Automated semantic segmentation of NiCrBSi-WC optical microscopy images using convolutional neural networks

Dylan Rose[a], Justin Forth[b], Hani Henein[a], Tonya Wolfe[c], Ahmed Jawad Qureshi[d]

[a]*Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada*
[b]*Consultant, Edmonton, AB, Canada*
[c]*Centre for Innovation in Manufacturing, Red Deer Polytechnic, Red Deer, AB, Canada*
[d]*Department of Mechanical Engineering, University of Alberta, Edmonton, AB, Canada*

Abstract

Convolutional neural networks (CNNs) were used for the semantic segmentation of angular monocrystalline WC from NiCrBSi-WC optical microscopy images. This deep learning approach was able to emulate the laborious task of manual segmentation effectively, with a mean intersection over union (IOU) and a mean dice coefficient (DC) of 0.911 and 0.953, respectively, across the entire test dataset. From the model output, the carbide percent can be determined by dividing the area of positively labelled pixels by the total area of the image. Additionally, the mean free path can be quantified using the method described in ASTM STP 839, and by physically counting the black pixels (CPB) between the particles in the image. Comparing the models predictions to the ground truth, the carbide percent had an average difference of 1.2 area %, while the mean free path differed by 15.7 μm for the ASTM method, and 24.8 μm for the CPB method. The robustness of the model was tested on images containing both spherical eutectic WC and angular monocrystalline WC to determine whether the model was capable of accurately predicting the location of objects that were not part of the training dataset. The U-Net CNN was able to segment the spherical and angular WC with considerable accuracy. These results show that the application of computer vision models for microstructural characterization is not limited to complex imaging modalities, and can be applied to readily available methods such as optical microscopy.

List of Symbols

The following list describes the symbols used throughout this work. The symbols are organized according to field to avoid any confusion with overlapping symbols.

Computational Metallography

λ	Mean free path
f	Volume fraction of reinforcement particles
N_L	Number of particles per unit length

Non-local means filter

σ	Standard deviation of the noise
B	Kernel of pixels
d	Euclidean distance
h	Filtering parameter
i	Color channels of the image
j	Number of pixels away from the center of the kernel
p	Center of kernel being adjusted
q	Center of kernel being used for comparison
r	Radius of the kernel
u	Pixel intensity
w	The weight (value) the pixel is multiplied by

Neural Networks

α	Value of ELU saturation for negative values
β	Weights for a given layer
$\hat{p}$	The models probability that a pixel belongs to a particular class.
λ	L2 regularization term
FN	Type II error (false negative)
FP	Type I error (false positive)
FPR	False positive rate is the ratio of type I errors over the total number of pixels belonging to another class.
TP	True positive (correctly labelled pixels)
TPR	True positive rate is the models ability to correctly classify pixels.
y	Class value of the ground truth

Email addresses: `drose1@ualberta.ca` (Dylan Rose), `justinforth@gmail.com` (Justin Forth), `hhenein@ualberta.ca` (Hani Henein), `tonya.wolfe@rdpolytech.ca` (Tonya Wolfe), `ajqureshi@ualberta.ca` (Ahmed Jawad Qureshi)

Preprint submitted to Elsevier *Thursday 3rd March, 2022*

1. Introduction

Materials used in the mining and processing of natural resources require the ability to resist high levels of abrasive and erosive wear caused by direct exposure to various sands, ores, and rocks. A typical mitigation strategy is the application of a NiCrBSi-WC metal matrix composite (MMC) overlay, using plasma transferred arc welding (PTAW), to prolong the service life of the base part[1, 2]. With the estimated cost of wear in the Canadian industry being \$2.5 billion per year[3], there is a constant push to reduce the downtime of operation by increasing the wear resistance of Ni-WC MMCs. In the conventional manufacturing space, some of the ongoing research to extend overlay lifetime involves altering the: matrix chemistry [4, 2], carbide crystallinity and shape [1, 2], reinforcement/matrix ratio [5], and coating method [6, 7, 8, 9]. Advances in additive manufacturing have extended the capabilities of well-known welding techniques to become fully automated manufacturing processes, where the layer by layer deposition is based on the cross-sectional geometry of a digitally rendered product. Plasma transferred arc-additive manufacturing (PTA-AM) offers the ability to build parts using composite materials (NiCrBSi-WC), removing the requirement for wear-resistant overlays while theoretically enhancing the service life [10, 11].

The underlying philosophy regarding the use of composite materials is combining the benefits of two individual constituents so they may compensate for their individually poor material properties [15]. In abrasive wear applications, the hardness of the WC reduces material loss due to micro-cutting [5] while the ductility and toughness of the NiCrBSi matrix allow for adequate force transfer to the WC particles and significant plastic deformation, adding resistance to brittle fracture [16, 17, 18]. The wear performance of composite materials is also greatly affected by the distribution of the reinforcement particles within the metal matrix. Lack of homogeneity in the distribution of the carbide particles can lead to non-uniform wear rates and premature failure of the coating[19]. Thus, it is imperative to quantify the distribution of the reinforcement phase to optimize the wear resistance of MMCs. The mean free path of the matrix phase has shown a strong inverse correlation to the abrasive wear resistance of the composite material[2, 4, 20, 21]. The mean free path (MFP) can be determined using equation 1[22].

$$\lambda = \frac{1-f}{N_L} \tag{1}$$

The size of the reinforcement particles has also been shown to alter the tribological behavior of the MMC[23, 24], though particle size is unable to be quantified from 2-D projections, as were gathered for this work.

The carbide characteristics (mean free path, and volume fraction) can be determined using quantitative metallography on optical microscopy images [22]. This can be achieved by manually segmenting each carbide particle or semi-autonomously with various image processing techniques to threshold out the carbide particles from the image. Image processing modalities to segment desirable features from an image range from common approaches such as Otsu's method [12] and Canny edge detection [13], to more complex unsupervised learning algorithms like K-means clustering[14]. Image processing methods generally require the areas of interest to have distinct, non-overlapping pixel distributions, and distinct edges; However, the nature of optical microscopy induces significant user variance in the quality of the image attempting to be segmented. Some common technician sources of noise in the pixel distribution include poor metallographic preparation, and sub-optimal focus and contrast. There are also features beyond the equipment's resolution limits (∼0.2 µm [25]), which can contribute further to noise in the pixel distribution. During image processing, noise in the pixel distribution leads to type I (false positive) errors, type II (false negative) errors, and poor edge definition. [26, 27, 28]. Figure 1a displays a 512x512 pixel optical image of Ni-WC, and Figure 1b shows the ground truth image for the segmentation of the WC particles, where the location of the WC particles are highlighted by the white pixels. The segmentation of Figure 1a using Otsu's method [12], Canny edge detection [13], and k-means clustering [14] can be found in Figure 1 c, d, and e, respectively. For Otsu's method [12] the scikit-image [29] algorithm selected 170 as the optimal threshold value. Using Canny edge detection [13] with OpenCV [30], a minimum and maximum gradient value was set to 15 and 63, respectively, with a sobel kernel size of 3. The k-means clustering was done in OpenCV [30] using 3 clusters, max iterations of 100, ϵ of 0.1, and the centroids were initialized using the k-means++ method [31]. The parameters for the image processing methods found in Figure 1 were selected through trial and error, and were analyzed visually. It can be seen that due to the pixel luminance overlap of the features contained within the WC particles, and the surrounding matrix, poor segmentation is achieved when compared to the ground truth (Figure 1b). Although optimization of the image processing algorithms shown in Figure 1 could increase the accuracy of the segmentation, one set of parameters may not work for all images, making it difficult to automate without some degree of manual intervention. The aforementioned shortcomings in traditional image processing, highlighted by Figure 1, show these techniques may not be the best approach for an automated pipeline to segment Ni-WC optical images.

Semantic segmentation is a branch of computer vision that uses machine learning (ML) models to assign a class label to each pixel in an image with an accuracy that can out perform humans in some contexts [32]. Convolutional neural networks (CNNs) use filters that are rastered (scanned) over an image to create a feature map [33, 34]. A feature map is a tensor representing the multiplication of the features in the image, or input feature maps, by the weights in the kernel of a given layer; this can give insight into the signal that the network has learned to be important to the task at hand. Each filter represents a single layer in the network, and are organized in blocks of multiple layers that have similar spatial dimensions. After each filter passes over the image, the feature activation values are passed through an activation function to account for the nonlinearity of the image pixel data[35]. The last stage of a convolution block

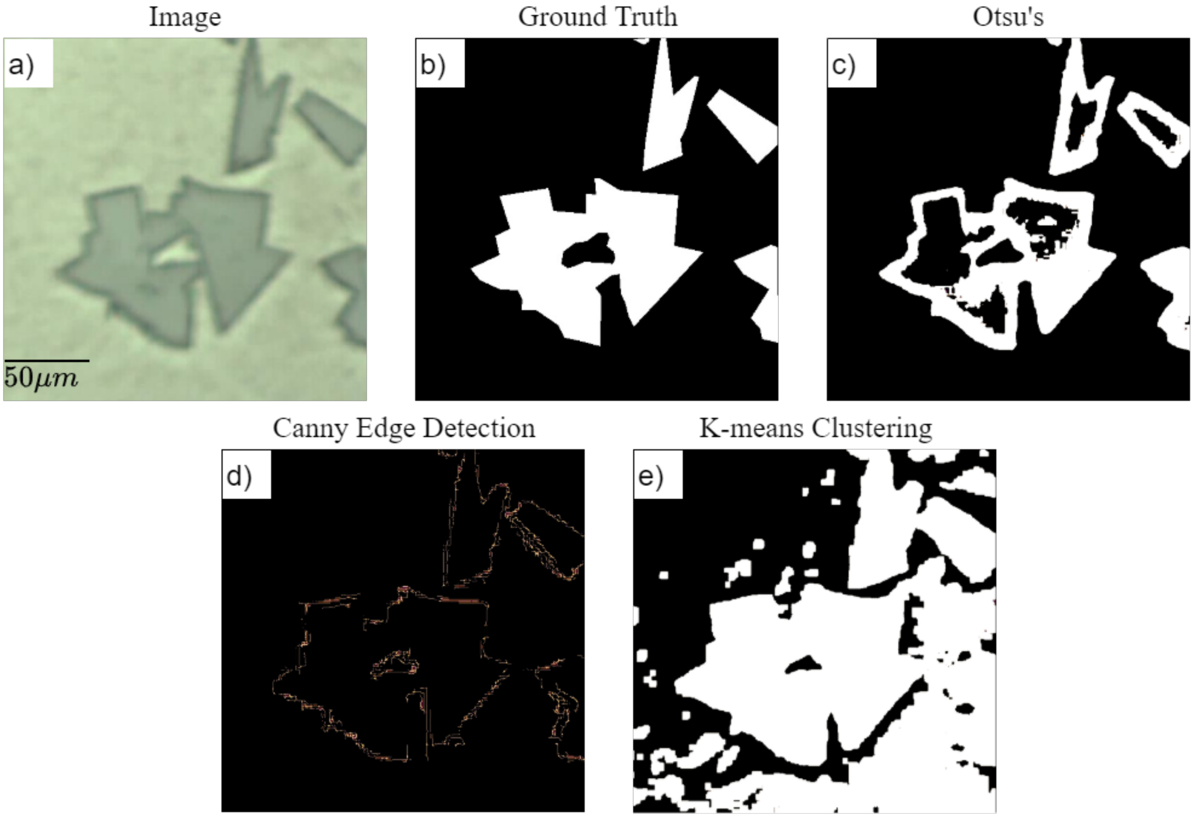

Figure 1: a) NiCrBSi-WC optical image, where the dark green particles represent the WC and the surrounding lighter portion is the Ni matrix, b) the ground truth corresponding to image a), where the location of the WC particle is shown in white; c) thresholding of image a) using Otsu's method [12]; d) thresholding of detectable edges in image a) using Canny edge detection [13]; e) segmentation of image a) using K-means clustering [14].

is downsampling, which decreases the (x,y) dimensionality of the output by reducing a rectangular kernel of pixels to a single value (typically the max value of the kernel[36]). The number of convolutional blocks is based on the depth and architecture of the model being used. For image classification, the final layer from the encoder is flattened to a feature vector, which is then used for the classification stage [37, 38]. On the other hand, for semantic segmentation, the activations from the final layers of the encoder must be restored to the original size of the image, using a decoder network, to perform pixel-wise classification while replicating the original dimensionality of the image[39]. Typically these models are developed to classify: people [40, 41], natural objects [33, 42, 34, 40], and biomedical phenomena[43]. Semantic segmentation models' ability to extract features in an image extends beyond their original intent and can be used for more abstract concepts such as phase segmentation in microstructural images [44, 45, 46, 47, 48, 49]. The development of easy-to-use APIs [50, 51], makes the application of computer vision models far easier for researchers outside the immediate industry.

The present study aims to test state-of-the-art CNNs for the semantic segmentation of Ni-WC MMCs deposited using PTA-AM. The effect of varying the hyperparameters, as well as the encoder architecture, will be discussed. Following the segmentation of the carbides, the quantification of the carbide volume percent, and mean free path (MFP) can be determined. The computational uncertainty associated with the quantification of the microstructure and the carbide characteristics will be disclosed, and the ability for the trained CNN to be able to segment more complex optical microscopy images is shown.

2. Materials and Methods

The framework for the experimental work conducted in this study can be found in Figure 2. Each step in the framework corresponds to a chronological sub-section of Materials and Methods, where the details of each step will be discussed. A high-level overview of the section is as follows. The first step was image acquisition which included the PTA-AM deposition of Ni-WC MMC deposits, the sample preparation, and dataset generation through optical microscopy. The images were preprocessed with non-local means denoising and pixel mean shift algorithms prior to being passed through the CNN. The CNNs had to be optimized for the task of semantic segmentation of Ni-WC optical microscopy images. This included a hyperparameter search, k-fold cross-validation, and receiver operating characteristic curve analysis. Finally, the predictions from the CNN are subject to various post-processing methods before the final evaluation.

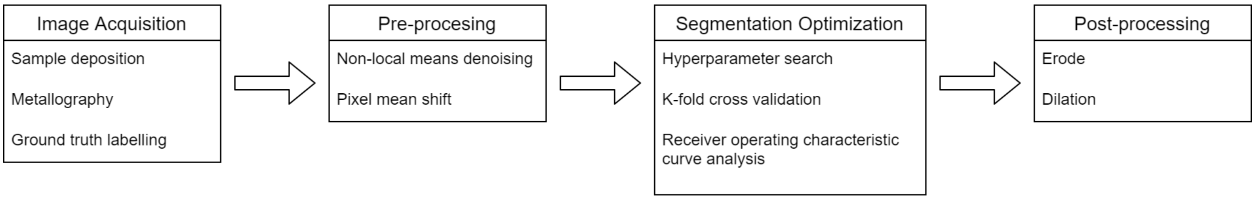

Figure 2: Framework for segmentation pipeline used for this study.

2.1. *Image Acquisition*

The Ni-WC samples were deposited using the PTA-AM system developed by Rojas *et al.*, consisting of a Kennametal Stellite ^{TM} STARWELD 400A PTA and a positioning table capable of moving 365mm x 170mm x 300mm in the X, Y, and Z plane, respectively [10, 11]. The powdered feedstock used for the samples contained in the training data was 60wt% (45 vol.%) angular monocrystalline WC, with the remaining 40wt% (55 vol.%) being a NiCrBSi matrix with a hardness of 30 HRC. Additional samples were deposited to test the transferability of the trained CNN using a powdered feedstock of 35wt% (∼26 vol.%) angular monocrystalline WC, 35wt% (∼26 vol.%) spherical eutectic WC, and 30wt% (48 vol. %) NiBSi. The size distribution for the angular and spherical WC were (-180, +63µm), and (-180, +125 µm), respectively. The single track multi-layer deposits consisted of 40x0.75mm layers resulting in sample dimensions of 100mm x 7mm x 30mm for the length, thickness, and height, respectively. Following metallographic preparation, the samples were observed under an Olympus PMG3 optical microscope, and images were captured using an Olympus Q color 5 camera with Quartz PCI V5 software. An example of a larger sample image taken at 50x magnification can be seen in Figure 3. From the larger images, 256x256 pixel crops were taken to increase the size of the dataset to a total of 229 images. The difference in density between Ni (8.9 gcm^{-3}) and WC (15.6 gcm^{-3}) caused there to be settling of the WC particles, which is evident in Figure 3. The settling of the WC caused the class balance of the images to vary depending on where the cropped image originated in the sample. When the images were split into training and test sets, it was ensured that the class balance was as close to the same as possible. The images were all taken at the same magnification, as the magnification would effect the resolution of features in the images, impacting the final results.

The ground truth for each image was made by manually tracing each carbide in the images using the labelme [52] python package and exporting the recorded vertices to a binary image. An example of a cropped image and its corresponding ground truth can be seen in Figure 1a and Figure 1b, respectively. Pores, cracks, and defects were counted as being part of the background class to simplify the problem to be binary segmentation. During the initial experimentation, it was evident that the increased computational ~~complexity~~ of a multi-class system was ~~too much of an out of discipline leap~~. The end goal was to quantify the WC distribution in the composite, providing a metallurgical justification for the improved wear resistance, with an automated pipeline with base level complexity, that can be built upon with future iterations. Although simplicity was conserved, including microstructural artifacts as part of the background class shows a lack of foresight to the potential ramifications of this decision on the calculation of mean free path. An additional ~~poor~~ ground truth decision was not ensuring the shadowing around the WC particle was not included in the mask. From a metallurgical standpoint, the error or including this boundary is insignificant and the occasional inclusion makes little difference to the overall quantification of the WC distribution~~; However, the effects of this decision from a machine learning perspective were unbeknown. The lessons learned from the above ground truth decisions are important part of out of discipline experimentation.~~

2.2. *Pre-processing*

The first pre-processing step was done to reduce the noise induced by the phase structure of the Ni matrix material, and reduce the variability in the pixel intensities belonging to the WC. This has been shown to decrease the generalization error, and may allow the user to reduce the depth of the model required to perform the task [53]. A pixel-wise non-local means filter [54] was used for denoising, which attempts to reduce the variance between similar rectangular kernels found throughout the image. An example of the results of the non-local means filter on a Ni-WC optical micrscopy image is shown in Figure 4 , where the black square represents the kernel containing the pixel p being adjusted, and the red and purple squares are examples of kernels that would be used to adjust p. The pixel values provided in Figure 4 are samples of the pixel intensities from the green color channel for each region, and the red pixel represents the pixel p, while the green pixel is the pixel q. The first step is determining the squared Euclidean distance between the like patches, shown in equation 2. An exponential kernel, shown in equation 3, is then used to calculate the per pixel weight factor. Due to the exponential nature of the weight, distance values less than $2\sigma^2$ result in a value of 1, while more significant variances between the patches are adjusted exponentially[54]. The weights are then used in equation 4 to adjust the value of the pixel p. Referring back to Figure 4 the euclidean distance between the pixels belonging to the black kernel and purple kernel, will be smaller than the red kernel; This means the purple kernel will have a larger influence on the pixel value of p. Following the adjustment, the center of the kernel is moved the adjacent pixel, and the process is repeated until the entire image has been analyzed. The decision to use the non-local means filter coincided with the use of image augmentation in the early stages of the model development; This prevents the decoupling

replace crossed out text with this

considered overly complex at thsi stage

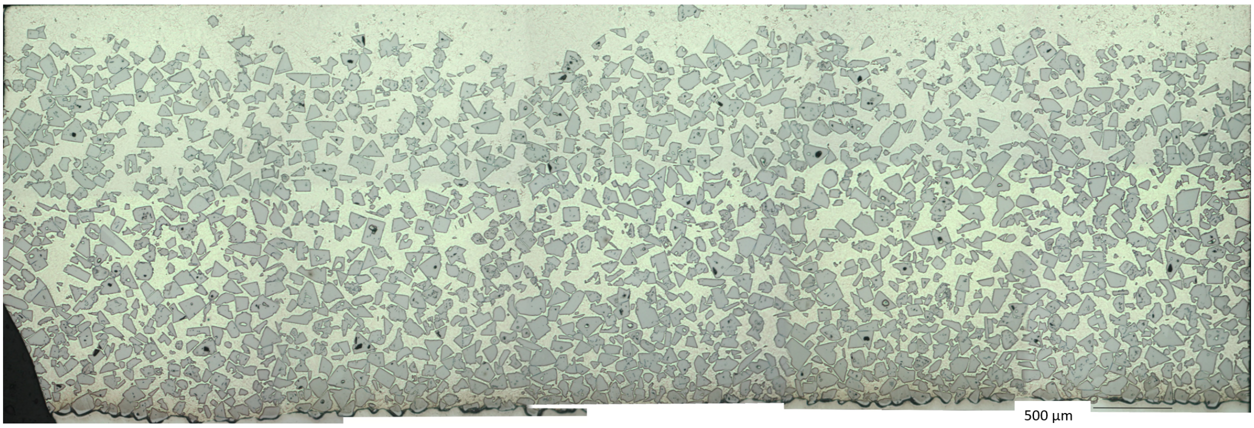

Figure 3: An example of a large optical microscopy image used to compile the training and test datasets. The above image is 9138W x 3322H pixels, meaning that 468 256x256 pixel crops can be taken from one image.

of the contribution from the non-local means pre-processing. However, convergence was unable to be achieved prior to the implementation of these pre-processing methods.

$$d^2(B(p,r),B(q,r)) = \frac{1}{3(2r+1)^2}\sum_{i=1}^{3}\sum_{j\in B(0,r)}(u_i(p+j) - u_i(q+j))^2 \qquad (2)$$

$$w(p,q) = e^{-\frac{max(d^2-2\sigma^2,0.0)}{h^2}} \qquad (3)$$

$$\hat{u}_i(p) = \frac{1}{\sum\limits_{q\in B(p,r)} w(p,q)} \sum_{q\in B(p,r)} u_i(q)w(p,q) \qquad (4)$$

Following the denoising, the mean and standard deviation of any outlier input image was adjusted to match the pixel distribution of the training data. If the mean of any of the color channels of the input image were greater than 20 away from the mean of the same channel in the training set, all channels of the input image were normalized to match the distribution of the training data. The model only has the capability to generalize features based on the data available during training. The small dataset used during training limited the variability of the features belonging to the carbide class. Normalizing the outlying images increased the mean IOU by 5%.

Image augmentation was employed to expand the size of the dataset used during training in order to increase the model's ability to generalize across the range of features that belong to the carbide class, and reduce the chances of overfitting to the training data[55]. Prior to be fed into the CNN, the images were randomly augmented with either a horizontal or vertical shift by 51 pixels, a horizontal or vertical shearing of the image by 5°, a rotation by 25° both clockwise and counterclockwise, a brightness shift by 0.8-1.2, and a shift in the intensity of one of the three color channels by 10 (4%). Augmentations were chosen to account for user errors that occur during the procurement of images using an optical microscope.

2.3. *Optimization*

This section of the paper will outline the process taken to optimize the CNN for the semantic segmentation of WC particles in Ni-WC optical images, which includes: Neural network selection, hyperparameter search, and k-fold cross-validation.

2.3.1. *Neural Network Selection*

The primary convolutional neural network architecture used for this work is the U-net developed by Ronneberger *et al.* [43], which was originally developed for the segmentation of cell structure images for biomedical applications. A visual depiction of the U-Net is shown in Figure 5. The selection of this architecture was primarily due to the similarity in the features between cell structures and microstructural images, since both contain areas of interest surrounded by background pixels with a similar pixel distribution. Additionally, the U-Net was developed for optimal performance with limited data (30 training images [43]) available for training. Architecturally, the U-net has a symmetrical encoder and decoder. In the context of CNNs, the encoder creates low resolution feature maps from the input image, and the decoder increases the resolution of the feature maps to perform pixelwise classification [56]; both the encoder and decoder use trained parameters. The encoder consists of a repeated sequence of two 3x3 unpadded convolutions rastered across images and their derived feature maps, each followed by an activation function (eg. ReLU[57]), and then downsampled using a 2x2 max pooling with a stride of 2. The convolutional, and pooling layers are illustrated in Figure 5 by the purple, and blue rectangles, respectively. Following each downsampling step, the amount of feature maps is doubled to increase the models' ability to learn complex structures. The increase in depth and decrease in the size of the output increases the model's ability to extract features from the image but sacrifices the spatial context of these features. Thus, to expand the dimensionality of the output to contain precise pixel-wise predictions, the model requires a symmetric decoder path. Each upsampling block begins with a transpose convolution, causing a reduction in the depth of the output while increasing the size, shown by the red rectangles in Figure 5. The upsampled feature

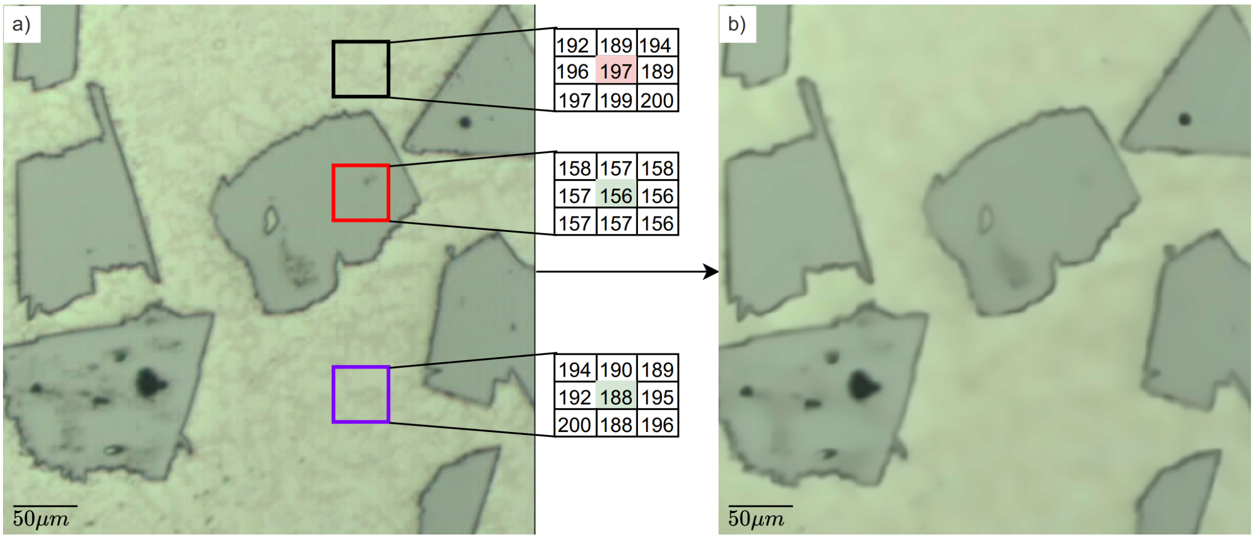

Figure 4: a) The input image before non-local means filtering and b) post non-local means filtering. The black square is the kernel surrounding the pixel p that is being adjusted, which is shown in red. The red and purple kernels are used to calculate the weight by which p will be adjusted, with the green pixel denoting the center q. The euclidean distance between the pixels belonging to the purple kernel is lower than that of the red kernel, causing the purple kernel to have a larger weight than that of the red kernel.

map is concatenated with the feature maps from the encoder block of the same depth, increasing the spatial precision of the predictions. Similar to the encoder, the combined feature maps and transposed convolution is passed through two 3x3 convolutional layers, each followed by a ReLU. The final layer contains a 1x1 convolutional layer, which maps the final feature vector to the corresponding number of classes [43], shown in green in Figure 5.

LinkNet, developed by Chaurasia *et al.*[58] and shown in Figure 6, was also used for the segmentation of WC particles from Ni-WC optical images and compared with the results from the U-net. Convolutional layers used in this architecture are all followed by batch normalization, where the activations are rescaled to have a mean of 0 and a standard deviation of 1 [59], and a ReLU activation function. The convolutional layers are shown in green, and the relu is shown in purple in Figure 6. The encoder begins with a 7x7 convolutional layer, followed by a 3x3 max pooling with a stride of 2. This output enters four consecutive residual blocks [60], each containing two sets of 3x3 convolutional layers. Every two convolutional layers are followed by a skip-connection, where the input and output of the two convolutional layers are combined to reduce the likelihood of a vanishing gradient without impacting the model's performance. The residual connections are highlighted by the combination of the blue arrows, and purple spheres in Figure 6. Furthermore, the depth of the model can be increased without sacrificing its ability to learn identity-like mappings [60]. The decoder path includes four blocks containing two 1x1 convolutional layers, with a 3x3 transpose convolutional layer in between. The output of each decoder block is combined with the input from the corresponding encoder block to recover the spatial knowledge that was lost during feature extraction; These are illustrated by the red arrows in Figure 6. Before localization, the output of the decoder is passed through two more transpose convolutional layers, with a 3x3 convolutional layer in between [58].

Along with the original architectures the encoders from VGG16[34], resnet50[60], and mobilenet[61] were also tested. The corresponding decoders were designed to conserve the original U-net and LinkNet structure. The final layer activation was achieved using the sigmoid activation function, which outputs the class probabilities between 0 and 1, shown in equation 5 [62]. The sigmoid activation was used for all models analyzed in this work.

$$f(x) = \left(\frac{1}{1+exp^{-x}}\right) \tag{5}$$

Keras[63], and Tensorflow [51] were used as the machine learning API for all the models in this work. The python/Keras version of the models was written by Yakubovskiy *et al.*, and provided in the Segmentation Models Github repository [64]. Computing resources were provided by Industry Sandbox and AI Computing (ISAIC) at the University of Alberta, which included 2 16GB NVidia Tesla V100 GPUs.

2.3.2. Performance Metrics

The two performance metrics used in this work were the Jaccard index [65], also known as intersection over union (IOU), and the dice coefficient[66](DC), shown in equations 6 and 7, respectively.

$$IOU = \frac{TP}{TP+FP+FN} \tag{6}$$

$$DC = \frac{2TP}{2TP+FP+FN} \tag{7}$$

Although their formulation is similar and the metrics are positively correlated, there are some differences when the evaluation is across many instances. Though both the IOU and dice

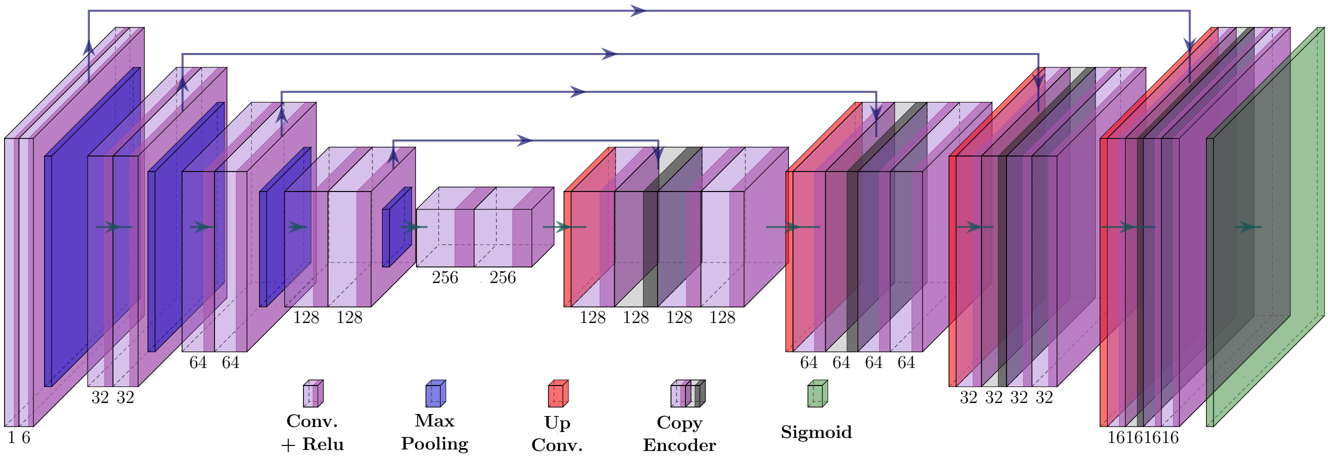

Figure 5: The U-Net CNN architecture proposed by Ronneberger *et al.* [43]. The depth of the output for each layer is shown at the bottom; However, magnitude of these values reflect the current study. The blue arrows indicate where the feature maps from the encoder are added to the decoder.

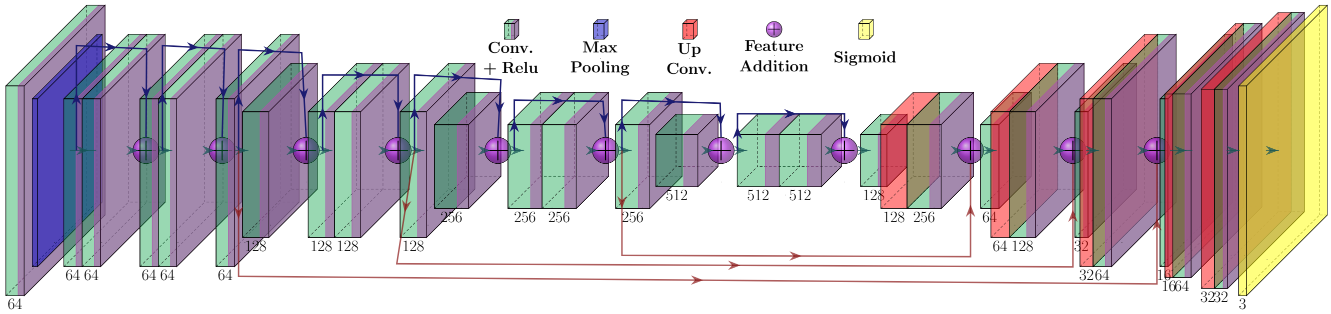

Figure 6: The LinkNet CNN architecture created by Chaurasia *et al.*[58]. The depth of the output from each layer are shown at the bottom, and these values are the same as the original architecture. The blue arrows indicate the residual connections in the encoder, and the red arrows indicate where the feature maps from the encoder and decoder are added.

coefficient metrics are positively correlated, it is of note that the IOU is more sensitive to poorly performing outliers than the dice coefficient and, therefore, their relative values will differ depending on the dataset. One disadvantage both the IOU and the dice coefficient share is that the lower the percentage of true positive pixels compared to background pixels, the higher the penalty for having any type I or type II errors; though as discussed previously it will likely be exacerbated for the IOU. Thus, it is essential to consider the imbalances in the dataset during the evaluation[67].

The receiver operating characteristic (ROC) curve measures the model's ability to correctly predict the features belonging to a particular class. The use of the sigmoid activation function causes the output of the CNN to be pixel-wise probabilities of that pixel belonging to the carbide class. Thus, the ROC curve allows for the selection of the optimal probability threshold to maximize the predictive ability of the model. This is done by plotting the true positive rate (TPR a.k.a. sensitivity) against the false positive rate (FPR), which are shown in equations 8, and 9, respectively.

$$TPR(Sensitivity) = \frac{TP}{TP+FN} \tag{8}$$

$$FPR = 1 - Specificity = \frac{FP}{TN+FP} \tag{9}$$

The closer the area under the ROC curve (AUC-ROC) is to unity, the higher the probability that the model will correctly classify each pixel to the correct class. Since different probability thresholds will give rise to varying TPRs and FPRs, it is important to determine the area under the ROC curve for a range of values. This can be seen in Figure 7, where the ROC curve for fold 1 of the U-net is plotted for various threshold values. For this particular model, the optimal threshold value was 0.77, where values higher and lower had reduced area under the ROC curve. During the training of the models in this work, 50 threshold values between 0 and 1 were tested to find the best possible threshold value. For the hyperparameter search, predictions were based on the validation set, and for k-fold cross-validation, the predictions were from the test set.

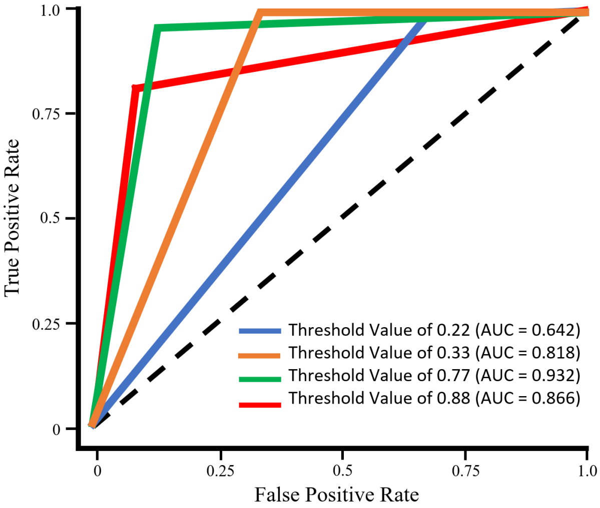

Figure 7: The ROC curve for fold 1 of the U-net used in this work at varying thresholds. The threshold value for each line is given in the legend, along with the corresponding area under the ROC curve. The dashed line represents the lowest possible denomination of the ROC curve.

2.3.3. *Hyperparameter Optimization*

Machine learning models attempt to define the features in a given dataset using a non-linear function. This is achieved by using affine transformations that are dictated using learned parameters and non-linear activation functions, such as the ReLU. The model parameters are internal coefficients that are updated using a gradient-based optimization algorithm to produce the smallest possible error on the dataset. Hyperparameters are configurational arguments that are specified by the user to help facilitate the training process and tailor a model's performance to a particular dataset. Typically, the general effect each hyperparameter has on the models functionality is known. However, the various hyperparameters may have non-linear interactions, resulting in drastic differences in performance.

The hyperparameters tested and their range of values for the U-net are shown in Table 1, and the alternative backbones for the U-net and LinkNet are shown in Table 2. As mentioned above, batch normalization refers to rescaling all of the activations from the hidden layers to be between 0 and 1 [59]. Batch size is the number of training images that are used to estimate the expected value of the gradient for a particular loss function over the entire dataset[76]. Smaller batch sizes have been shown to improve the model's ability to generalize features [77, 78, 79], while reducing the memory footprint during training. The learning rate controls the magnitude the weight parameters of the model are updated per epoch [80]. Decaying the value of the learning rate during training has been shown to improve the model's ability to learn complex patterns and reduces the chance of the model being stuck in a local minima[81]. The decay patience is the number of epochs, with no improvement to the validation loss, it takes for the learning rate to decrease by the specified decay rate. The value for the learning rate decay refers to the amount in which the learning rate drops after there has been no improvement in validation loss after the epochs specified by the patience. Dropout is a regularization technique that temporarily removes neurons and their connections from the network based on the probability defined by the dropout rate to help reduce overfitting[82]. L2 regularization adds a $\lambda \sum_{j=1}^{p} \beta_j^2$ to the objective function to reduce the value of the weights to be closer to zero. Kernel size is the size of the convolution window that is used during the convolutional layers, while the number of filters increases the depth of the convolutional layers. Kernel weights were initialized from a truncated normal distribution, with a mean of 0 and a standard deviation of $\sqrt{2/n}$, where n is the number of input units in the weight tensor [83]. ReLU [57] and ELU [69], shown in equations 10 and 11, are non-linear activation functions that follow each convolutional layer, to account for the complex nature of the pixel data.

Table 1: The hyperparameters and their corresponding value ranges used for the hyperparameter optimization of the U-net. Note that for SGD, momentum was set to the default of 0 and for Adam β_1 and β_2 were set to the recommended values of 0.9 and 0.999 respectively [68]. Larger incremental ranges are shown in brackets (start,finish,increment)

Hyperparameter	Value Range
Batch Normalization	True, False
Batch Size	2, 4, 6, 8
Learning Rate	0.00001, 0.0001, 0.001, 0.01, 0.1
Learning Rate Decay	0, 0.25, 0.5, 0.75, 0.99
Dropout Rate	(0, 1, 0.1)
L2 Regularization term	0, 0.0001, 0.001, 0.01, 0.1
Kernel Size	2, 3, 4, 5
Number of Filters	16, 32
Activation	ReLU [57], ELU [69]
Loss Function	binary cross entropy (BCE)[70], dice coefficient loss (DL)[71], BCE plus dice coefficient loss (BCE+DL)[72]
Optimizer	stochastic gradient decent (SGD)[73][74], RMSprop[75], Adam[68]

$$R(z) = \begin{cases} z & z > 0 \\ 0 & z <= 0 \end{cases} \tag{10}$$

$$R(z) = \begin{cases} z & z > 0 \\ \alpha(e^z - 1) & z <= 0 \end{cases} \tag{11}$$

In deep learning, the objective function that is used to quantify the model's ability to perform a particular task is called the loss function. The cumulative error associated with the entirety of the model is distilled down to a single value, where a reduction in this value indicates improvement in the model's performance. The loss functions tested in this work are pixel-wise binary cross entropy[70], the dice coefficient loss[71], and

Table 2: The hyperparameters and their corresponding value ranges used for the hyperparameter optimization of the U-net and LinkNet for the different backbones. These ranges are the same for all of the different backbones trialed. Note that for SGD, momentum was set to the default of 0 and for Adam β_1 and β_2 were set to the recommended values 0.9 and 0.999 respectively. [68]

Hyperparameter	Value Range
Batch Normalization	True, False
Batch Size	2, 4, 6, 8
Learning Rate	0.00001, 0.0001, 0.001, 0.01, 0.1
Learning Rate Decay	0, 0.25, 0.5, 0.75, 0.99
Decay Patience	5, 10, 20, 30
Decoder Block Type	nearest-neighbor (nn) upsampling, transpose convolution
Encoder Weights	he normal, ImageNet
Freeze Encoder	True, False
Loss Function	binary cross entropy (BCE)[70], dice coefficient loss[71], BCE plus dice coefficient loss[72]
Optimizer	stochastic gradient decent (SGD)[73][74], RMSprop[75], Adam[68]

a combination of the two[72], shown in equations 12, 13, and 14 respectively.

$$L_{BCE}(y,\hat{p}) = -(y log(\hat{p}) + (1-y) log(1-\hat{p})) \tag{12}$$

$$L_{DL}(y,\hat{p}) = 1 - \frac{2y\hat{p}+1}{y+\hat{p}+1} \tag{13}$$

$$L_{BCE,DL} = \beta L_{BCE} - (1-\beta)L_{DL} \tag{14}$$

Parameters of the model are updated after each batch through gradient descent using an optimization algorithm. The optimizers trialed were stochastic gradient decent (SGD)[73], RMSprop[75], and Adam [68].

Different hyperparameters were able to be analyzed for the different backbones based on the input arguments specified by Yakubovskiy's Github repo [64], which include changing the encoder weights, freezing the encoder weights, and altering the decoder block type. Transfer learning [84] using an encoder that was pre-trained on the ImageNet dataset[85] was used to take advantage of the model's ability to generalize and extract features. Otherwise, the weights were initialized from a truncated normal distribution [83]. These weights from ImageNet could be frozen and prevented from being adjusted during training, resulting in only the decoder parameters being updated. The decision to include freezing the weights was made so the large gradient from the new dataset doesn't override the ImageNet weights, eliminating the feature extraction benefits of the pre-trained encoder, potentially causing the entire model to be trained from scratch. The decoder block type was also altered between nearest-neighbor upsampling, which does not use learned parameters, or a transpose convolution.

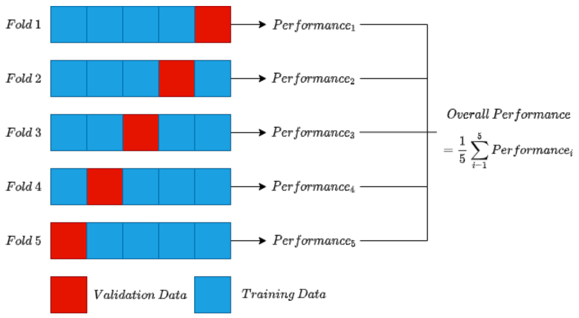

Figure 8: A visual representation of the k-fold cross validation procedure.

Hyperparameter optimization was done using the Talos python package [86], using a random search method. The training was stopped after 50 epochs with no improvement to the validation loss, with a max of 500 epochs, and only the top-performing model was evaluated. The dataset was manually split into training (80%) and validation (20%) to conserve class balance between the datasets and to ensure comparable results across the experiments. The metrics used to evaluate the raw output were IOU (equation 6) and the dice coefficient (equation 7), averaged across the entire validation set.

2.3.4. *K-fold Cross Validation*

Once the optimal hyperparameters were determined for each model, the evaluation was done using k-fold cross-validation. This is a widely used technique for model evaluation since it improves the models ability to generalize by using all of the available data for training and validation [87]. A value of k=5 was selected to reduce the computational load, while being large enough to reduce the bias and variance of the model's performance[88]. The total 229 training images were split into a training set (192 images) and a test set (37 images), where the test set was selected to closely match the class distribution of the training set. During training, the training set is split into five folds, one fold being used as the validation set and four folds being used for training, as shown in Figure 8. The model is trained for 500 epochs, with the goal being to minimize the loss associated with the validation set as a representation of how the model performs on new data. Following training, the test set is passed through the model, and the mean IOU and dice coefficient are determined across the entire test set. This process is repeated five times, and the ultimate performance of the model is the averaged performance across the folds.

2.4. *Post-processing*

The overall model performance was determined before and after post-processing, using morphological operations, of the CNN predictions. Morphological operations use a structuring element that is rastered over the image to adjust the arrangement of features in an image. For this work, the chosen structuring element was a square, with the origin being at the center.

Two morphological operations were used for post-processing of the binary images, erosion, and dilation. Erosion eliminates all pixels (reduces value from 1 to 0) within a structuring element except for the origin, where that structuring element fits within an object (all pixel values of the structuring element are 1). Dilation increases the dimensions of objects by changing the value of the origin of the structuring element from 0 to 1 if any part of the structuring element contacts a pixel with a value of 1 [89]. A visual depiction of erosion and dilution of an arbitrary object by a 3x3 square structuring element can be seen in Figure 9a and Figure 9b, respectively. Opening is erosion followed by dilation, which removes noisy patches and spurs from the image while maintaining the original shape and size of the objects. Closing is dilation followed by erosion which helps in filling small gaps found in the objects in the image[89]. The sequence of morphological operations that yielded the best results was: erosion, opening, closing, and erosion. All morphological transformations were performed using the python package OpenCV [30]. For each morphological step, the structuring element size was iterated between 1x1 and 100x100, and the size that yielded the greatest improvement to the IOU for that particular image was chosen.

3. Results

3.1. Hyperparameter Optimization

The raw output from the top 5 hyperparameter configurations for the segmentation of WC in Ni-WC optical microscopy images using the vanilla U-Net CNN are shown in Table 3. The top model had an area under the ROC curve (AUC-ROC) of 0.944, a mean IOU of 0.87, and a mean dice coefficient of 0.929. Although the hyperparameter values listed in the table are similar in magnitude, the range in configurations between the models is significant. One hyperparameter that was consistent across the top 5 models was the SGD optimization function. It is important to note that during the hyperparameter optimization, only vanilla SGD (i.e. no momentum) was tested. There is no conclusive evidence to show that one optimization function performs best across all types of data; however, for the samples analyzed in this paper, SGD had the best performance as shown in Table 3 [90]. Even exclusively using SGD as the optimizer, model architectures varied enormously, and resulted in large differences in performance. These differences in performance are expected since each model differs in the way it extracts features from the data [91]. Ultimately, it is the symphony of hyperparameters and model architecture that dictate the performance. SGD's positive performance is possibly due to the fact that it has been shown to generalize features better than adaptive optimization methods (e.g. Adam) [92, 93]. One known disadvantage of SGD is the potential to get stuck in local minima, due to the gradient being small in all directions. Using momentum, which incorporates a velocity vector of past gradients to SGD, can help propel backpropagation through problematic areas and towards the global minima [94]. To determine momentum's effect on the top model's performance, a second hyperparameter search was done with momentum ranging from 0 to 1, in 0.1 increments. Incorporating a momentum of 0.8 increased the mean IOU to 0.882 with a mean dice coefficient of 0.937.

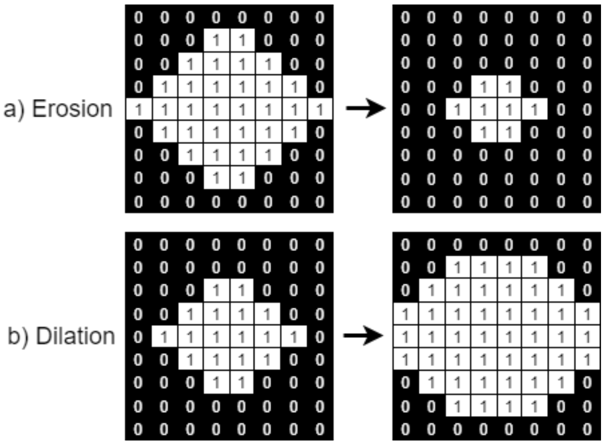

Figure 9: The a) erosion and b) dilation morphological transformations used in the post processing of the CNN output. For both processes the structuring element is a 3x3 square.

Table 3: The hyperparameters and their corresponding value for the top 5 vanilla U-net models trained during the hyperparameter optimization. Note that for SGD that momentum was set to the default of 0.

Hyperparameter	1	2	3	4	5
Batch Normalization	True	True	True	True	True
Batch Size	2	2	2	6	6
Learning Rate	0.01	0.001	0.01	0.001	0.1
Learning Rate Decay	0.5	0.75	0.25	0.25	0.75
Dropout Rate	0.8	0.5	0.5	0.2	0.1
L2 Regularization term	0	0	0.0001	0.01	0.001
Kernel Size	4	3	5	3	2
Number of Filters	16	16	16	32	16
Activation	elu	elu	elu	relu	elu
Loss Function	DL	BCE	BCE + DL	DL	BCE
Optimizer	SGD	SGD	SGD	SGD	SGD
AUC-ROC	**0.944**	0.942	0.942	0.936	0.935
Mean IOU	**0.870**	0.864	0.863	0.849	0.848
Mean DC	**0.929**	0.926	0.925	0.917	0.916

The results for the hyperparameter optimization of the various backbones tested for both the U-Net and LinkNet are shown in Table 4; These values are the raw outputs from the models with no post-processing. No post-processing was done during the hyperparameter search to ensure efficient use of the computational resources provided by ISAIC. The best backbone for the U-Net used the mobilenet encoder achieving an AUC-ROC of 0.939, a mean IOU of 0.855, and a mean dice coefficient of 0.922, which is slightly worse than the original U-Net architecture (IOU of 0.882 and DC of 0.937). Comparatively, the top backbone for LinkNet was the resnet50 encoder, which had slightly better performance than the U-Net with an AUC-ROC of 0.945, a mean IOU of 0.879, and a mean dice coefficient of 0.935. The resnet50 encoder maintains the original residual block structure of the encoder of LinkNet, only slightly deeper [58]. All of the models utilized BCE as part of the loss function. BCE is a common loss function for binary classification as it mimics using maximum likelihood estimation to fit the model. It also pairs well with the sigmoid activation function during backpropagation since the sigmoid function outputs values between 0 and 1, which are the required range for $\hat{p}$ in equation 12[95].

The only instance where transfer learning outperformed training a model from scratch is the U-net using the VGG16 encoder, Here the encoder weights from training the model on the ImageNet dataset [85] were unchanged during training. Thus, the model's learned ability to generalize and extract features was able to be transferred to the Ni-WC optical images used in this work. The success of transfer learning highlights the power of a large dataset (e.g. ImageNet [85]), and the models trained on it, to generalize to a variety of problems. Even though there is a stark difference between the features belonging to natural images compared to microstructures, the model was still able to achieve a mean IOU of 0.851. Due to time and resource constraints, fine-tuning the ImageNet weights was not ~~able to be~~ done; however, this could potentially increase the accuracy of the predictions.

Table 4: The hyperparameters and their corresponding value used for the training of the top performing model for each backbone and model architecture. Note that for SGD that momentum was set to the default of 0 and for Adam β_1 and β_2 were set to 0.9 and 0.999 respectively.

Hyperparameter	Model Architecture				
		Unet		LinkNet	
	mobilenet	renet50	vgg16	mobilenet	resnet50
Batch Normalization	True	True	True	True	False
Batch Size	2	4	4	2	4
Learning Rate	0.1	0.01	0.001	0.01	0.01
Learning Rate Decay	0.75	0.75	0.99	0.25	0.75
Decay Patience	5	5	10	20	30
Decoder Block Type	nn upsampling	transpose	nn upsampling	transpose	transpose
Encoder Weights	he normal	he normal	ImageNet	he normal	he normal
Freeze Encoder	False	False	True	False	False
Loss Function	BCE+DL	BCE	BCE+DL	BCE+DL	BCE
Optimizer	SGD	Adam	RMSprop	SGD	RMSprop
AUC-ROC	0.939	0.927	0.937	0.939	0.945
mean IOU	0.855	0.834	0.851	0.859	0.879
mean DC	0.922	0.903	0.918	0.923	0.935

Table 5: The 5-fold cross validation performance of each model on the Ni-WC test dataset. The final column shows the mean performance across all of the folds, with the raw output values in brackets.

Model	Metric	Value for each for each fold					Mean
		1	2	3	4	5	
U-Net	AUC-ROC	0.968	0.962	0.964	0.967	0.966	**0.965** (0.935)
	Mean IOU	0.918	0.904	0.909	0.914	0.911	**0.911** (0.841)
	Mean DC	0.957	0.949	0.952	0.955	0.953	**0.953** (0.913)
U-Net/mobilenet	AUC-ROC	0.889	0.902	0.947	0.857	0.854	0.890 (0.871)
	Mean IOU	0.769	0.782	0.872	0.711	0.702	0.767 (0.741)
	Mean DC	0.868	0.876	0.829	0.932	0.824	0.866 (0.848)
U-Net/resnet50	AUC-ROC	0.926	0.925	0.941	0.926	0.936	0.931 (0.910)
	Mean IOU	0.831	0.829	0.864	0.833	0.854	0.843 (0.814)
	Mean DC	0.907	0.906	0.927	0.908	0.921	0.914 (0.896)
U-Net/VGG16	AUC-ROC	0.941	0.951	0.900	0.951	0.959	0.940 (0.913)
	Mean IOU	0.870	0.874	0.782	0.878	0.900	0.860 (0.808)
	Mean DC	0.930	0.932	0.876	0.935	0.945	0.924 (0.892)
LinkNet/mobilenet	AUC-ROC	0.955	0.947	0.946	0.952	0.947	0.950 (0.910)
	Mean IOU	0.890	0.871	0.894	0.881	0.869	0.876 (0.803)
	Mean DC	0.941	0.931	0.931	0.936	0.929	0.934 (0.889)
LinkNet/resnet50	AUC-ROC	0.945	0.937	0.937	0.935	0.938	0.939 (0.911)
	Mean IOU	0.867	0.861	0.855	0.850	0.855	0.858 (0.814)
	Mean DC	0.928	0.925	0.921	0.918	0.921	0.922 (0.898)

3.2. *K-fold Cross Validation*

The results from the 5-fold cross-validation experiments on the test dataset are shown in Table 5. The values shown are after post-processing to illustrate the best performance that can be achieved with an automated pipeline. In the final column, the mean of the raw outputs are shown in brackets. The average increase in performance with post-processing is 0.02 - 0.07 across all metrics. When comparing the raw outputs from k-fold cross validation with the hyperparameter search, the generalization error increased for all models. A slight decrease in performance of the U-Net was achieved compared to the results of the top model during the hyperparameter optimization shown in Table 3. Since the test set is unseen by the model it is used to showcase the model's classification ability; the minimal increase in error on the test set compared to the validation set during the hyperparameter search highlights the model's ability to perform on unseen data. Similar results were found when comparing the cross-validated and hyperparameter search results for the different model architectures of the U-Net and LinkNet, with the exception of the U-Net with a mobilenet encoder. This could be due to this model being biased towards the validation set during the hyperparameter search, giving a false sense of the model's ability to perform on unseen data.

The CNN struggled to properly classify the various defects that can be formed during deposition. These problems were consistent across all models used in this work. Thus, to reduce redundancy, only the output of the U-Net will be explicitly discussed. One specific instance is cracking in the sample, which can be found at the bottom of the input image shown in Figure 10a. Although this crack is classified as part of the background, as seen in the ground truth in Figure 10b, the CNN falsely classified it as part of the carbide class, shown in Figure 10c. The type I error still remained after post-processing which can be seen in Figure 10d. Similar results were also found with pores in the carbides, where the model preferred to classify them as part of the carbide. This is expected given that both show similar pixel intensity values under optical microscopy. During the grinding and polishing stage of metal-

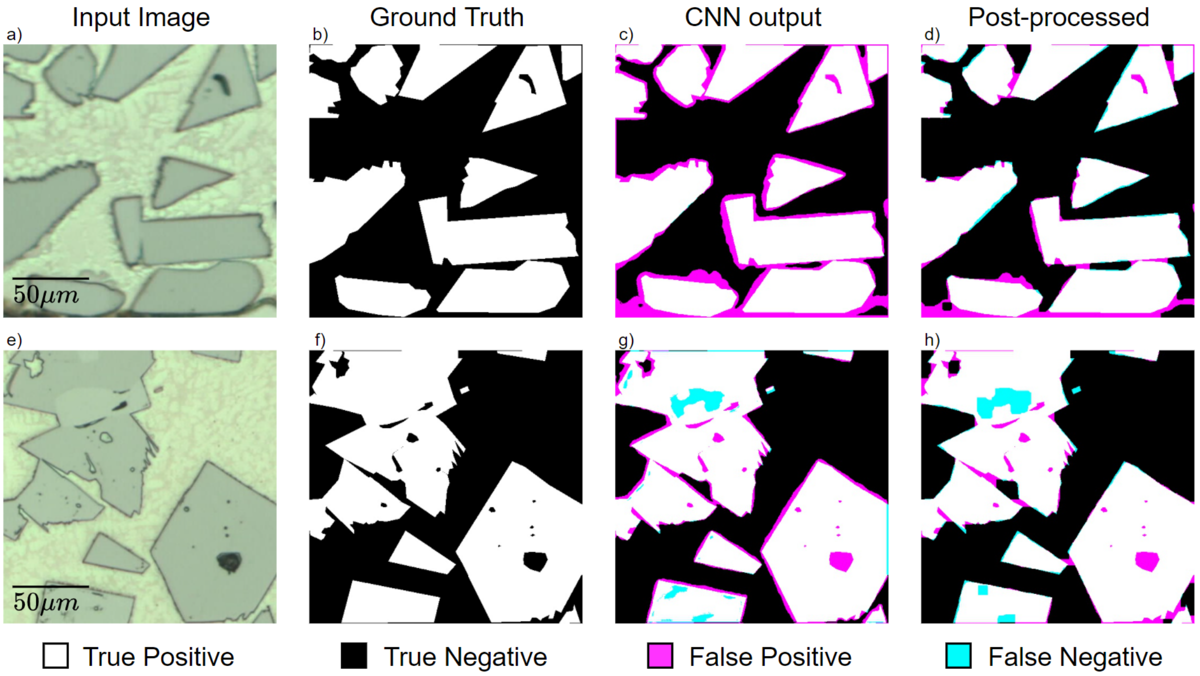

Figure 10: Visual depiction of the segmentation results where a) and e) show the input images into the CNN; b) and f) show the corresponding ground truth for images a) and e) respectively; c) and g) show the raw outputs of the CNN; d) and h) show the post-processed output. For all images the white pixels show the true positives, the black pixels show the true negatives, the magenta pixels show the false positives, and the teal pixels show the false negatives.

lurgical sample perperation, the softer Ni matrix is preferentially worn, resulting in the WC particles being elevated in the sample. This is characterized by the dark outline that is found around each particle, that share a similar pixel distribution as the pores and cracks. During the ground truth labelling there were instances where the dark outline was included in the mask of the carbide. Therefore, the model was being trained to recognize the distribution of pixels that belong to the pores and cracks as belonging to the carbide particle, contributing to the type I errors found in the predictions.

Thermal degradation of the WC particles was another defect that was largely falsely classified by the CNN. Dissolution of WC in NiCrBSi has been shown to form W-Ni-Cr intermetallics along the particle boundary during PTA deposition [2]. A large dissolution zone can be seen at the bottom of Figure 10a that resulted in a false positive error due to the similarity in the pixel distributions between these regions and the WC particles, relative to the matrix. The model also performed poorly at accurately determining the edge of the WC particles, shown by the ring of type I errors surrounding the particles in Figure 10c and Figure 10g. Downsampling is important for CNNs to reduce the dimensionality of the image data, to reduce the chance of overfitting, and to allow the network to learn larger spatial context. The trade-off to this methodology is the model reduces its ability to distinguish high-frequency details, reducing the accuracy of classifying object boundaries[96]. Therefore, the coupling of the dissolution of W and C surrounding the WC particles, and the poor edge definition of the model's output, resulted in poor resolution when classifying WC particles exhibiting thermal degradation. In the development of the U-Net, Ronneberger *et al.* integrated an energy function into the loss function that increased the cost associated with the incorrect classification of border pixels around HeLa cells [43]. The additional cost forced the model to learn the boundaries around cells and could also be used to improve the boundary detection of WC particles. This may reduce the error caused by the poor classification of the thermal degradation of WC.

Similar to the inability to consistently characterize carbide that experience thermal degradation, the model also struggled to separate the carbides that had boundaries close together. Instead, the CNN tends to connect these areas reducing the individual particle count found in each image. This tends to be exacerbated by post-processing, as seen in the bottom right of Figure 10g and h, where type I errors are introduced after post-processing. The post-processing pipeline only accepts changes to the image that will result in an increase in the IOU of the image. Although new type I errors are introduced, the objective outcome does improve. Using simple non-local morphological transformations may not be the optimal choice for the post-processing of CNN output. Other post-processing methods such as conditional random fields [97], and localized adaptive methods [98] may improve the end results.

3.3. Model Uncertainty

For this work, the model's uncertainty was not quantified. Quantification of the carbide characteristics in Ni-WC microstructures in the context of PTA-AM will be used as a tool for maximizing the retention of WC during deposition. In a research context, the amount of data pertaining to a single sample is minimal, and the bottleneck for creating samples lies in the time required for sample preparation prior to optical microscopy. The size of the features is large enough to visually determine whether the model's prediction is correct, and the risk associated with uncertain predictions is low. Thus, the computational expense for quantifying the model's uncertainty outweighed the potential risk.

4. Calculating Carbide Percent and Mean Free Path

The carbide percent and the mean free path were determined for the entire test dataset and compared with the values from the ground truth images as seen in Table 6. The carbide percent was calculated by determining the area of the white pixels over the total area of the image. Mean free path was quantified using two different methods. One of the methods uses equation 1, where N_L is determined by drawing 25 lines horizontally across the image, and counting the number of carbide intercepts, as described in ASTM STP 839 [22]. The location of these lines is randomly generated to try and mimic the manual calculation described in the ASTM standard. This process is repeated 15 times to have a valid statistical representation of the range of possible values for the mean free path. A problem with automating this process is that a carbide/matrix intercept of a binary image is when the pixels change from back to white or vise versa. The transition in pixel value is not always indicative of a carbide/matrix intercept. For example, the model tends to incorrectly classify pores and cracks as belonging to the carbide class, which would underestimate the mean free path. The second method used for calculating the mean free path was by simply scanning the image horizontally and counting the number of black pixels in between white pixels. This method operates under the assumption that all black pixels belong to the matrix, which is not always correct. Pores, cracks, and false negatives incurred during the model's prediction are all cases that reject this assumption.

Table 6: The carbide percent and mean free path calculated over the entire testing dataset. The values stated are the mean ± the standard deviation for the test set. Mean free path (ASTM) refers to the methods described in ASTM STP 839 [22], and mean free path (CBP) refers to the method of counting the number of black pixels between areas of white pixels.

Measurement	Ground Truth	Model Prediction
Carbide Percent (Area %)	40 ± 11	41 ± 11
Mean Free Path (ASTM) (µm)	64.4 ± 30	79.9 ± 40
Mean Free Path (CBP) (µm)	89.7 ± 29.4	115 ± 35

A minor discrepancy can be seen between the values of the carbide percent for the ground truth images and the model's predictions, even though the model's predictions contain type I errors. During the post-processing step, the elimination of a large portion of type I errors also tend to incorporate more type II errors due to the global application of the morphological operations. The addition of type II errors offsets some of the type I errors in images that the model predicted poorer on, resulting in the carbide percentage being closer to that of the ground truth. Both the ground truth and the predictions had similar standard deviations in carbide percentages. These results empirically show that the carbide percent determined from the model's predictions are reflective of what is contained in the optical image.

The difference of mean free path values for the ground truth using the ASTM calculation and the counting black pixels (CBP) method are within 25 microns of the model's predictions. A difference of 25 microns in the mean free path could drastically effect the MMCs ability to resist abrasive wear depending on the size distribution on the abrasive [4]. One major contributing factor to the large difference is including defects in the background class of the ground truth images, highlighted in Figure 10f. Since these areas are represented by small areas of black in the ground truth images, they will be falsely included in the quantification of the mean free path for the ground truth; this will drastically decrease the average value. The models predictions typically included defects as being part of the carbide class, thus not including them in the mean free path. Also, the intricate details of the carbides are smoothed out in the models predictions, omitting them from the mean free path calculation. Therefore, the lower value for mean free path from the ground truth images may be inaccurate, and the higher values from the models predictions may be a closer estimation of the mean free path.

Another important thing to note from Table 6 is the large standard deviation in the MFP values for both the models predictions and the ground truth. It is evident in Figure 3 that the physical phenomena of particle settling could be a large contributing factor, since there would be large discrepancy in the MFP depending on where the test image originated in the sample. The impact the depth where the image was taken from on the MFP is shown in Table 7, and a visual representation of the data for the CPB method can be seen in Figure 11. All points shown are from test images taken from the same sample, and the distance values reported are from the bottom of the sample. The mean free path tends to be smaller when comparing the bottom and the top of the sample; the middle contains a range of values. When the effects of particle settling are removed, the mean and standard deviation of the mean free path for the predictions and the ground truth, as well as the general trend of the values, are similar. The large overlap in the mean free path highlights the accuracy of the predictions. However, images that were taken at similar depths in the sample still display large deviations in the mean free path. Therefore, the stochastic nature of the particle settling that occurs during deposition creates local deviations in the mean free path, based on the particle shape, size, and orientation in the melt. The local differences in mean free path are also exacerbated by taking 256x256 pixel crops during the image acquisition process. Therefore, analyzing the mean free path from 256x256 crops of random portions of the image may not be representative of the entire image, and stitching the images together prior to determining the mean free path may help reduce the deviation. This could not be done in the current study, due to the sparseness of the crops used for the test set; stitching them together would not be metallurgically representative of the sample and result in a poor comparison.

When comparing the ASTM and the CBP methods for de-

Table 7: The mean free path calculated at different depths in the sample. The values stated are the mean ± the standard deviation for the specific depth. Mean free path (ASTM) refers to the methods described in ASTM STP 839 [22], and mean free path (CBP) refers to the method of counting the number of black pixels between areas of white pixels. Note that depth is measured from the bottom of the sample.

Depth	Ground Truth (ASTM)	Model Prediction (ASTM)	Ground Truth (CBP)	Model Prediction (CBP)
300	46.9 ± 12.2	55.5 ± 12.4	71.4 ± 15.6	90 ± 17.8
1200	79.2 ± 30.6	106.2 ± 36.1	103.9 ± 27.2	147.6 ± 38.9
1300	69.3 ± 10.9	67.5 ± 7.6	81.1 ± 16.6	106.4 ± 6.8
1500	60.5 ± 16.8	76.8 ± 21.2	88.4 ± 21.8	114.7 ± 31
1750	60 ± 13.6	72.4 ± 12.3	87.6 ± 17.1	105.5 ± 16.3
2100	52 ± 4.9	65.4 ± 6.9	77.6 ± 4.1	108 ± 6.4
2300	140.9 ± 27	175.1 ± 49.3	158.7 ± 16.3	184.3 ± 22.5

termining the mean free path, the CBP method tends to have higher values for the predictions, and ground truth. The difference in the mean free path between the ground truth and the models predictions is also greater for the CPB method. A key distinction between the two methods is the representation of the mean free path. The ASTM method is an estimation of the average distance between the particles, since it uses multiple horizontal lines to sample the image to estimate the number of carbide particles per length. The CBP method is a direct quantification of the mean free path and accounts for all of the data provided by the image. In the context of digital images, counting the number of carbide intercepts is rudimentary compared to physically measuring the distance between particles over the entire image. Additionally, performing the physical measurements is computationally inexpensive. Therefore, for computationally determining MFP, the methods described in ASTM STP 839 may be outdated and may need to be improved to reflect the current state-of-the-art.

4.1. *Uncertainty of Measuring Carbide Characteristics*

The uncertainty of measuring carbide percentage and the mean free path is an extension of the epistemic and aleatoric uncertainty from the model. This is due to the calculations of the carbide percent and mean free path being direct quantifications of models output. Therefore, without the quantification of the uncertainty of the model, the calculations of carbide percent and mean free path should be taken as qualitative evidence.

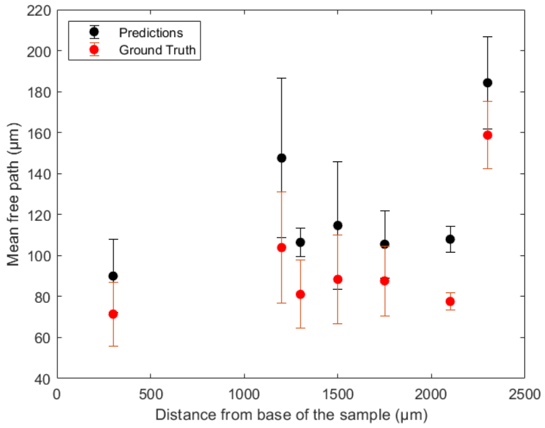

Figure 11: The mean free path as a function of depth in the sample. All points are the mean free path of test images that were taken from the same sample. The depth is measured from the bottom of the sample.

5. Discussion

5.1. *Reduction in Processing Time*

Although the predictions made by the model are not as accurate as those achieved by manually labeling the image, the average mean free path and carbide percent are within 25 µm and 1%, respectively. The largest difference between the two methods is in the processing time. Manually segmenting images from a sample that has a height of 30mm and a length of 100mm would take on the order of hours, whereas the model can semantically segment that sample in less than a minute. In the context of PTA-AM, the lower processing time could drastically reduce the time to optimize the deposition strategy to maximize the carbide percent and minimize the mean free path for the context of abrasive wear resistance.

5.2. *CNN Transferability*

As mentioned above, the training set only contained images of monocrystalline WC in a NiCrBSi matrix. Spherical eutectic WC is also used in industrial applications as they have been shown to have improved wear resistance under high-stress conditions compared to monocrystalline WC, warranted to the combination of high hardness and toughness [2]. Creating a ground truth for spherical WC has proven to be a difficult task, as many of the carbides undergo considerable dissolution during deposition. Degraded carbides have a fractured appearance, as seen in Figure 12a, making them challenging to accurately hand label. However, since spherical eutectic WC are still prevalent in the industry, it would be beneficial for the model to have the capability to segment them from optical images.

To test whether a U-Net only trained on angular monocrystalline WC has the robustness to also predict on spherical WC, the image shown in Figure 12a was cropped into 256x256 sections, totalling 48 images. The images were passed through the same post-processing pipeline prior to being passed through the model. Only the top fold from the original U-Net architecture was tested, and the classification threshold that performed best for the test set was also used for these images. No ground truths were made for these images due to the morphological complexity of the spherical carbides induced by the thermal degradation. No post-processing was done due to the inability to optimize the IOU against the ground truth. Manual application of morphological operations could have been used to improve the results, but the goal of this section was to show what is capable in an automated process that can be applied to any image. Additionally, no evaluation of the carbide percent or mean free path was done since there was no ground truth images to compare the results to.

The raw predictions made by the model can be seen in Figure 12b. There are some sources of errors and uncertainty that became apparent during this experiment. One of those being bevelling of the corners of the sample, illustrated by the shadowing seen in the top left of Figure 12b, resulting in type I

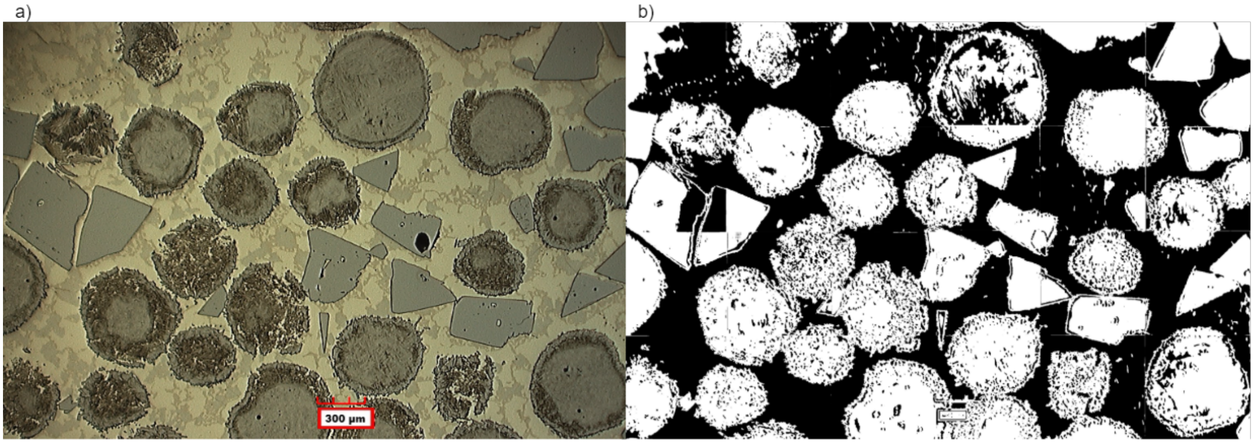

Figure 12: a) Image of a PTA-AM sample containing monocrystalline angular WC and spherical eutectic WC. The scale bar size is increased for reader clarity. b) The raw output from the U-Net on image a).

errors. Another source of error is artifacts left behind from polishing that resulted in additional type I errors. The model's predictions also tended to vary based on the other features contained within the cropped images that aren't representative of the sample as a whole; this was typically a source of type II errors in the prediction. An area where the U-Net seemed to improve on over images with just angular WC is the ability to correctly label the thermal degradation of the WC. The model was also better at separating particles that were close together. Even though it is apparent that many errors are present in the predictions, a visual comparison between the0 input image and the model predictions, the U-Net is capable of segmenting the spherical eutectic WC. It is speculated that significant improvement would be made if some data containing spherical eutectic carbides were added to the training dataset, and the model optimization process was repeated with the new dataset.

5.3. Machine Learning in Materials Science

The results of this study show that CNNs are capable of extracting the carbide percent and mean free path from optical images. Extrapolating these results to other materials systems, the equivalents of these measurements would be phase fraction and secondary dendrite arm spacing; both of which involve significant manual intervention to determine. Based on the ability of a U-Net trained only on angular WC to be able to also segment spherical WC, it can be speculated that if a dataset of optical microstructural images with the same intraclass feature variance as the ImageNet [85] dataset was developed, the model's ability to generalize features contained in microstructures would be able to be transferred to a multitude of metallic systems. The model's ability to extract information from images could be coupled with compositional data from energy-dispersive x-ray spectroscopy (EDX) and phase data from x-ray diffraction to create an automated phase-detection pipeline. This could reduce the need for using advanced characterization methods to extract microstructural information from material systems and allow for the same information to be gathered by using simpler, more cost-effective methods. With the rapid advancement of new manufacturing technologies like additive manufacturing, reducing the need for advanced imaging techniques to properly characterize microstructures would decrease the time to optimize the deposition strategy for new materials. Compiling a dataset of this caliber would require significant collaboration across the materials science research community. However, such an effort would allow materials scientists to be able to fully exploit the capabilities of machine learning models, and revolutionize the way that materials research is done.

6. Conclusion

The effectiveness of semantically segmenting optical images of NiCrBSi-WC metal matrix composites using an automated convolutional neural network (CNN) pipeline has been demonstrated. The methodology described in this work is capable of more accurate segmentation of WC particles when compared with conventional image processing methods discussed in section 1. From the output of the U-Net CNN, the mean free path and carbide percent can be quantified as an effective empirical estimation of what is present in the sample. There is a significant reduction in the required user time to perform quantitative metallography on Ni-WC optical images. This may allow for better optimization of deposition strategies for maximum carbide retention and correlations between the carbide distribution and wear resistance that may not have been made previously. The robustness of the U-Net was also tested by applying the best-performing model to images that contained spherical eutectic WC that were not present in any of the training data. The model was able to segment the spherical WC with considerable accuracy. These results show that the applications of computer vision for materials science are not limited to complex imaging modalities and can be applied to more accessible methods like optical microscopy. Continued development of computer vision models will likely lead to CNNs replacing humans for the task of microstructure quantification.

7. Acknowledgements

This work would not be possible without the financial aid from the Syncrude-NSERC CRD (CRDPJ 514752-17) and HI-AM (NSERC HI-AM NETGP 494158). The authors would like to thank Stefano Chiovelli for his support and patience through the development of this work. Additionally, the authors would like to thank Industry Sandbox and Artificial Intelligence Computing (ISAIC) for supplying the computing resources required for CNN training and testing.

8. Data Availability

The raw data required to reproduce these findings are available to download from DOI: 10.17632/2wmbc95xy9.1. The processed data required to reproduce these findings are available to download from DOI: 10.17632/2wmbc95xy9.1.

References

[1] G. Fisher, T. Wolfe, and K. Meszaros, “The Effects of Carbide Characteristics on the Performance of Tungsten Carbide-Based Composite Overlays, Deposited by Plasma-Transferred Arc Welding,” *Journal of Thermal Spray Technology*, vol. 22, no. 5, pp. 764–771, 2013. [Online]. Available: https://doi.org/10.1007/s11666-012-9877-4

[2] R. Sundaramoorthy, S. X. Tong, D. Parekh, and C. Subramanian, “Effect of matrix chemistry and WC types on the performance of Ni-WC based MMC overlays deposited by plasma transferred arc (PTA) welding,” *Wear*, vol. 376-377, pp. 1720–1727, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164817300959

[3] N. R. C. o. C. A. C. on Tribology, “A strategy for tribology in Canada: enhancing reliability and efficiency through the reduction of wear and friction,” Tech. Rep., 1986.

[4] T. Liyanage, G. Fisher, and A. P. Gerlich, “Microstructures and abrasive wear performance of PTAW deposited Ni–WC overlays using different Ni-alloy chemistries,” *Wear*, vol. 274-275, pp. 345–354, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164811006090

[5] H. Wang, W. Xia, and Y. Jin, “A study on abrasive resistance of Ni-based coatings with a WC hard phase,” *Wear*, vol. 195, no. 1, pp. 47–52, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0043164895067620

[6] P. F. Mendez, N. Barnes, K. Bell, S. D. Borle, S. S. Gajapathi, S. D. Guest, H. Izadi, A. K. Gol, and G. Wood, “Welding processes for wear resistant overlays,” *Journal of Manufacturing Processes*, vol. 16, no. 1, pp. 4–25, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1526612513000807

[7] N. M. Melendez, V. V. Narulkar, G. A. Fisher, and A. G. McDonald, “Effect of reinforcing particles on the wear rate of low-pressure cold-sprayed WC-based MMC coatings,” *Wear*, vol. 306, no. 1, pp. 185–195, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164813004596

[8] S. A. Alidokht and R. R. Chromik, “Sliding wear behavior of cold-sprayed NI-WC composite coatings: Influence OF WC content,” *Wear*, p. 203792, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164821001812

[9] E. Badisch and M. Kirchgaßner, “Influence of welding parameters on microstructure and wear behaviour of a typical NiCrBSi hardfacing alloy reinforced with tungsten carbide,” *Surface and Coatings Technology*, vol. 202, no. 24, pp. 6016–6022, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S025789720800621X

[10] J. Mercado Rojas, T. Wolfe, B. Fleck, and A. J. Qureshi, “Plasma transferred arc additive manufacturing of Nickel metal matrix composites,” *Manufacturing Letters*, vol. 18, pp. 31–34, oct 2018.

[11] J. G. M. Rojas, M. Ghasri-Khouzani, T. Wolfe, B. Fleck, H. Henein, and A. J. Qureshi, “Preliminary geometrical and microstructural characterization of WC-reinforced NiCrBSi matrix composites fabricated by plasma transferred arc additive manufacturing through Taguchi-based experimentation,” *The International Journal of Advanced Manufacturing Technology*, vol. 113, no. 5, pp. 1451–1468, 2021. [Online]. Available: https://doi.org/10.1007/s00170-020-06388-2

[12] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[13] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[14] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[15] T. W. Clyne and P. J. Withers, *An Introduction to Metal Matrix Composites*. Cambridge: Cambridge University Press, 1993. [Online]. Available: https://www.cambridge.org/core/books/an-introduction-to-metal-matrix-composites/1F8F6103DBF4CE7E48EAE95EAA56CBA4

[16] Š. Houdková, F. Zahálka, M. Kašparová, and L. M. Berger, “Comparative Study of Thermally Sprayed Coatings Under Different Types of Wear Conditions for Hard Chromium Replacement,” *Tribology Letters*, vol. 43, no. 2, pp. 139–154, 2011. [Online]. Available: https://doi.org/10.1007/s11249-011-9791-9

[17] Y. Pan, D. Y. Li, and H. Zhang, “Enhancing the wear resistance of sintered WC–Co composite by adding pseudo-elastic TiNi constituent,” *Wear*, vol. 271, no. 9, pp. 1916–1921, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164811001451

[18] M. Anderson, Chiovelli, S, and R. Llewellyn, “The Use of Tungsten Carbide Materials for Oilsand Wear Applications,” *Thermal Spray 2003 : Advancing the Science and Applying the Technology*, vol. 1, pp. 509–518, 2003.

[19] T. Wolfe, “Homogeneity of Metal Matrix Composites Deposited by Plamsa Transferred Arc Welding,” Ph.D. dissertation, University of Alberta, Edmonton, 2010.

[20] H. Saito, A. Iwabuchi, and T. Shimizu, “Effects of Co content and WC grain size on wear of WC cemented carbide,” *Wear*, vol. 261, no. 2, pp. 126–132, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164805004588

[21] J. Y. Sheikh-Ahmad and J. A. Bailey, “The wear characteristics of some cemented tungsten carbides in machining particleboard,” *Wear*, vol. 225-229, pp. 256–266, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164898003615

[22] J. H. Steele and J. L. McCall, Eds., *Application of Quantitative Microscopy to Cemented Carbides*. West Conshohocken, PA: ASTM International, jan 1984. [Online]. Available: https://www.astm.org/DIGITAL{_}LIBRARY/STP/SOURCE{_}PAGES/STP839.htm

[23] A. H. Jones and P. Roffey, “The improvement of hard facing coatings for ground engaging applications by the addition of tungsten carbide,” *Wear*, vol. 267, no. 5, pp. 925–933, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164809002592

[24] K. Van Acker, D. Vanhoyweghen, R. Persoons, and J. Vangrunderbeek, “Influence of tungsten carbide particle size and distribution on the wear resistance of laser clad WC/Ni coatings,” *Wear*, vol. 258, no. 1, pp. 194–202, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164804002662

[25] P. Evennett and C. Hammond, “Microscopy | overview,” in *Encyclopedia of Analytical Science (Second Edition)*, second edition ed., P. Worsfold, A. Townshend, and C. Poole, Eds. Oxford: Elsevier, 2005, pp. 32–41. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0123693977003769

[26] A. Iskakov and S. R. Kalidindi, “A Framework for the Systematic Design of Segmentation Workflows,” *Integrating Materials and Manufacturing Innovation*, vol. 9, no. 1, pp. 70–88, 2020. [Online]. Available: https://doi.org/10.1007/s40192-019-00166-z

[27] H. Peregrina-Barreto, I. R. Terol-Villalobos, J. J. Rangel-Magdaleno, A. M. Herrera-Navarro, L. A. Morales-Hernández, and F. Manríquez-Guerrero, “Automatic grain size determination in microstructures using image processing,” *Measurement*, vol. 46, no. 1, pp. 249–258, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/

S0263224112002539

[28] A. Campbell, P. Murray, E. Yakushina, S. Marshall, and W. Ion, “New methods for automatic quantification of microstructural features using digital image processing,” *Materials & Design*, vol. 141, pp. 395–406, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127517311620

[29] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, “scikit-image: image processing in Python,” *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: https://doi.org/10.7717/peerj.453

[30] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[31] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” vol. 8, 01 2007, pp. 1027–1035.

[32] A. Jalalian, S. B. T. Mashohor, H. R. Mahmud, M. I. B. Saripan, A. R. B. Ramli, and B. Karasfi, “Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review,” *Clinical Imaging*, vol. 37, no. 3, pp. 420–426, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0899707112002938

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

[34] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.

[35] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Feedforward Networks,” in *Deep Learning*. MIT Press, 2016, ch. 6, pp. 165–166. [Online]. Available: http://www.deeplearningbook.org

[36] Zhou and Chellappa, “Computation of optical flow using a neural network,” in *IEEE 1988 International Conference on Neural Networks*, 1988, pp. 71–78 vol.2.

[37] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608014002135

[38] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, vol. 187, pp. 27–48, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215017634

[39] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” 2016.

[40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” 2017.

[41] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model-based human pose and shape estimation,” 2018.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2014.

[43] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.

[44] C. Shashank Kaira, X. Yang, V. De Andrade, F. De Carlo, W. Scullin, D. Gursoy, and N. Chawla, “Automated correlative segmentation of large Transmission X-ray Microscopy (TXM) tomograms using deep learning,” *Materials Characterization*, vol. 142, no. May, pp. 203–210, 2018. [Online]. Available: https://doi.org/10.1016/j.matchar.2018.05.053

[45] D. S. Bulgarevich, S. Tsukamoto, T. Kasuya, M. Demura, and M. Watanabe, “Pattern recognition with machine learning on optical microscopy images of typical metallurgical microstructures,” *Scientific Reports*, vol. 8, no. 1, pp. 3–9, 2018. [Online]. Available: http://dx.doi.org/10.1038/s41598-018-20438-6

[46] O. Abouelatta, “Classification of Copper Alloys Microstructure Using Image Processing and Neural Network,” *Journal of American Science*, vol. 9, pp. 213–223, jun 2013.

[47] A. Chowdhury, E. Kautz, B. Yener, and D. Lewis, “Image driven machine learning methods for microstructure recognition,” *Computational Materials Science*, vol. 123, pp. 176–187, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.commatsci.2016.05.034

[48] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, and F. Mücklich, “Advanced steel microstructural classification by deep learning methods,” *Scientific Reports*, vol. 8, no. 1, pp. 1–14, 2018.

[49] T. Stan, Z. T. Thompson, and P. W. Voorhees, “Optimizing convolutional neural networks to perform semantic segmentation on large materials imaging datasets: X-ray tomography and serial sectioning,” *Materials Characterization*, vol. 160, no. January, p. 110119, 2020. [Online]. Available: https://doi.org/10.1016/j.matchar.2020.110119

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” 2019.

[51] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[52] K. Wada, “labelme: Image Polygonal Annotation with Python,” https://github.com/wkentaro/labelme, 2016.

[53] I. Goodfellow, Y. Bengio, and A. Courville, “Applications,” in *Deep Learning*. MIT Press, 2016, pp. 448–449. [Online]. Available: https://www.deeplearningbook.org/contents/applications.html

[54] A. Buades, B. Coll, and J.-M. Morel, “Non-Local Means Denoising,” *Image Processing On Line*, vol. 1, pp. 208–212, 2011, https://doi.org/10.5201/ipol.2011.bcm_nlm.

[55] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[56] R. Yasrab, N. Gu, and X. Zhang, “An encoder-decoder based convolution neural network (cnn) for future advanced driver assistance system (adas),” *Applied Sciences*, vol. 7, no. 4, 2017. [Online]. Available: https://www.mdpi.com/2076-3417/7/4/312

[57] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML’10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[58] A. Chaurasia and E. Culurciello, “Linknet: Exploiting encoder representations for efficient semantic segmentation,” *CoRR*, vol. abs/1707.03718, 2017. [Online]. Available: http://arxiv.org/abs/1707.03718

[59] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[60] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[61] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[62] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” 2018.

[63] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[64] P. Yakubovskiy, “Segmentation models,” https://github.com/qubvel/segmentation_models, 2019.

[65] P. Jaccard, “The distribution of the flora in the alpine zone.1,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912. [Online]. Available: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x

[66] L. R. Dice, “Measures of the Amount of Ecologic Association Between Species,” *Ecology*, vol. 26, no. 3, pp. 297–302, apr 1945. [Online]. Available: http://www.jstor.org/stable/1932409

[67] W. R. Crum, O. Camara, and D. L. G. Hill, “Generalized overlap measures for evaluation and validation in medical image analysis,” *IEEE Transactions on Medical Imaging*, vol. 25, no. 11, pp. 1451–1461, 2006.

[68] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.

[69] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” 2016.

[70] Ma Yi-de, Liu Qing, and Qian Zhi-bai, “Automated image segmentation using improved pcnn model based on cross-entropy,” in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, Oct 2004, pp. 743–746.

[71] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” *Lecture Notes in Computer Science*, p. 240–248, 2017. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-67558-9_28

[72] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, “Combo loss: Handling input and output imbalance in multi-organ segmentation,” 2018.

[73] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400 – 407, 1951. [Online]. Available: https://doi.org/10.1214/aoms/1177729586

[74] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” 2018.

[75] T. Tieleman and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude,” COURSERA: Neural Networks for Machine Learning, 2012.

[76] D. Masters and C. Luschi, “Revisiting small batch training for deep neural networks,” 2018.

[77] D. Wilson and T. R. Martinez, “The general inefficiency of batch training for gradient descent learning,” *Neural Networks*, vol. 16, no. 10, pp. 1429–1451, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608003001382

[78] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient BackProp BT - Neural Networks: Tricks of the Trade: Second Edition,” G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. [Online]. Available: https://doi.org/10.1007/978-3-642-35289-8_3

[79] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, “On large-batch training for deep learning: Generalization gap and sharp minima,” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–16, 2017.

[80] L. N. Smith, “Cyclical learning rates for training neural networks,” 2017.

[81] K. You, M. Long, J. Wang, and M. I. Jordan, “How does learning rate decay help modern neural networks?” 2019.

[82] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[83] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[84] S. Bozinovski, “Reminder of the first paper on transfer learning in neural networks, 1976,” *Informatica*, vol. 44, 09 2020.

[85] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[86] M. Kotila, “Autonomio Talos,” 2019. [Online]. Available: http://github.com/autonomio/talos.

[87] S. Raschka, “Model evaluation, model selection, and algorithm selection in machine learning,” 2020.

[88] G. James, D. Witten, T. Hastie, and T. Robert, “Resampling Methods,” in *An Introduction to Statistical Learning with Applications in R*, 2nd ed. Springer US, 2013, ch. 5, pp. 177–205.

[89] K. N. Ngan, T. Meier, and D. Chai, “Chapter 4 - model-based coding,” in *Advanced Video Coding: Principles and Techniques*, ser. Advances in Image Communication, K. N. Ngan, T. Meier, and D. Chai, Eds. Elsevier, 1999, vol. 7, pp. 183–249. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0928147999800060

[90] I. Goodfellow, Y. Bengio, and A. Courville, “Optimization for Training Deep Models,” in *Deep Learning*. MIT Press, 2016, ch. 8, pp. 271–325. [Online]. Available: http://www.deeplearningbook.org

[91] I. Kandel, M. Castelli, and A. Popovič, “Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images,” *Journal of Imaging*, vol. 6, no. 9, 2020. [Online]. Available: https://www.mdpi.com/2313-433X/6/9/92

[92] N. S. Keskar and R. Socher, “Improving generalization performance by switching from adam to sgd,” 2017.

[93] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, “The marginal value of adaptive gradient methods in machine learning,” 2018.

[94] B. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0041555364901375

[95] I. Goodfellow, Y. Bengio, and A. Courville, “Probability and Information Theory,” in *Deep Learning*. MIT Press, 2016, ch. 3, pp. 65–66. [Online]. Available: http://www.deeplearningbook.org

[96] D. Marmanis, K. Schindler, J. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 158–172, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092427161630572X

[97] A. Dhawan, P. Bodani, and V. Garg, “Post processing of image segmentation using conditional random fields,” in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2019, pp. 729–734.

[98] X. Pan, J. Zhao, and J. Xu, “An end-to-end and localized post-processing method for correcting high-resolution remote sensing classification result images,” *Remote Sensing*, vol. 12, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/5/852