209, p. 109971, Nov. 2021, doi: 10.1016/j.matdes.2021.109971.
- [2] J. Park, J. Kim, J. Kim, M. Kim, T. Song, and U. Paik, "Sustainable and cost-effective electrode manufacturing for advanced lithium batteries: the roll-to-roll dry coating process," *Chem. Sci.*, p. 10.1039/D5SC00059A, 2025, doi: 10.1039/D5SC00059A.
- [3] A. Du Baret De Limé *et al.*, "Impact of Electrode Defects on Battery Cell Performance: A Review," *Batter. Supercaps*, vol. 5, no. 10, p. e202200239, Oct. 2022, doi: 10.1002/batt.202200239.
- [4] N. Choudhary *et al.*, "Autonomous Visual Detection of Defects from Battery Electrode Manufacturing," *Adv. Intell. Syst.*, vol. 4, no. 12, p. 2200142, Dec. 2022, doi: 10.1002/aisy.202200142.
- [5] A. Schoo, R. Moschner, J. Hülsmann, and A. Kwade, "Coating Defects of Lithium-Ion Battery Electrodes and Their Inline Detection and Tracking," *Batteries*, vol. 9, no. 2, p. 111, Feb. 2023, doi: 10.3390/batteries9020111.
- [6] D. Mohanty, E. Hockaday, J. Li, D. K. Hensley, C. Daniel, and D. L. Wood, "Effect of electrode manufacturing defects on electrochemical performance of lithium-ion batteries: Cognizance of the battery failure sources," *J. Power Sources*, vol. 312, pp. 70–79, Apr. 2016, doi: 10.1016/j.jpowsour.2016.02.007.
- [7] T. Rieder and E. U. Schlünder, "Selective evaporation of a binary mixture into dry or humidified air," *Chem. Eng. Process. Process Intensif.*, vol. 27, no. 2, pp. 83–93, Apr. 1990, doi: 10.1016/0255-2701(90)85012-S.
- [8] G. L. Brown, "Formation of films from polymer dispersions," *J. Polym. Sci.*, vol. 22, no. 102, pp. 423–434, Dec. 1956, doi: 10.1002/pol.1956.1202210208.
- [9] C. J. Martinez and J. A. Lewis, "Shape Evolution and Stress Development during Latex–Silica Film Formation," *Langmuir*, vol. 18, no. 12, pp. 4689–4698, Jun. 2002, doi: 10.1021/la0114833.
- [10] F. Scheepers, A. Stähler, M. Stähler, M. Carmo, W. Lehnert, and D. Stolten, "Steering and in situ monitoring of drying phenomena during film fabrication," *J. Coat. Technol. Res.*, vol. 16, no. 5, pp. 1213–1221, Sep. 2019, doi: 10.1007/s11998-019-00206-5.
- [11] C.-S. Tang, Y.-J. Cui, A.-M. Tang, and B. Shi, "Experiment evidence on the temperature dependence of desiccation cracking behavior of clayey soils," *Eng. Geol.*, vol. 114, no. 3–4, pp. 261–266, Aug. 2010, doi: 10.1016/j.enggeo.2010.05.003.
- [12] M. S. Tirumkudulu and W. B. Russel, "Cracking in Drying Latex Films," *Langmuir*, vol. 21, no. 11, pp. 4938–4948, May 2005, doi: 10.1021/la048298k.
- [13] R. C. Chiu and M. J. Cima, "Drying of Granular Ceramic Films: II, Drying Stress and Saturation Uniformity," *J. Am. Ceram. Soc.*, vol. 76, no. 11, pp. 2769–2777, Nov. 1993, doi: 10.1111/j.1151-2916.1993.tb04014.x.
- [14] C. Bauer, M. Cima, A. Dellert, and A. Roosen, "Stress Development During Drying of Aqueous Zirconia Based Tape Casting Slurries Measured by Transparent Substrate Deflection Method," *J. Am. Ceram. Soc.*, vol. 92, no. 6, pp. 1178–1185, Jun. 2009, doi: 10.1111/j.1551-2916.2009.03000.x.
- [15] P. Xu, A. S. Mujumdar, and B. Yu, "Drying-Induced Cracks in Thin Film Fabricated from Colloidal Dispersions," *Dry. Technol.*, vol. 27, no. 5, pp. 636–652, Apr. 2009, doi: 10.1080/07373930902820804.
- [16] L. David *et al.*, "Identifying degradation mechanisms in lithium-ion batteries with coating defects at the cathode," *Appl. Energy*, vol. 231, pp. 446–455, Dec. 2018, doi: 10.1016/j.apenergy.2018.09.073.
- [17] D. Dais, İ. E. Bal, E. Smyrou, and V. Sarhosis, "Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning," *Autom. Constr.*, vol. 125, p. 103606, May 2021, doi: 10.1016/j.autcon.2021.103606.

- [18] A. Ahmadi, S. Khalesi, and M. Bagheri, "Automatic road crack detection and classification using image processing techniques, machine learning and integrated models in urban areas: A novel image binarization technique," *J. Ind. Syst. Eng.*, vol. 11, no. 2018, pp. 85–97.
- [19] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, Feb. 2012, doi: 10.1016/j.patrec.2011.11.004.
- [20] M. Pak and S. Kim, "Crack Detection Using Fully Convolutional Network in Wall-Climbing Robot," in *Advances in Computer Science and Ubiquitous Computing*, vol. 715, J. J. Park, S. J. Fong, Y. Pan, and Y. Sung, Eds., in Lecture Notes in Electrical Engineering, vol. 715, Singapore: Springer Singapore, 2021, pp. 267–272. doi: 10.1007/978-981-15-9343-7_36.
- [21] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, Apr. 2019, doi: 10.1016/j.neucom.2019.01.036.
- [22] G. S. Junior, J. Ferreira, C. Millán-Arias, R. Daniel, A. C. Junior, and B. J. T. Fernandes, "Ceramic Cracks Segmentation with Deep Learning," *Appl. Sci.*, vol. 11, no. 13, p. 6017, Jun. 2021, doi: 10.3390/app11136017.
- [23] S. Dorafshan, R. J. Thomas, and M. Maguire, "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks," *Data Brief*, vol. 21, pp. 1664–1668, Dec. 2018, doi: 10.1016/j.dib.2018.11.015.
- [24] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, Apr. 2020, doi: 10.1109/TITS.2019.2910595.
- [25] M. Eisenbach *et al.*, "How to get pavement distress detection ready for deep learning? A systematic approach," in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA: IEEE, May 2017, pp. 2039–2047. doi: 10.1109/IJCNN.2017.7966101.
- [26] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, Mar. 2020, doi: 10.1007/s10845-019-01476-x.
- [27] K. Yang, Y. Liu, S. Zhang, and J. Cao, "Surface Defect Detection of Heat Sink Based on Lightweight Fully Convolutional Network," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2022.3188033.
- [28] 吴启科 吴, "NEU-DET." IEEE DataPort. doi: 10.21227/J84R-F770.
- [29] A. Grishin, BorisV, iBardintsev, and Oleg, "severstal-steel-defect-detection." 2019. [Online]. Available: <https://kaggle.com/competitions/severstal-steel-defect-detection>
- [30] P. Kodytek, A. Bodzas, and P. Bilik, "A large-scale image dataset of wood surface defects for automated vision-based quality control processes," *F1000Research*, vol. 10, p. 581, Jun. 2022, doi: 10.12688/f1000research.52903.2.
- [31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., in Lecture Notes in Computer Science, vol. 9351, Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.
- [32] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106.
- [33] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL: IEEE, Dec. 2017, pp. 1–4. doi: 10.1109/VCIP.2017.8305148.

- [34] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6230–6239. doi: 10.1109/CVPR.2017.660.
- [35] S. Patil, A. Miglani, P. K. Kankar, and D. Roy, "Deep learning-based methods for detecting surface defects in steel plates," in *Smart Electrical and Mechanical Systems*, Elsevier, 2022, pp. 87–107. doi: 10.1016/B978-0-323-90789-7.00001-4.
- [36] M. Baygin, M. Karakose, A. Sarimaden, and E. Akin, "Machine vision based defect detection approach using image processing," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya: IEEE, Sep. 2017, pp. 1–5. doi: 10.1109/IDAP.2017.8090292.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015, *arXiv*. doi: 10.48550/ARXIV.1512.03385.
- [38] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," 2024, *arXiv*. doi: 10.48550/ARXIV.2410.17725.
- [39] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022, *arXiv*. doi: 10.48550/ARXIV.2201.03545.
- [40] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019, doi: 10.48550/ARXIV.1905.11946.
- [41] Sampath, Vignesh (2025). CoatingVision: A Defect Dataset for Coating Manufacturing. figshare. Dataset. <https://doi.org/10.6084/m9.figshare.29260121.v1>
- [42] Bryntesen, Silje Nornes, et al. "Opportunities for the state-of-the-art production of lib electrodes—a review." *Energies* 14.5 (2021): 1406.
- [43] Reynolds, Carl D., et al. "Rheology and structure of lithium-ion battery electrode slurries." *Energy Technology* 10.10 (2022): 2200545.
- [44] Reynolds, Carl, et al. "Impact of Formulation and Slurry Properties on Lithium-ion Electrode Manufacturing." *Batteries & Supercaps* 7.2 (2024): e202300396.
- [45] Huang, Tianlun, et al. "Numerical and experimental investigation on the defect formation in lithium-ion-battery electrode-slot coating." *Chemical Engineering Science* 258 (2022): 117744.
- [46] B. Zi, K. Tang, Y. Li, K. Feng, Y. Liu, and L. Wang, "Coating defect detection in intelligent manufacturing: Advances, challenges, and future trends," *Robot. Comput.-Integr. Manuf.*, vol. 97, p. 103079, 2026, doi: 10.1016/j.rcim.2025.103079.
- [47] K. Tang, B. Zi, F. Xu, W. Zhu, and K. Feng, "Coating defect detection method based on data augmentation and network optimization design," *IEEE Sens. J.*, vol. 23, no. 13, pp. 14522–14533, Jul. 1, 2023, doi: 10.1109/JSEN.2023.3277979.
- [48] K. Tang, Y. Li, B. Zi, and K. Feng, "A multimodal unsupervised coating defect detection method based on dual-branch hybrid CNN-Mamba U-Net," *IEEE Trans. Instrum. Meas.*, vol. 74, Art. no. 5026617, pp. 1–17, 2025, doi: 10.1109/TIM.2025.3556907.
- [49] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Red Hook, NY, USA: Curran Assoc. Inc., 2021, Art. no. 924, pp. 1–14.
- [50] H. Cao et al., "Swin-Unet: Unet-like pure transformer for medical image segmentation," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds., *Lecture Notes in Computer Science*, vol. 13803, Cham, Switzerland: Springer, 2023. doi: 10.1007/978-3-031-25066-8_9.
- [51] J. Chen et al., "TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers," *Med. Image Anal.*, vol. 97, p. 103280, 2024, doi: 10.1016/j.media.2024.103280.
- [52] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 6000–6010.

Figure legends:

Figure 1: High-level schematic of the electrode coating process, highlighting slurry preparation, die slot coating, drying, and compaction.

Figure 2: Schematic of the experimental setup and surface defect images captured using a Pixelink PL-D753CU camera with LED lighting.

Figure 3: Workflow of the iterative annotation process for the CoatingVision dataset.

Figure 4: Snapshot of the QC tool interface used by annotators to manually review and correct pseudo-labels, addressing misclassifications and boundary errors in segmentation masks.

Figure 5: Sample images alongside their segmentation masks, with surface cracks highlighted in blue, delamination in green, and pinholes in red.

Figure 6. Visual overview of pinhole defects in the CoatingVision dataset and the associated segmentation masks.

Figure 7: Directory structure of the CoatingVision segmentation dataset, with images and their corresponding masks stored in the 'images' and 'masks' folders.

Figure 8: Histogram depicting the distribution of roundness values for all pinhole defects in the dataset.

Figure 9: Directory structure of the Coating Vision classification dataset, with images in the 'images' folder and annotations in the 'labels' CSV file.

Figure 10: An example of a bounding box and YOLO annotation format.

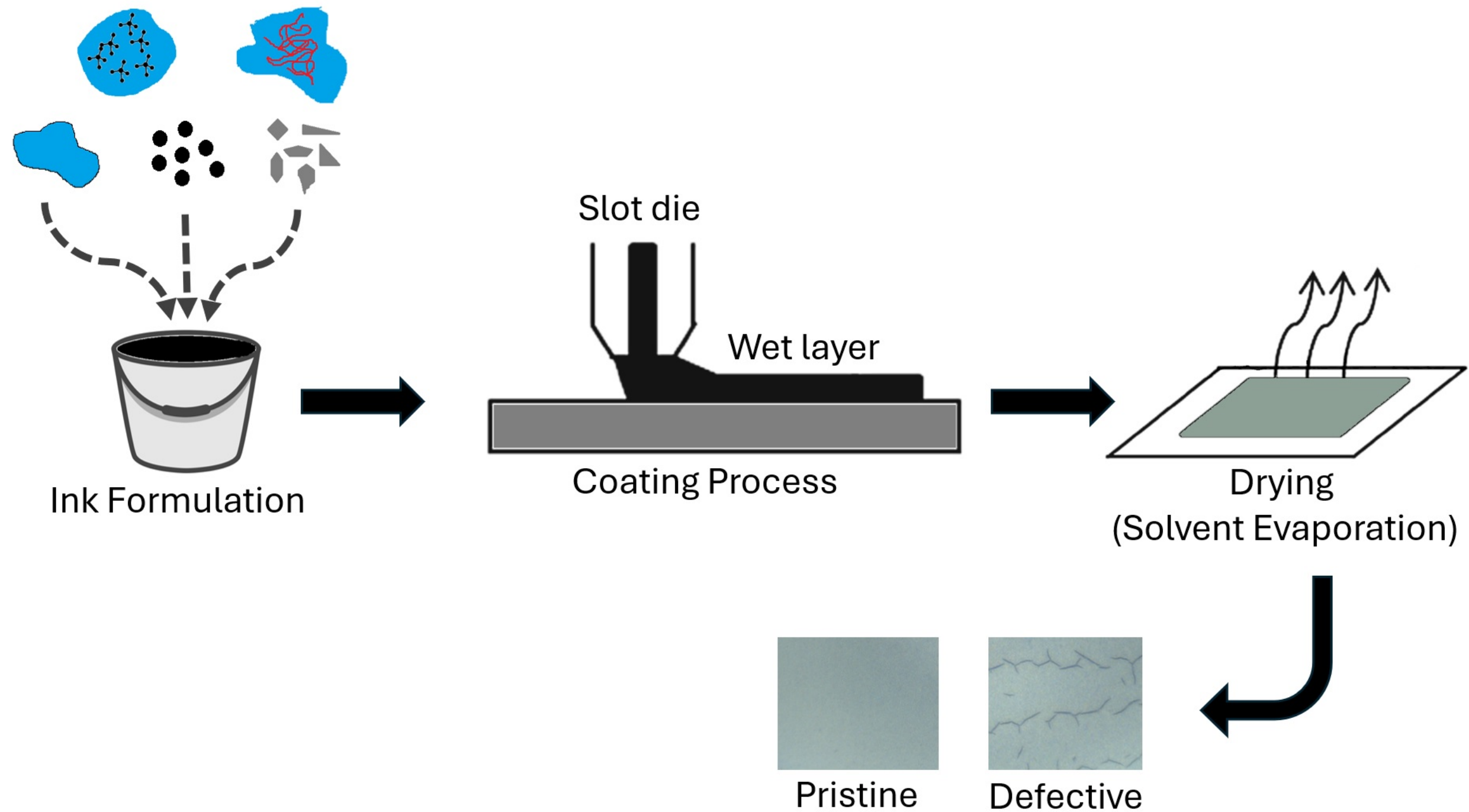
Figure 11: Directory structure of the CoatingVision detection dataset, with images stored in the 'images' folder and corresponding annotation .txt files in the 'labels' folder.

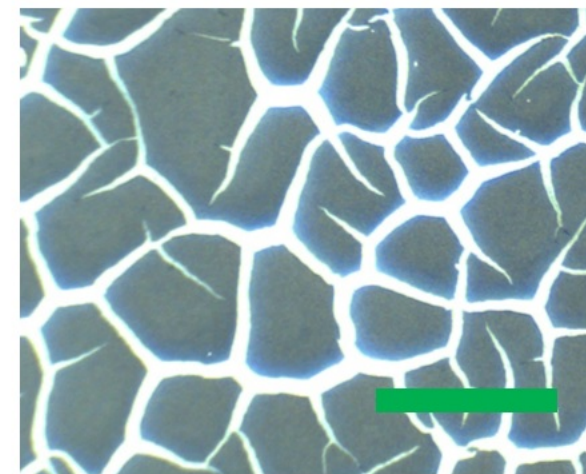
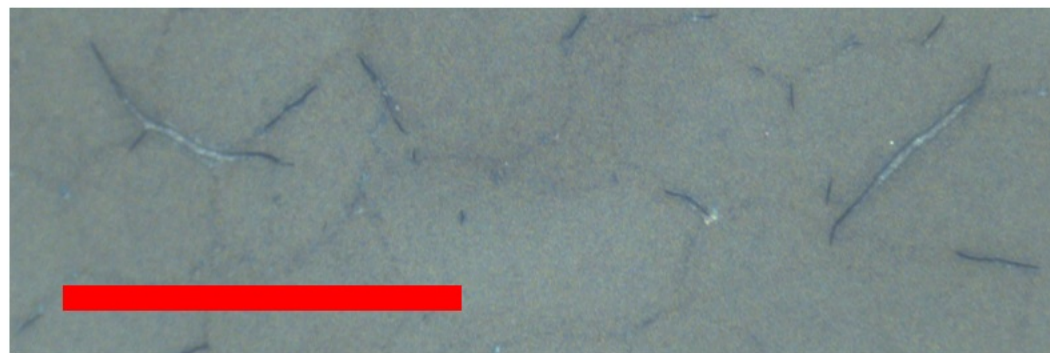
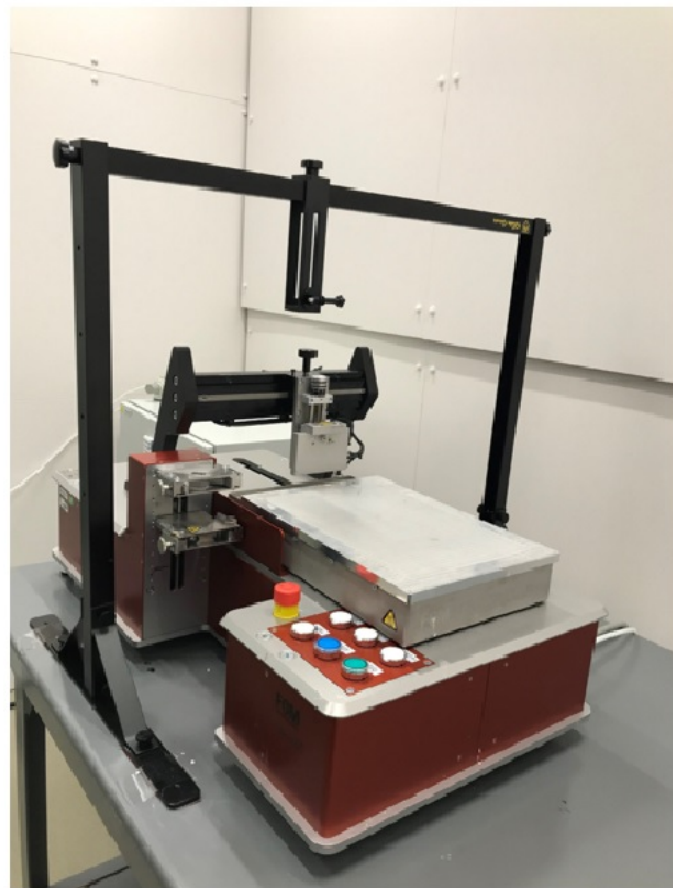
Figure 12: Sample images annotated with bounding boxes around pinhole defects.

Figure 13. Schematic representation of IoU calculation.

Figure 14: Training and validation loss across different models.

Figure 15: Precision-Recall Curve for YOLOV11 across defect Types.

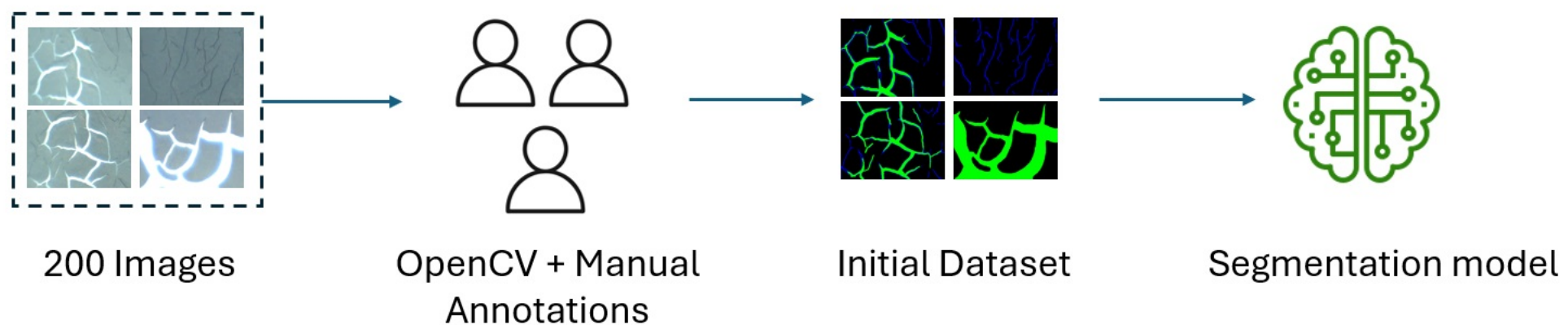




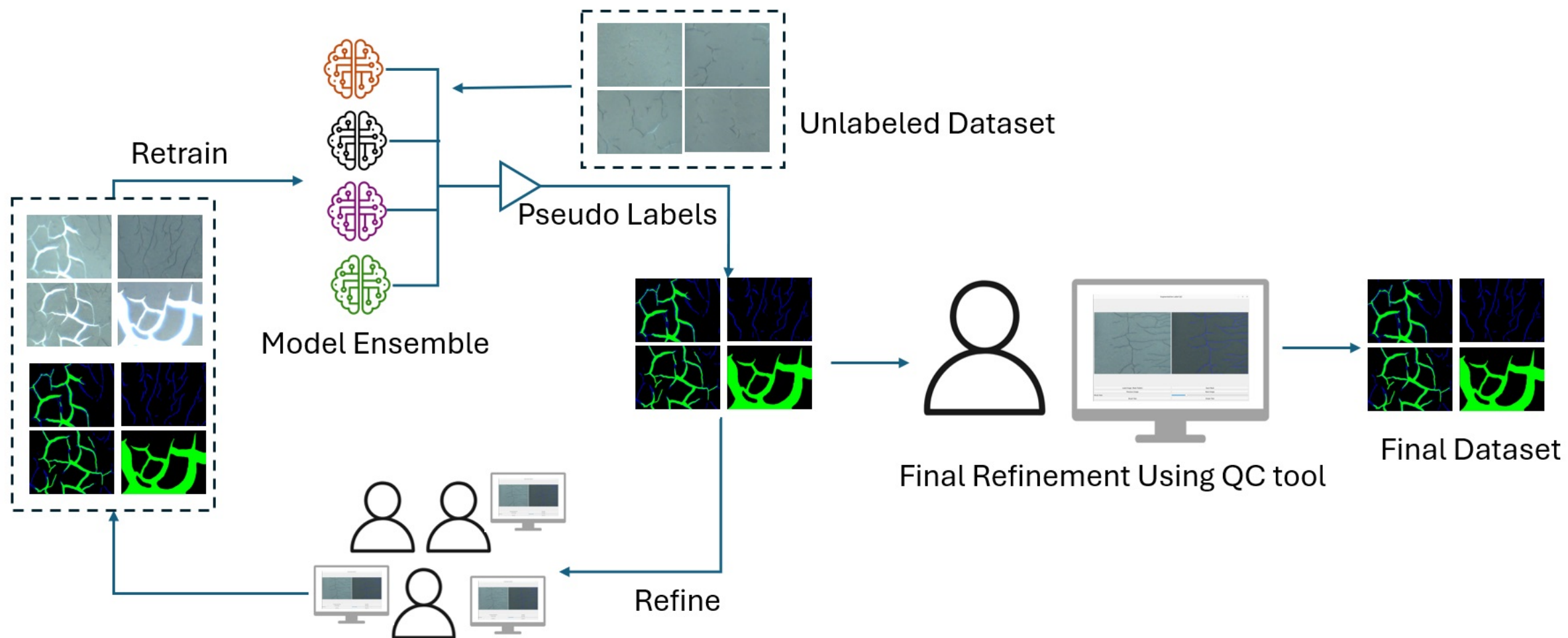
0.5 mm

1.0 mm

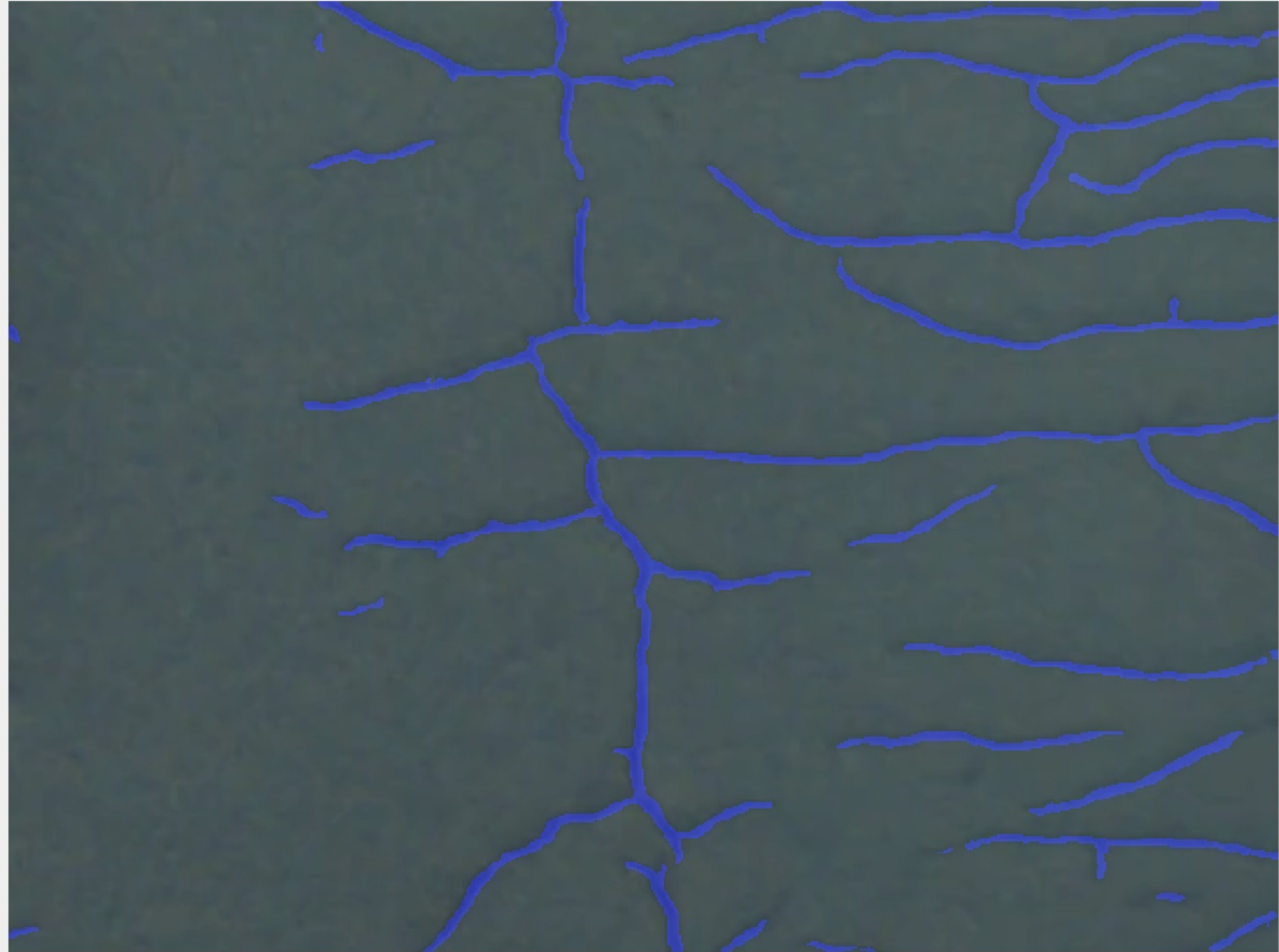
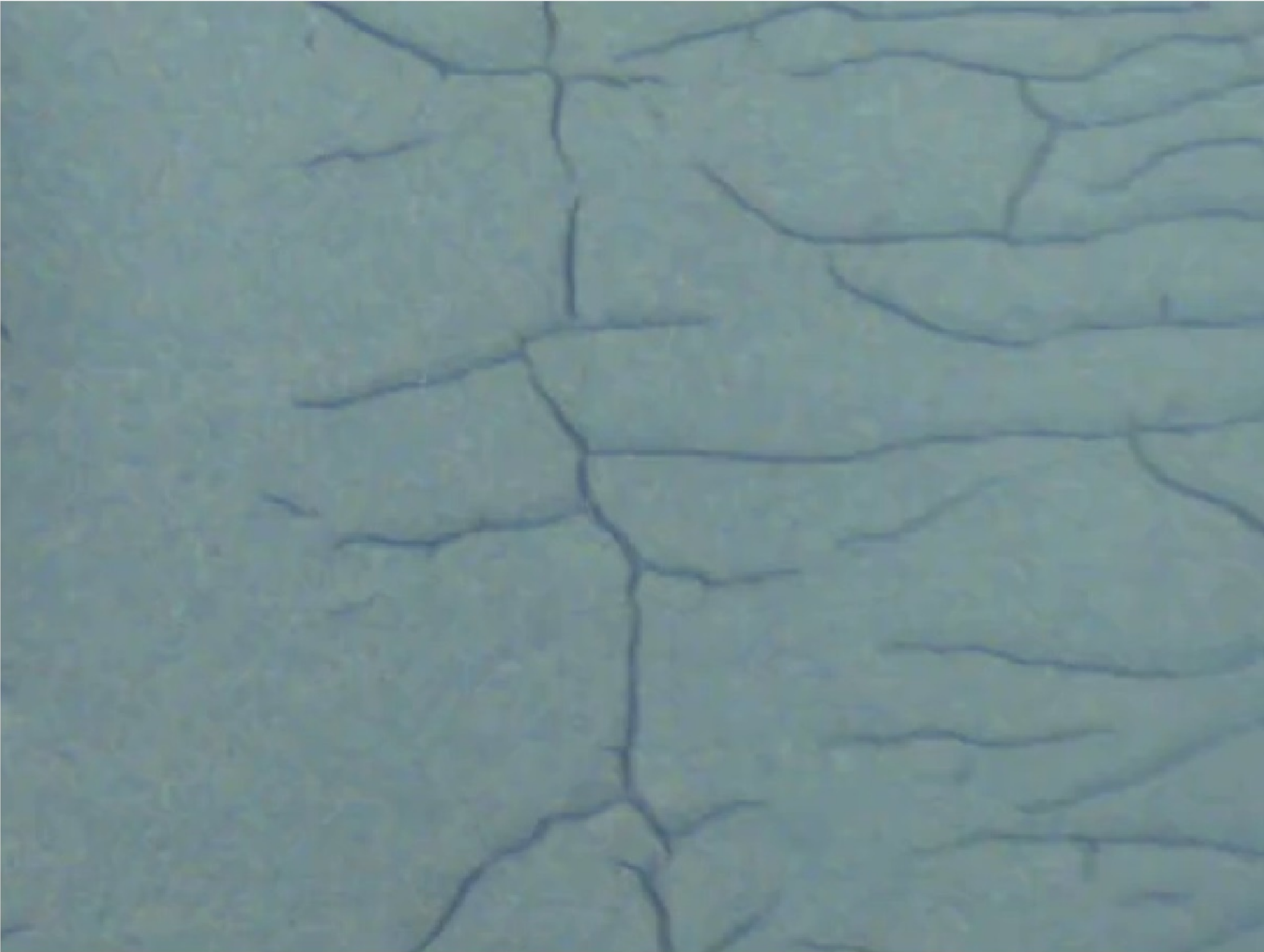
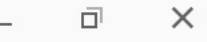
Step 1 & 2: Manually annotate with OpenCV and Initialize model



Step 3,4 & 5: Iteratively train ensemble model, segment and refine labels using QC tool



Segmentation Label QC



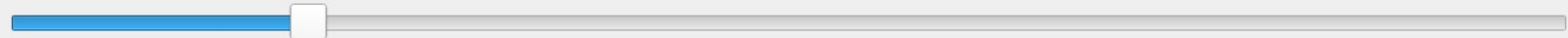
Load Image _Mask Folders

Save Mask

Previous Image

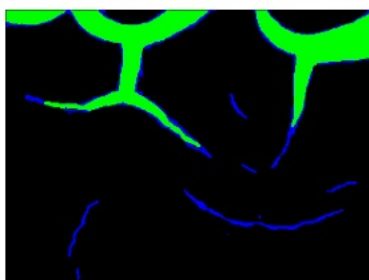
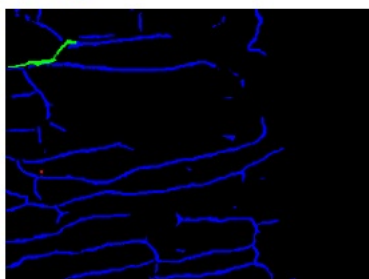
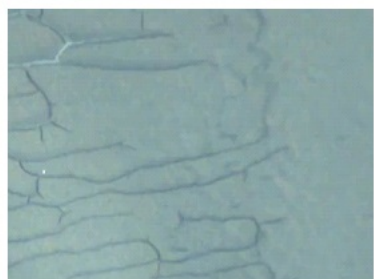
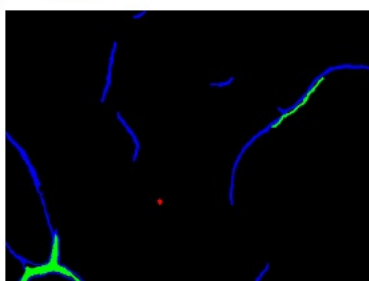
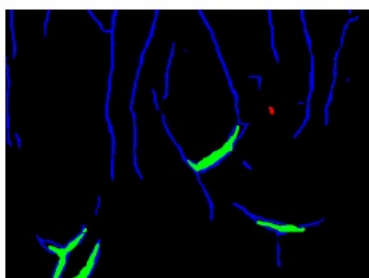
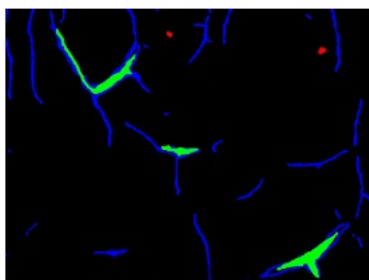
Next Image

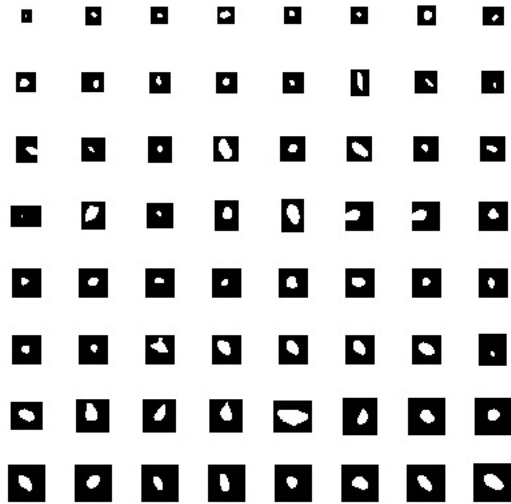
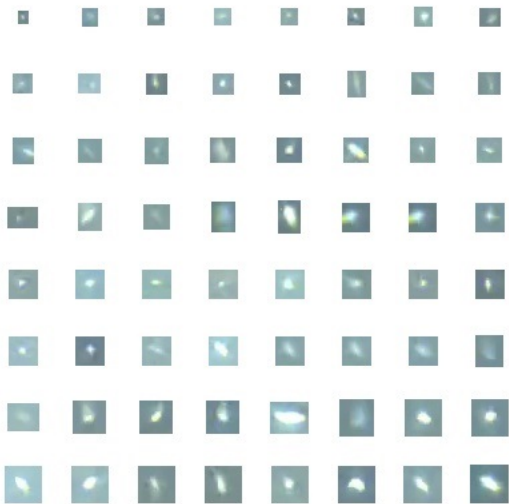
Brush Size
















Brush Tool

Eraser Tool

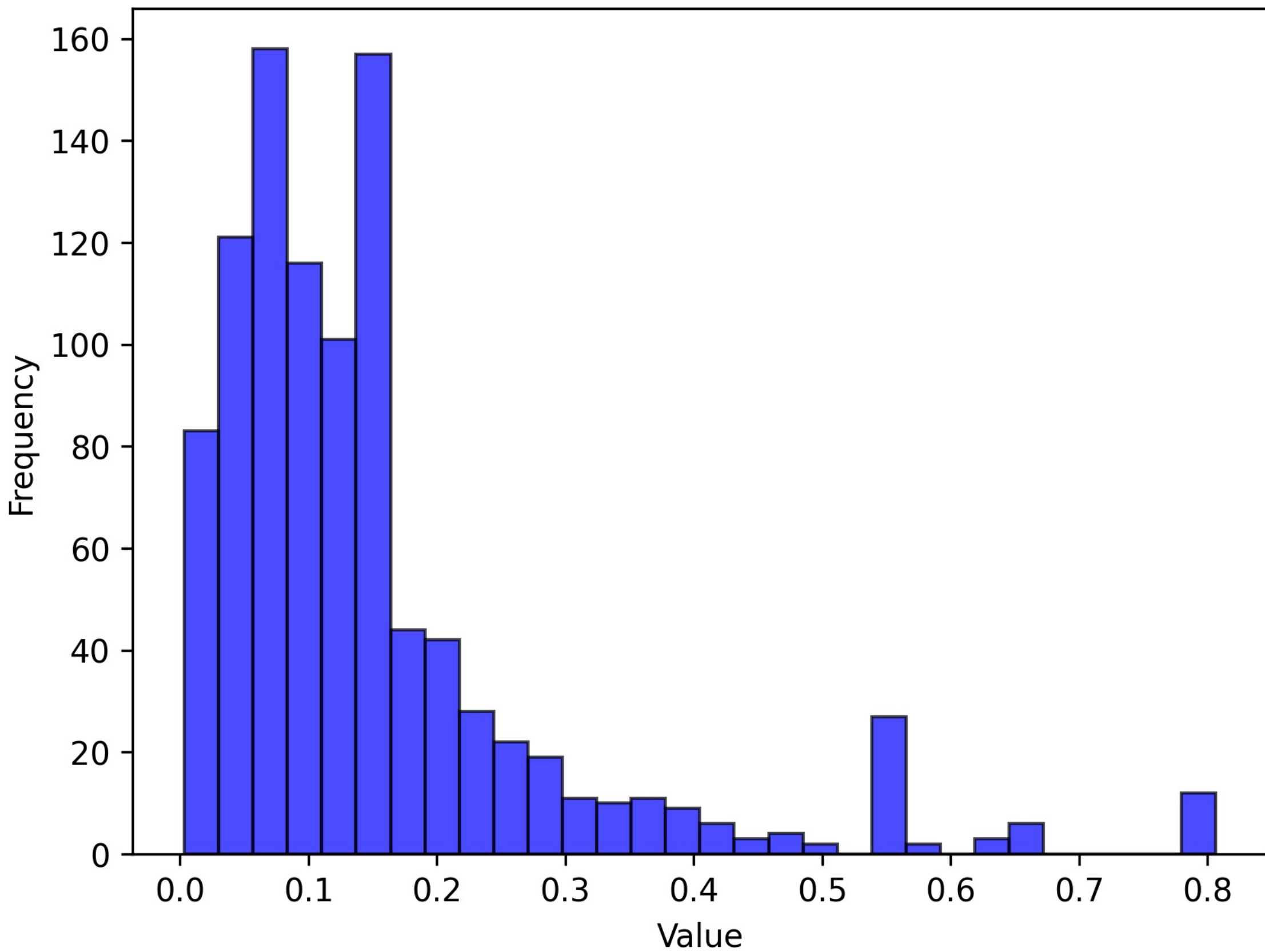













- ▼  segmentation
 - ▼  images
 -  image_1.jpg
 -  image_2.jpg
 -  image_3.jpg
 -  image_4.jpg
 -  image_5.jpg
 - ▼  masks
 -  image_1.png
 -  image_2.png
 -  image_3.png
 -  image_4.png
 -  image_5.png

PRESS














Histogram of roundness values

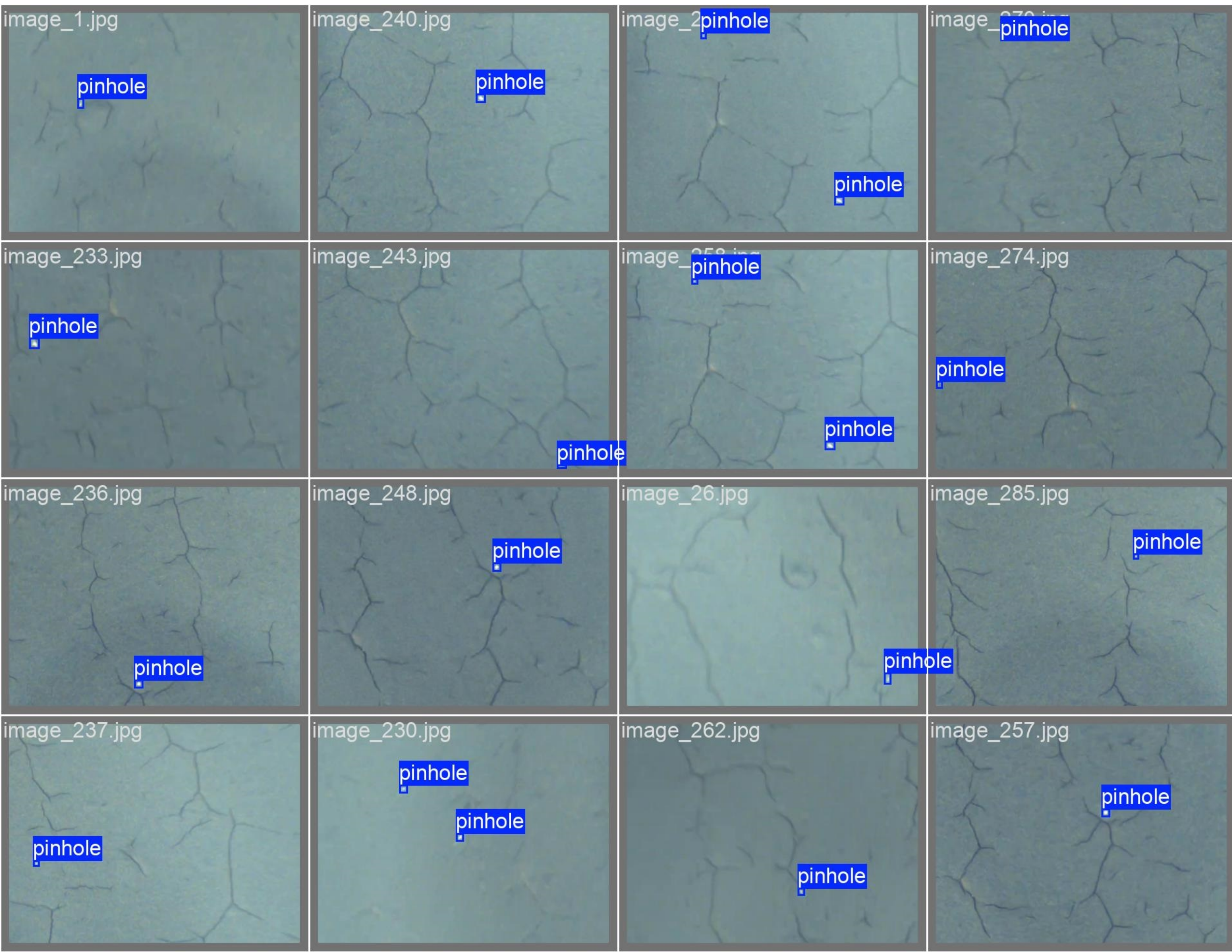


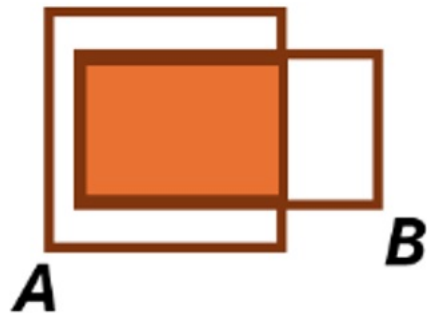
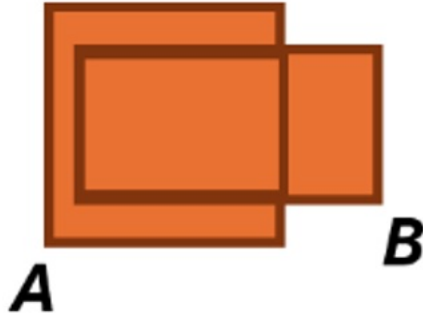
- ▼  classification
 - ▼  images
 -  image_1.jpg
 -  image_2.jpg
 -  image_3.jpg
 -  image_4.jpg
 -  image_5.jpg
 - ▼  labels
 -  annotations.csv



Class_id	x	y	width	height
----------	---	---	-------	--------

- ▼  detection
 - ▼  images
 -  image_1405.jpg
 -  image_1415.jpg
 -  image_491.jpg
 -  image_916.jpg
 -  image_973.jpg
 - ▼  labels
 -  image_1405.txt
 -  image_1415.txt
 -  image_491.txt
 -  image_916.txt
 -  image_973.txt



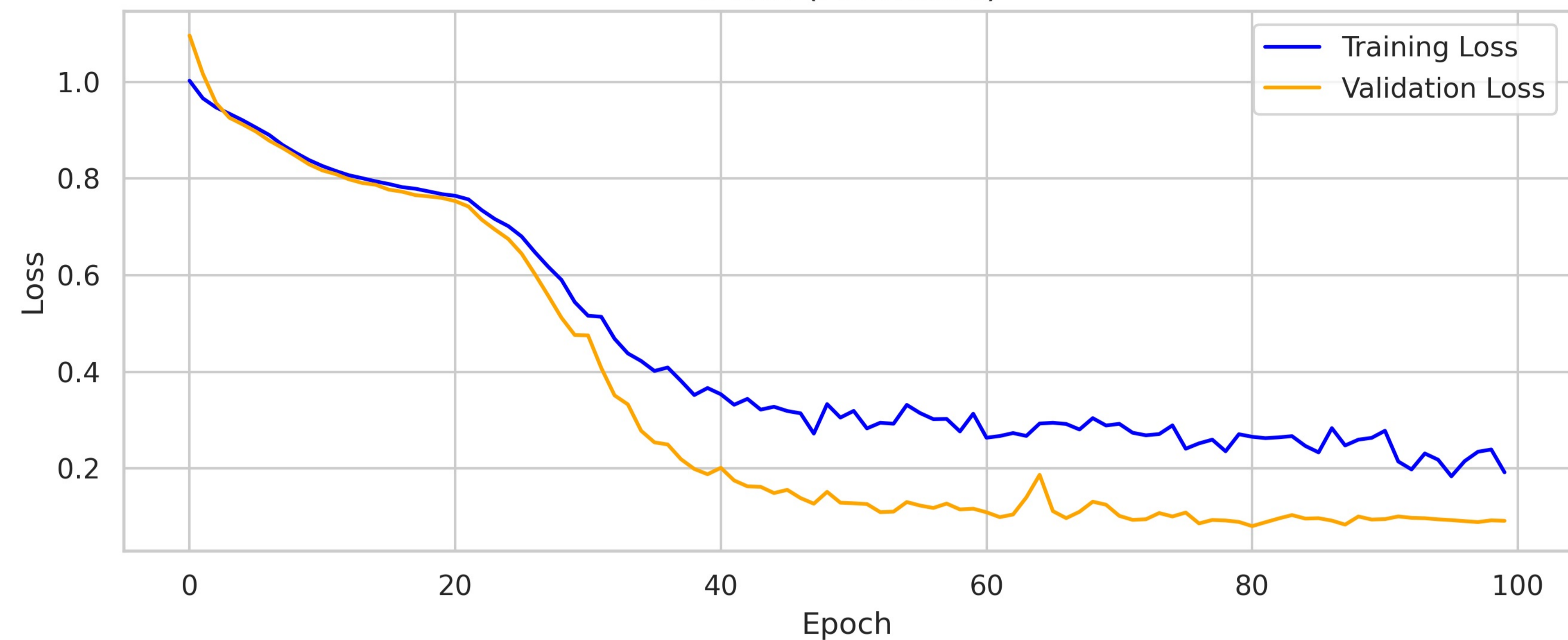
Intersection**Union****Intersection over Union**

$$IOU = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{Intersection}}{\text{Union}}$$

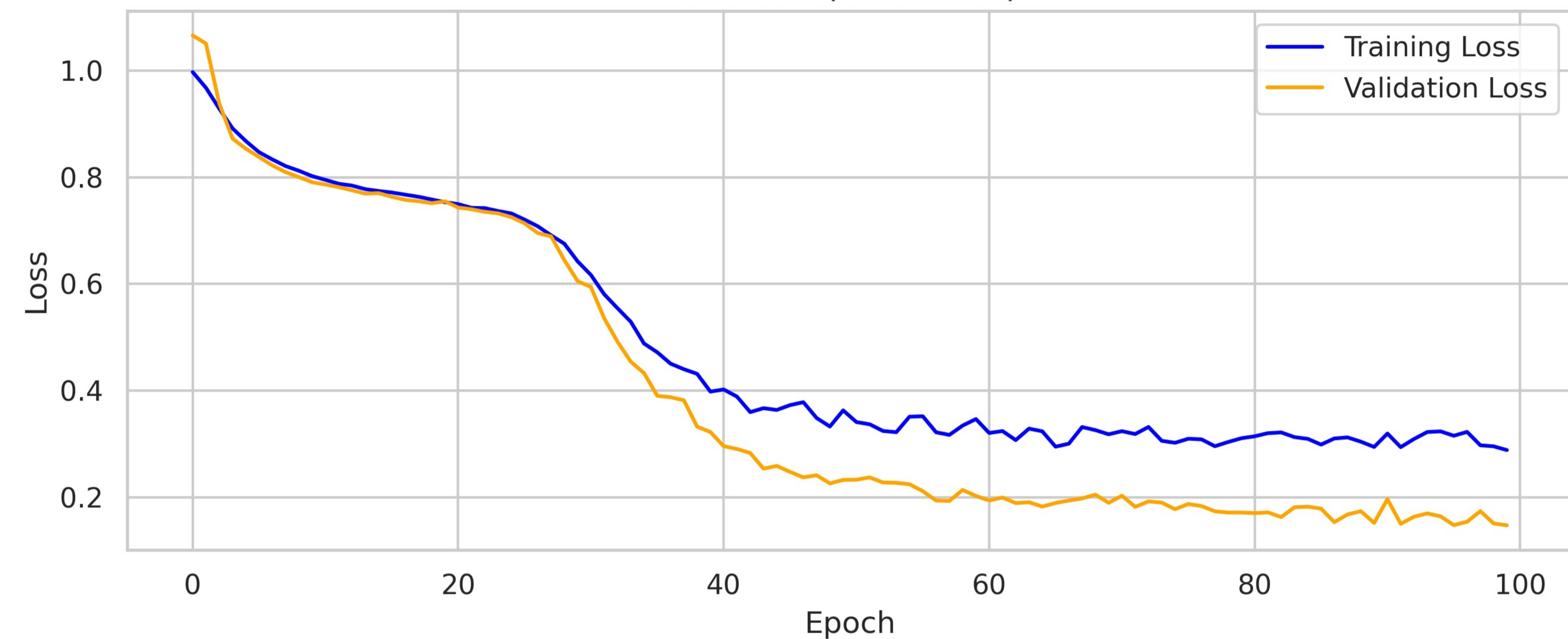
Diagram illustrating the Intersection over Union (IOU) calculation. The numerator is the intersection of two overlapping rectangles, and the denominator is the union of the two rectangles.

Training & Validation Loss Across Models

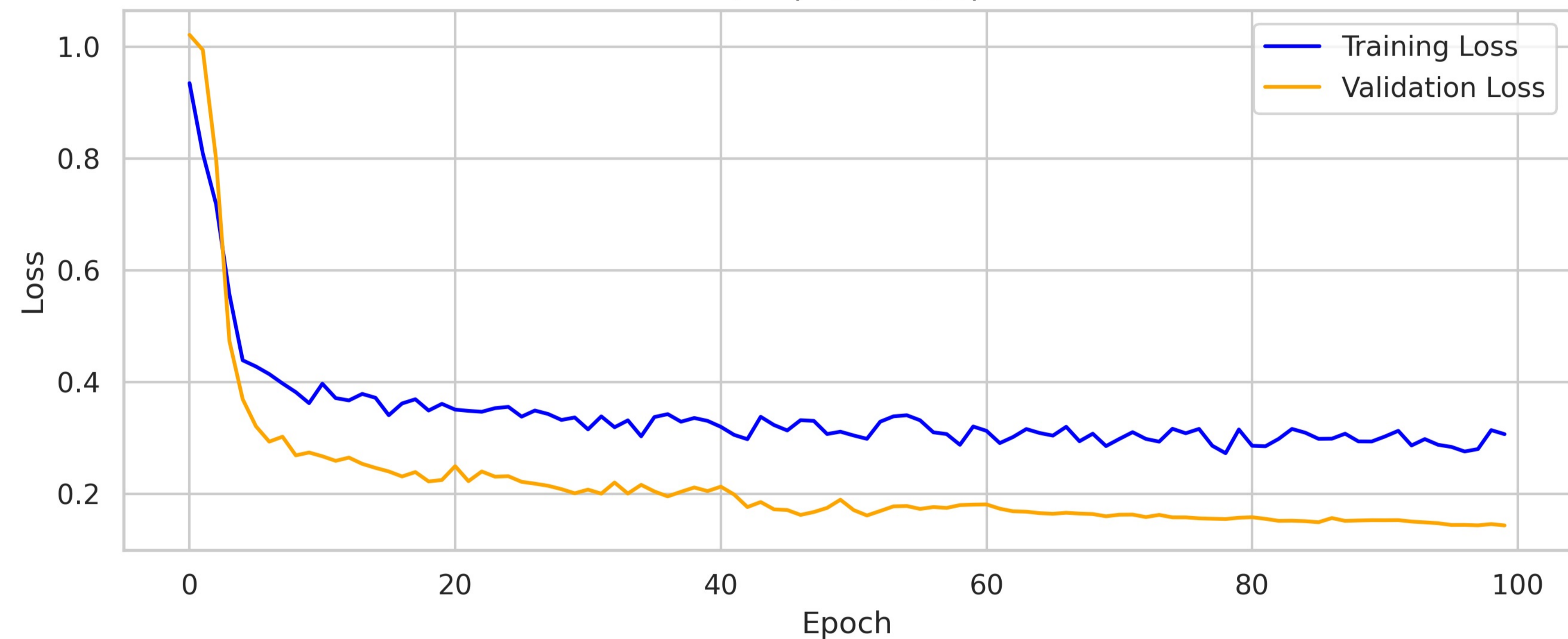
U-Net (ResNet50)



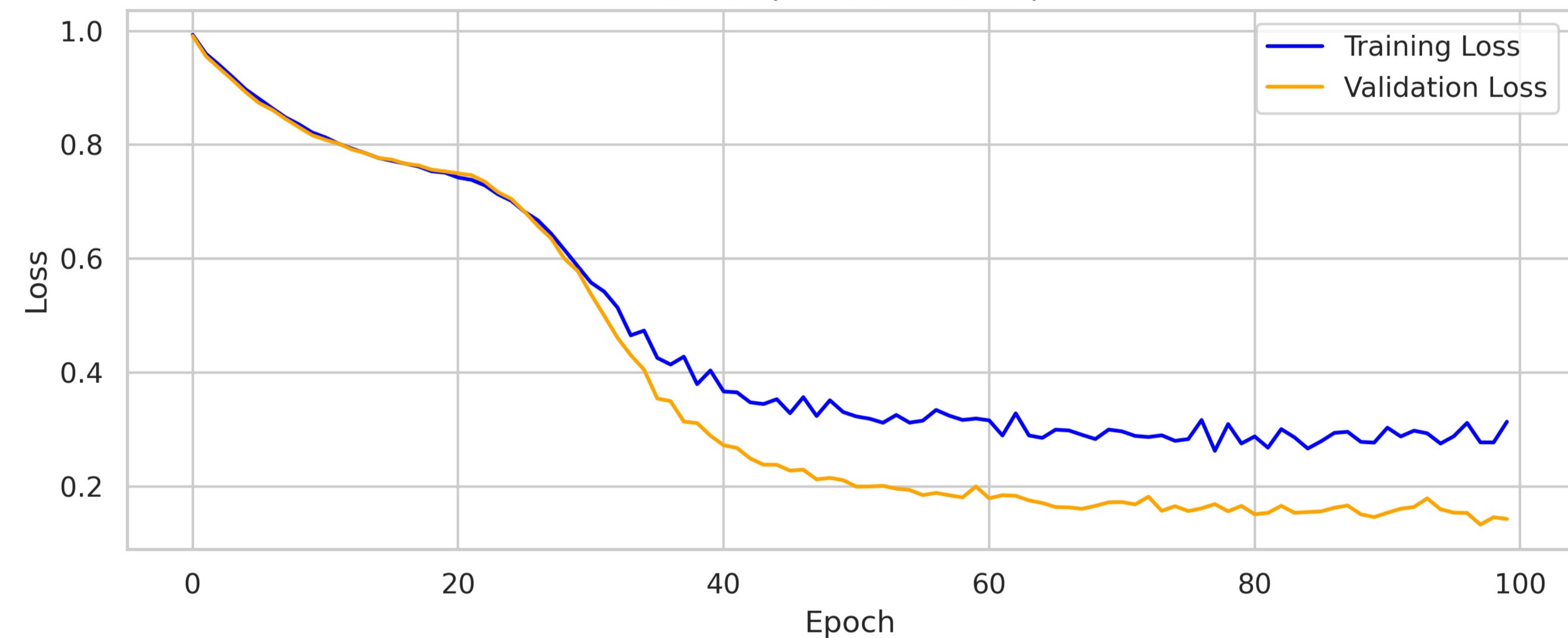
LinkNet (ResNet50)



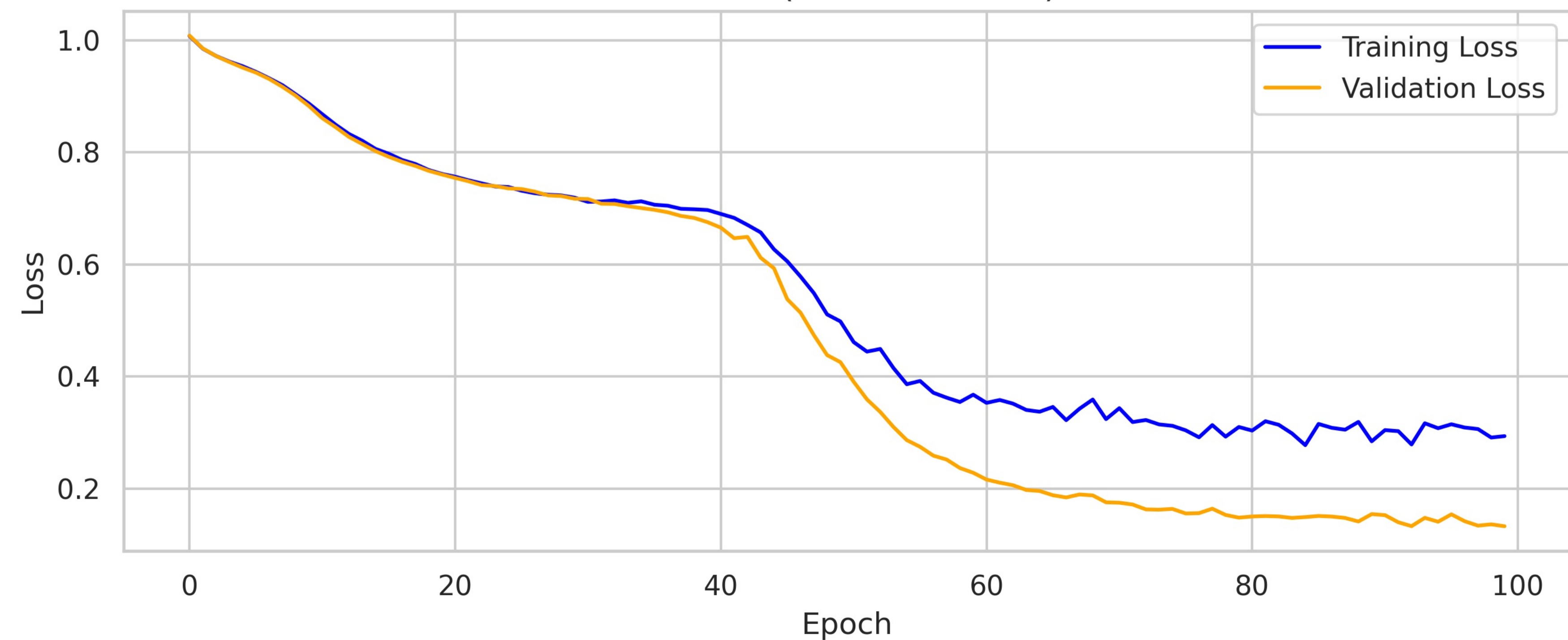
FPN (ResNet50)



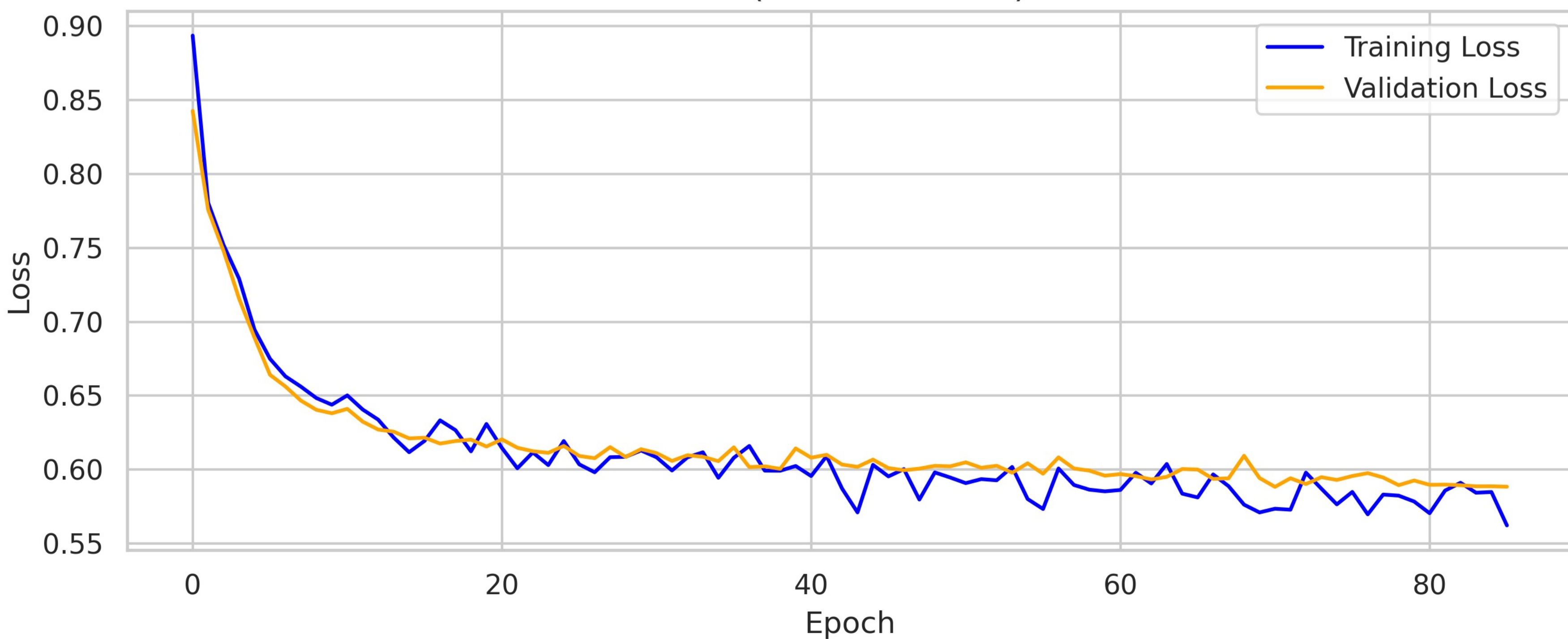
U-Net (EfficientNetB6)



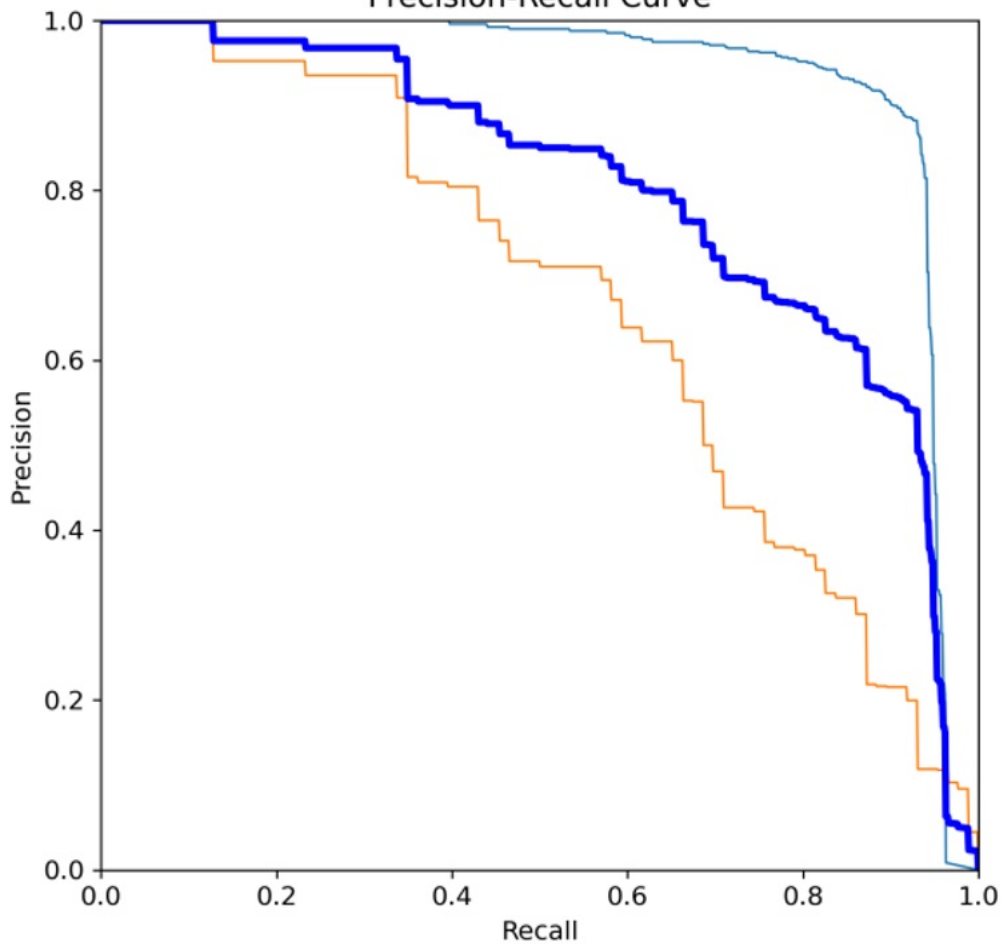
LinkNet (EfficientNetB6)



FPN (EfficientNetB6)



Precision-Recall Curve



pinhole 0.931
unclassified 0.666
all classes 0.799 mAP@0.5