A DEEP LEARNING APPROACH FOR DEFECT DETECTION AND SEGMENTATION IN X-RAY COMPUTED TOMOGRAPHY SLICES OF ADDITIVELY MANUFACTURED COMPONENTS

Pradip Acharya*[1], Tsuchin P. Chu[1], Khaled R. Ahmed[2], Subash Kharel[2]

[1]Department of Mechanical Engineering and Energy Processes
[2]School of Computing
Southern Illinois University, 1230 Lincoln Drive, Carbondale, IL 62901

ABSTRACT

Additive manufacturing is an emerging and crucial technology that can overcome the limitations of traditional manufacturing techniques to accurately manufacture highly complex parts. X-ray Computed Tomography (XCT) is a widely used method for non-destructive testing of AM parts. However, detection and segmentation of defects in XCT images of AM have many challenges due to contrast, size, and appearance of defects. This study developed deep learning techniques for detecting and segmenting defects in XCT images of AM. Due to a large number of required defect annotations, this paper applied image processing techniques to automate the defect labeling process. A single-stage object detection algorithm (YOLOv5) was applied to the problem of defect detection in image data. Three different variants of YOLOv5 were implemented and their performances were compared. U-Net was applied for defect segmentation in XCT slices. Finally, this research demonstrates that deep learning techniques can improve the automatic defect detection and segmentation in XCT data of AM.

KEYWORDS

Additive Manufacturing, X-ray Computed Tomography, Convolutional Neural Network, YOLOV5, U-Net.

1. INTRODUCTION

Additive manufacturing (AM) is an emerging technique that involves the construction of a three-dimensional object from a digital 3D CAD model in a layer-wise fashion. AM possesses several advantages such as the manufacturing of components with complex geometries, rapid prototyping, design flexibility, and material customization [1]. The final property of an additively manufactured part is a result of a combination of several processing parameters such as laser power, scanning hatch distance, and scan speed. However, the effect of these parameters and the correlation of each of these parameters to the mechanical properties and defects has not been fully defined. There is a greater chance of defect formation in AM parts without fully optimized parameters. Additionally, internal defects can occur due to other reasons such as residual stress[2]. These defects can lead to failure of the parts during operation or early stage of performance.

Several non-destructive evaluation methods such as infrared thermography, laser ultrasonics, and X-ray computed tomography (XCT) have been implemented to detect defects in AM parts. X-ray Computed Tomography (XCT) is widely used to inspect the AM parts to evaluate internal defects and internal properties such as porosity. XCT utilizes either a moving X-ray beam and detector or

DOI: 10.5121/ijaia.2022.13401

an object rotating within an X-ray beam and detector to obtain 'slices' (or cross-sections) of physical objects which can be used to reconstruct a 3-dimensional (3D) model of the scanned object.

Developing a machine learning model that can classify and localize the defects in the post-build XCT images possesses many advantages. Knowing the location of defects in post-build data will help correlate how that defect appears in the in-situ monitoring sensor signal. Those location and defect information can be further utilized for data preparation and training a machine learning model for online monitoring of AM. Gobert et. al. [3] applied supervised machine learning for defect detection during metallic powder bed additive manufacturing using high-resolution imaging. They found the ground truth location of defects in post-build CT data using image processing algorithms and transferred the information to in-situ data by coordinate transformation using the least-squares approach for labeling the layer-wise images. From those in-situ sensor data, they applied an ensemble classification scheme and used K-fold cross-validation for the performance evaluation of discontinuity detection. X-ray CT can be used to inspect the quality of AM parts and thus advanced method to automate the tedious process of defect detection and segmentation is essential. However, several challenges such as contrast, size, distribution, and appearance make it difficult to automate the process of defect detection and segmentation in XCT data of AM. As a result of such difficulties traditional image processing techniques are not always effective. Accurate detection of defects requires skilled manpower to examine every slice in the scan.

To automate defect detection and segmentation, this study applied two state-of-art deep learning algorithms, YOLOv5 [4] and U-Net [5] respectively. The dataset used in the study is publicly available XCT data [6] of additively manufactured cobalt chrome samples. To generate the annotation required for the training of those supervised deep learning networks, a series of image processing was carried out. An open-source software ImageJ [7] was used to apply Bernsen's auto local thresholding algorithm. A method for generating an annotation for the training of YOLOv5 and U-Net was developed using Python programming. In this study, YOLOv5 has been implemented for defect detection and U-Net for automatic defect segmentation in XCT slices of AM samples. The focus of the study was to implement deep learning techniques on the XCT data. The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 presents detailed information about dataset preparation, the architecture of the model applied, and details of implementation. Section 4 discusses the results of the experiments. Finally, Section 5 presents a conclusion with future research recommendations.

2. RELATED WORKS

The neural network has been used for defect detection and classification in several applications. Poudel and Chu [8] applied a combination of fuzzy logic and artificial neural network (ANN) techniques to achieve automated discontinuity detection in carbon/carbon (C/C) composite aircraft. Their research applied a hybrid fuzzy neural method to the infrared thermography and ultrasonic testing (UT) data for the C/C samples. Similarly, Pan et al. [9] applied fuzzy logic expert rules to improve the accuracy of anomaly detection and reduce human error in the analysis of infrared thermography inspection data. In their study, fuzzy logic achieved reasonable results for porosity location and distribution for the test result of advanced NDT methods such as infrared thermography.

Kim et al. [10] developed an image segmentation and analysis method for defect detection and the measurement of pore structure in an additively manufactured component. They prepared 6 samples from the process of laser powder bed fusion with varying process parameters, scan speed, and hatching distance, resulting in porosity levels ranging from 0.1% to 70%. A local thresholding algorithm applying Bernsen's method [11] was successfully used for the different 2D images. The quantitative value of porosity measured from the image segmentation process and the experimental process was in good agreement. Tokime et al. [12] showed an effective application of the deep convolutional neural network to detect porosity in the X-ray inspection of welded parts. They used an open-source database called GDXray, which has radiographic images of welding with porosity defects. SegNet,[13] a deep convolutional network based on encoder-decoder architecture, was applied based on their need. Kwon et al. [14] found deep learning-based classification models effective to monitor and classify the in-situ melt pool images in 6 categories based on laser powers. Davtalab et al. [15] applied deep learning for the detection of layer deformation in additive manufacturing. They developed a CNN architecture based on SegNet for the segmentation of defects in input images and have received an F1 score greater than 90%. Similarly, [16] Gonzalez-Val et al. developed a novel CNN-based approach to extract features and indicators of quality from the real-time medium wavelength infrared co-axial images.

Alqahtani et al. [17] applied a convolutional neural network to the micro-computed tomography image data set of three different porous sandstone (Bentheimer, Berea, and Gosford) to evaluate the various physical properties. For the training and testing, both greyscale and binary images were used. The results of the testing image were able to provide the average relative error between ground truth labels and predictions from binary images. The errors were 2.7 % for porosity, 5.8% for a specific area, and 6% for mean pore size. However, with training and testing using a grey scale image the relative error of less than 6.3% for porosity, 5.8% for specific surface area, and 6.7% for mean pore size were achieved. Their study suggested that machine learning can play an important role in digital rock analysis. Mutiargo et al. [18] used deep learning (U-Net architecture) for the detection of pores in X-ray computed tomography images of additively manufactured components. Their result shows that the data augmentation significantly increases detectability using a U-Net. The author concluded that the U-Net can be used to significantly reduce the amount of post-processing time needed and it improves the accuracy of detection compared to traditional image processing. Li et. al. [19] proposed a novel approach to solve the requirement of the huge volume of annotated data which is very expensive to generate. They developed a semi-supervised CNN model which combines labeled and unlabeled data during the training. Similarly, Yuan et al. [16]developed the semi-supervised CNN for online monitoring of the selective laser melting process.

Lee et al. [20] applied CNN for the classification of the defect in online monitoring image data. Similarly, Rand et. al. [21] found that the Faster RCNN [22] object detection model is capable of classifying and locating the re-coater defect in the in-situ monitoring images. Gobert et. al. [3] combined ground truth labels of defects from post-build CT data to the high resolution online monitoring data for defect detection in in-situ monitoring of additive manufacturing using supervised machine learning. To find the ground truth labels of a defect in post-build CT data they applied image processing and those labels were transferred to the in-situ sensor data. Features were extracted from layer-wise images and finally, a ensemble classification scheme was applied for the classification.

Detection and segmentation of defects in XCT data are usually done with image processing techniques. However, XCT data of AM parts usually need a custom method for detection and segmentation of defects based on the type of dataset since XCT data possess unique challenges due to material, size, contrast, etc. This work investigates the opportunities for applying a deep learning approach for automatic segmentation, detection, and localization of defects in XCT slices. To the best of our knowledge, we present the first work that implements the state-of-art object detection algorithm You Look Only Once (YOLO) v5[4] in the XCT data of AM.

3. METHODOLOGY

3.1. Dataset Preparation

The study was carried out with the publicly available XCT image dataset of additively manufactured cobalt chrome specimens. The image dataset was produced by Kim et al. [6] for the National Institute of Standards and Technology (NIST). The dataset comprises high-resolution XCT images of additively manufactured samples where the artificial defects were created by varying laser powder bed fusion processing parameters (scan speed and hatch spacing). The dataset contains a large number of small-sized defects, so image pre-processing was utilized to assist the data annotation. Bernsen's [23] automatic local thresholding algorithm was applied for the binarization of the image. Bernsen's algorithm works only with 8-bit images so 16-bit raw images were converted to 8-bit images. However, 16- bit raw images were used during the training of the machine learning framework. For binarization, 16-bit raw XCT images were converted to 8-bit images followed by 2D median filtering, denoising, and finally, auto local thresholding using Bernsen's algorithm in a similar approach as implemented by Kim et. al. [10]. The auto local thresholding using Bernsen's algorithm was implemented as available in the ImageJ open-source software [7]. The circular window radius (r) for the algorithm was chosen from the literature study [23] and local contrast threshold (LCT) values were found by varying it until the best segmentation result was found. The final image after thresholding as shown in Figure 1 for two different samples.

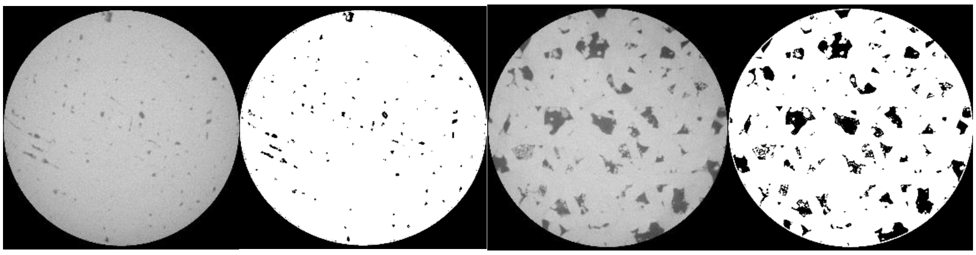
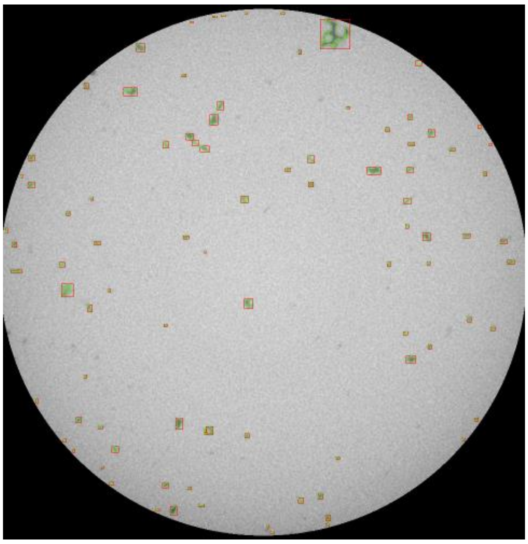


Figure a Figure b

Figure 1. Two of the sample raw images with their corresponding segmentation (Figure a and Figure b)

Rectangular bounding box annotation for each defect is required to train the YOLO algorithm. An approach was developed that automatically generates the rectangular bounding box for each defect in the image. Applying OpenCV [24], an open-source computer vision library available in Python, contours of each defect from the segmented image were found, and those contours were transferred and plotted in their corresponding raw image. OpenCV was further applied to find the smallest bounding rectangle of each defect contour, and the two-diagonal coordinate of the bounding box (X_{min}, Y_{min}, X_{max}, Y_{max}) was written in the .xml file as shown in Figure 2. In addition to the coordinate of the bounding box, filename, path, width, height, depth, and object class name were generated in a .xml file which exactly replicates the process of manual data annotation using the software LabelImg [25]. The annotation with two coordinates of the bounding box was further converted into the form of the first corner coordinate (x, y) of a rectangle, width (w), and height (h).

Training the U-Net requires image data with their corresponding mask where the mask represents the pixels for defects. The pixel which has defect are transformed to the pixel value of 1, whereas the background is 0. Firstly, a blank image with the same image size as the raw image was created, and the co-ordinate of those defects' contours in the binarized image was transferred to the blank images. All the pixels inside the defect contours were replaced with the pixel value of 1. Thus, any pixel which is deemed as a defect has a pixel value of 1, and metal or background has a pixel value of 0.




```xml
<annotation>
	<folder>rawdata</folder>
	<filename>1stRaw_103.tif</filename>
	<path>C:\1stRaw_103.tif</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>980</width>
		<height>1010</height>
		<depth>1</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>defect</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>607</xmin>
			<ymin>956</ymin>
			<xmax>615</xmax>
			<ymax>967</ymax>
```


Figure a　　　　　　　　　　　　Figure b

Figure 2. Image showing (a) rectangular bounding box enclosing defect contours, (b) annotation .xml file

3.2. Defect Detection

Classifying and localizing the defect in XCT of AM possess a challenge due to the contrast, size, and appearance of the defect. In this work, an object is a defect presented in XCT slices, and the problem is treated as defect detection. This work investigates the application of the state-of-art object detection algorithm YOLOv5 for the localization of defects in XCT slices. YOLO [4] is a single-stage object detector that has been mostly used in real-time object detection in various computer vision applications such as animal detection on the road [26]. YOLOv5 is a recent version of YOLO and has been implemented for object detection such as pothole detection on road [27]. However, based on the authors' knowledge, YOLOv5 has not been implemented in computed tomography data of additive manufacturing. The process of object detection involves 3 major steps: resizing an image, running data through a convolution network, and applying non-max suppression. Intersection over Union (IoU) calculates the overlap between two bounding boxes. IoU is mostly used to measure the overlap of prediction with the ground truth bounding box and is expressed as follows [28]:

$$Intersection\ Over\ Union\ (IOU) = \frac{A \cap B}{A \cup B} \qquad (1)$$

Where A represents the prediction and B represents the ground truth.

NMS algorithm is used to find the best bounding box among multiple predicted bounding boxes, and it works in the repetition of the following method. The bounding box with the maximum confidence score assigned to it is individually compared with all other bounding boxes that have an intersection with it. If their Intersection over Union (IoU) is greater than a provided threshold value, they are discarded. The YOLO detection network consists of 24 convolution layers followed by two fully connected layers as shown in Figure 3. Firstly, it divides the input image into an *s*s* grid cell, and *b* number of objects bounding boxes are predicted with confidence level and *c* class probabilities for each of those grids. The prediction results are encoded in the form of a tensor as *s*s* (b*5+c)*. The number 5 in this expression represents the results of bounding box prediction which is co-ordinate of bounding box *x*, *y*, *w*, *h*, and confidence.

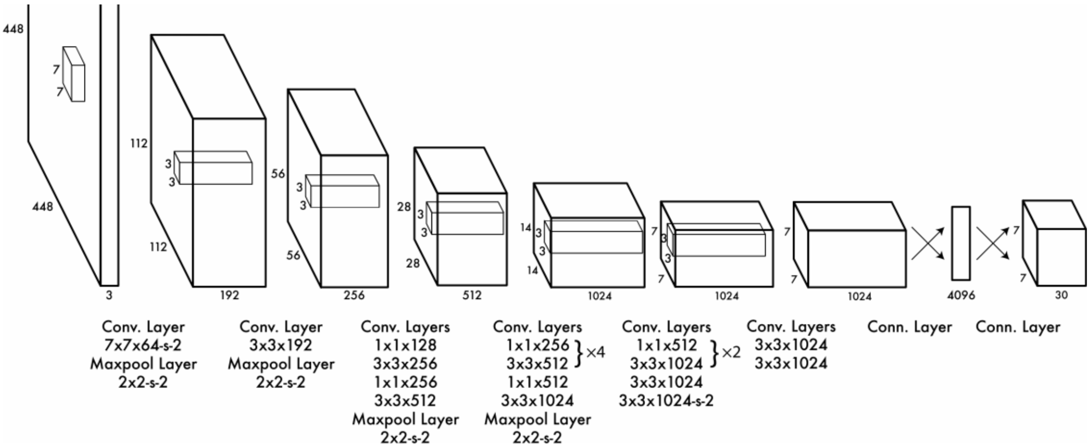


Figure 3. YOLO architecture [4]

3.3. Defect Segmentation

In this study, U-Net architecture was implemented for defect segmentation. U-Net is a CNN initially developed for biomedical image segmentation [5]. The U-Net architecture consists of mainly two-part: one which performs contraction (left side) and another which performs expansion (right side) as shown in Figure 4. Figure 4 shows the input image, network architecture, and segmentation result as an output. The contraction path consists of repeated application of two 3*3 unpadded convolutions each followed by a Rectified Linear Unit (*ReLU*) and a 2×2 max pooling operation for down sampling. The number of feature channels is doubled at every down sampling step. In the expansion path, every step consists of up sampling of the feature map followed by a 2×2 convolution (up convolution) which reduces the number of feature channels by half. Then there is a concatenation with a correspondingly cropped feature map from the contracting path and two 3×3 convolutions each, which is followed by a *ReLU*. The cropping is carried out because of the loss of boundary pixels in each convolution. Finally, a 1×1 convolution layer is applied to obtain the output mask. The network consists of a total of 23 convolutional layers.

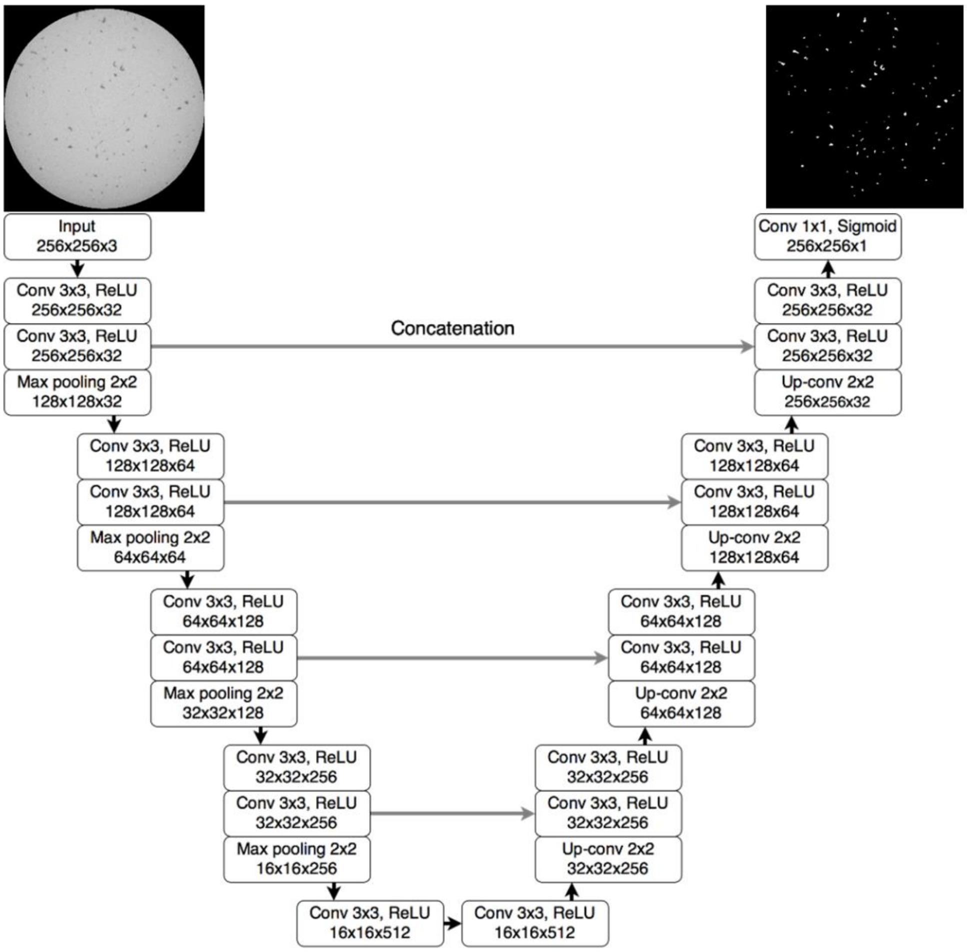


Figure 4. U-Net architecture used for defect segmentation [5]

3.4. Experimental Setup

In this research, a machine having operating system Ubuntu 20.4, embedded with Intel Core i5 CPU, NVIDIA RTX 2060 (8GB) GPU 16GB RAM was used for training. Several software packages such as OpenCV [24], CudatoolKit [29], Matplotlib [30], Numpy [31] and Tensorboard[32] were installed.

3.5. Hyper Parameters

Hyperparameters are utilized to control the learning process. In machine learning, getting good results and drawing a conclusion involves several experiments specific to the problem. For example, a set of parameters used to classify and localize pedestrians in the general image may not be suitable for detecting manufacturing defects in metal surfaces or anomalies in XCT test results. Learning rate is a tuning parameter in an optimization algorithm that provides the step size at each iteration during the process of moving towards the minimum value in the loss function. The YOLOv5 algorithm automatically calculates the learning rate, whereas the best learning rate for U-Net was found by running several experiments to choose the optimal learning rate. The batch size is the number of training samples that train through the network together in an epoch. The batch size was used based on the computational capacity of the machine used for the experiment. Table 1 shows the training parameters used for the YOLOv5 and U-Net. The same parameters were used for all 3 different models of YOLOv5.

Table 1. Training parameters for YOLOv5 and U-Net

Parameters	**YOLOV5**	**U-Net**
Batch size	4	2
Initial learning rate	0.01	
Final learning rate	0.2	0.0001
Optimizer	SGD	Adam
Epochs	500	100

3.6. Performance Evaluation Metrics

Metrics such as accuracy, recall, precision, and mean average precision was used for the performance evaluation of defect detection by the YOLOV5 algorithm. Recall provides information about how accurate the model is in detecting all defects present in the image, whereas precision measures the accuracy of the model in making correct detection. However, all these metrics are affected by the IoU threshold. The true positive is defined as the part of the XCT slices which has been considered a defect in training data annotation. The false positive is considered as detecting the background or metal part as a defect in the image slice. The following expression shows the mathematical relation of those metrics [28]. The values of precision and recall lie between 0 to 1.

$$Precision\ (p(r)) = \frac{True\ Positive}{True\ Positive +\ False\ Positive} \tag{2}$$

$$Recall\ (R(r)) = \frac{True\ \ positive}{True\ Positive + False\ Negative} \tag{3}$$

$$Average\ Precision\ (AP) = \int_0^1 p(r)dr \tag{4}$$

$$Mean\ Average\ Precision(mAP) = \frac{1}{N}\sum_{i}^{N} AP(i) \tag{5}$$

$$mAP\ @0.5{:}\ 0.95 = \frac{mAP_{0.50}\ + mAP_{0.55}\ \ldots + mAP_{0.95}}{N} \tag{6}$$

3.7. YOLO Training

In this study, the latest version of YOLO which is YOLOv5 was applied. YOLOv5 algorithm developed and published by Ultralytics LLC in 2020 [33] as a GitHub repository is implemented for defect detection. The open-source repository contains 4 major models of YOLOv5, namely, YOLOv5*s*, YOLOv5*m*, YOLOv5*l*, and YOLOv5*x*. This paper evaluates the performance of YOLOv5*s*, YOLOv5*m*, and YOLOv5*l* where the subscripts *s*, *m*, and *l* denotes small, medium, and large networks of the YOLOv5 version. The training was carried out with 1648 images containing different-sized defects. The data were randomly split in training validation and test set in the ratio of 0.7: 0.2:0.1, respectively. One of the challenges in defect detection was because of a large range of defect sizes ranging from a few pixels to hundreds of pixel sizes as shown in Figure 5.

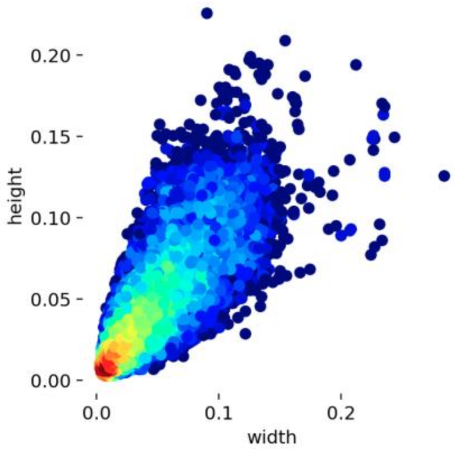


Figure 5. Dataset anchor clustering with X-axis representing annotation width and Y-axis height for YOLO model.

Loss values signify how well or poorly a certain model performs after each iteration of optimization of the model parameters. The training loss values for a different model of YOLOv5 are shown in Figure 6. The trend of change in loss values for YOLOv5*s*, YOLOv5*m*, and YOLOv5*l* are similar. However, the YOLOv5*l* model has relatively low values of loss followed by the YOLOv5*m* and YOLOv5*s* model. Table 2 shows the comparison of 3 different versions of YOLOv5: large, medium, and small. As expected, the YOLOv5*l* model has the highest number of training parameters, training time, number of layers, and model size followed by YOLOv5*m* and YOLOv5*s*.

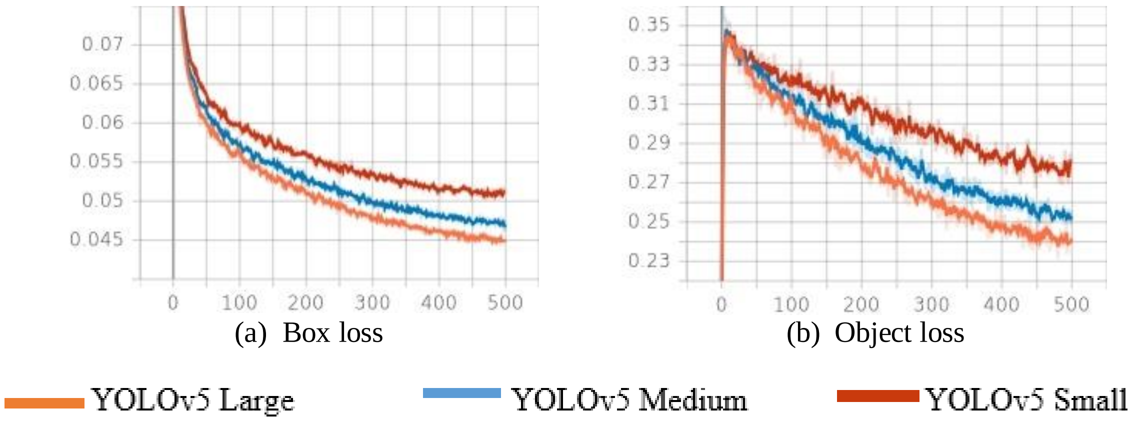


Figure 6. (a) Loss values during the training of YOLOv5: (a) Object loss (b) Box loss

Table 2. Comparison of 3 YOLOv5 models

Item	**YOLOv5*l***	**YOLOv5*m***	**YOLOv5*s***
Trainable Parameters	47,393,334	21,485,814	7,255,094
Training time per epoch	58 sec	42 sec	26 sec

3.8. U-Net Training

For the training, a total of 1648 image slices from two different samples were utilized in this study. The data were randomly split with a ratio of 80% for training and 20% for validation. Training in each epoch carries out in a batch of 2 images and completion of each epoch indicates that all images in the training set are trained. The model was trained for 100 epochs. Figure 7 shows the change in training and validation loss values during the learning process.

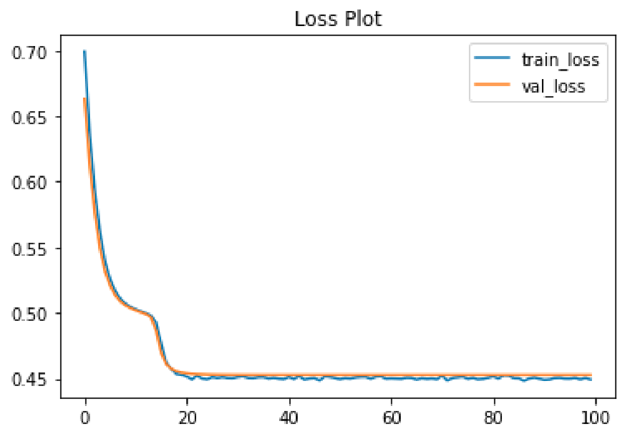


Figure 7. Training and validation loss values during the learning process of U-Net

The function of the optimizer is to update the various parameters that can reduce the loss. In this study, the Adam algorithm [34] with a learning rate of 0.0001 was applied to optimize the model. Adam (Adaptive Moment Estimation) is a modified form of RMSProp optimizer which takes the running average of both the gradients and second moments of gradients. The loss function utilized was a combination of dice loss and binary cross-entropy loss. The dice coefficient metric is analogous to IoU. The dice coefficient is calculated using equation 7. The binary cross-entropy loss function is used in binary classification tasks and is widely used in image segmentation. The binary cross-entropy loss function [35] is expressed as shown in equation 8.

$$\text{Dice coefficient} = 2 \times \frac{|X \cap Y|}{|X| + |Y|} \tag{7}$$

$$J_{bce} = -\frac{1}{M}\sum_{m=1}^{M}[y_m \times log\,(h_\theta(x_m)) + (1 - y_m) \times \log(1 - h_\theta(x_m))] \tag{8}$$

Where X, Y, M, y_m, x_m and h_θ represents prediction, target, number of training examples, target label for training example m, input for training example m, and model with neural network weight θ respectively [7- 8].

4. Results and Discussion

The present study shows that YOLOv5 can be used to classify and localize defects effectively in 2D image slices. Figure 8 shows the performance comparison of YOLOv5*s*, YOLOv5*m* and YOLOv5*l* during training based on evaluation metrics, namely precision, recall, *mAP@0.5*, and *mAP@0.5:0.95*. The *mAP*@0.5 increases rapidly first to reach a peak and then decreases gradually and finally remains constant when training continued beyond 500 epochs. YOLOv5*s* model was found to have relatively good performance with *mAP* at 88.45% for defect detection if

IoU threshold is 0.5. However, for the average precision for the IoU threshold value ranging from 0.5 to 0.95 YOLOv5*m* model performs best followed by the YOLOv5*s* and YOLOv5*l* model. The highest precision value of 71.61% was achieved by YOLOv5*l* followed by 69.19% of YOLOv5*m* and 63.66 % of YOLOv5*s*. Based on recall, relatively YOLOv5*s* perform best with a recall of 87.65% followed by YOLOv5*m* at 85.85% and YOLOv5*l* at 85.01%. Table 3 summarizes the performance of the YOLOv5 models.

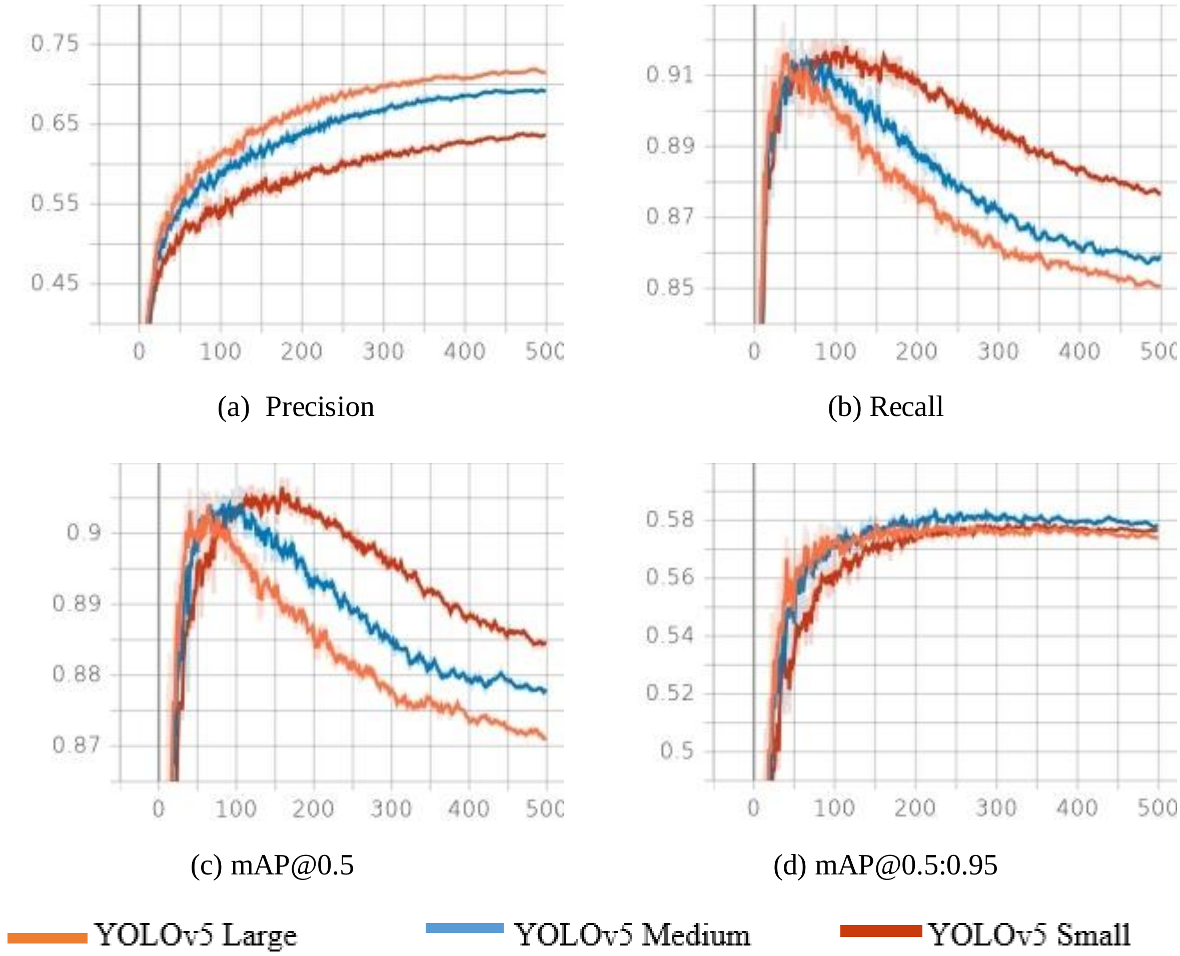
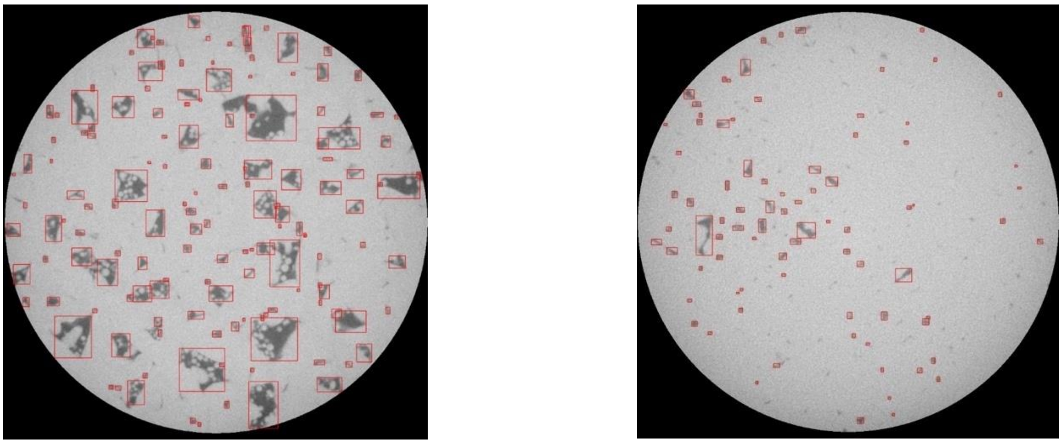


Figure 8. Comparison of recall (at different IoU thresholds) and precision for the different models of YOLOv5

Table 3. Performance comparison of YOLOv5 large, medium, and small model

Item	**YOLOv5*l***	**YOLOv5*m***	**YOLOv5*s***
Recall	85.01%	85.85%	87.65%
Precision	71.61 %	69.19 %	63.66%
Mean Average Precision (mAP@0.5)	87.11%	87.74%	88.45%
Model size	95.3 MB	43.3 MB	14.8 MB
Inference speed	0.019 sec	0.013 sec	0.008 sec

The trained models were able to detect defects of various sizes and shape. Non-max suppression (NMS) is applied to select the best prediction bounding box per defect out of multiple predicted bounding boxes. Figure 9 shows the sample result of defect detection using the YOLov5*s* model developed in this work. The detection result fairly agrees with the annotation provided as an input to the algorithm during the training. The inference speed is the average time taken to detect all defects in the image. The inference was fastest with the YOLOv5*s* model at 8 milliseconds followed by YOLOv5*m* at 13 milliseconds and YOLOv5*l* at 19 milliseconds. The inference was carried out using the same setup mentioned in section 3.4.

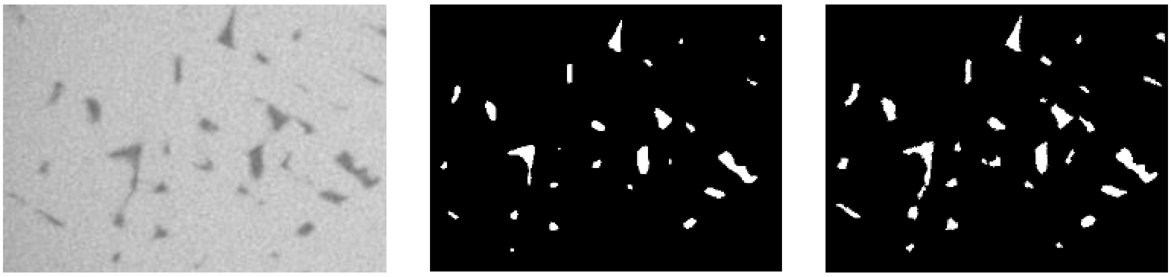


(a) Image slice of sample A

(b) Image slice of sample B

Figure 9. Defect detection result from trained YOLOv5*s* model

It was found that U-Net is capable of segmenting defects from a metal matrix background with a fair level of accuracy as shown in Figure 10. However, some inaccuracy can be seen in the test results which is because of errors in developing annotation. Figure 10 shows the result of U-Net segmentation and comparison with targeted annotation. Figure 10 and figure 11 show that the prediction of segmentation using U-Net agrees well with the target.

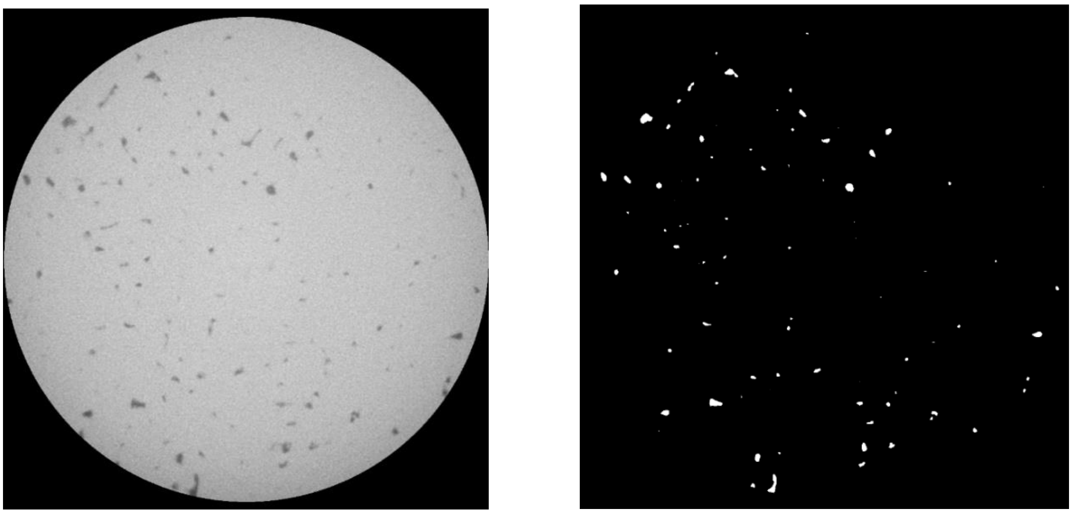


(a) Input raw image

(b) Predicted segmentation

(c) Target segmentation

Figure 10. Result of U-Net segmentation showing a portion of an XCT slice.



(a) (b)

Figure 11. (a) Raw XCT images slice, (b) defect segmentation prediction by trained U-Net model

5. CONCLUSION AND RECOMMENDATION

It is concluded that the state-of-art object detection algorithm YOLOv5 can recognize and localize AM defects in XCT 2D data with reliable accuracy. Additionally, it was found that U-Net, a CNN-based network originally developed for biomedical image segmentation, effectively segments defects from a metal matrix in XCT slices of additively manufactured cobalt chrome specimens. The study was carried out with 3 variants of YOLOv5 which are YOLOv5*l*, YOLOv5*m*, and YOLOv5*s*. For defect detection, it was found that the YOLOv5s model has the highest recall of 87.65%, whereas YOLOV5*l* reported the highest precision of 71.61%. If it is crucial to find all defects present, recall is more important and YOLOv5*s* can be more reliable compared to the other two variants of YOLOv5. Overall, both YOLOv5 and U-Net show reliable results. The result of defect localization by the YOLOv5 algorithm in post-built CT data can assist in data fusion with in-situ monitoring sensor signals to develop a machine learning model to predict the defect formation by analyzing the in-situ sensor data. Thus, it can be concluded that those deep learning techniques can assist in the quality assessment and control of additively manufactured parts and AM processes.

Future work could focus on optimizing the techniques used for a more accurate result. Image processing techniques can be optimized to get better annotation which could improve the result of defect detection and segmentation using YOLOv5 and U-net, respectively. Additionally, this work can be expanded by using the XCT AM data of different materials and geometries. Also, future work can consider redesigning the neural network architecture which may help to extract useful information more effectively and reduce the redundant layers.

REFERENCES

[1] Prakash, K.S., T. Nancharaih, and V.S. Rao, *Additive manufacturing techniques in manufacturing-an overview.* Materials Today: Proceedings, 2018. **5**(2): p. 3873-3882.

[2] Holzmond, O. and X. Li, *In situ real time defect detection of 3D printed parts.* Additive Manufacturing, 2017. **17**: p. 135-142.

[3] Gobert, C., et al., *Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging.* Additive Manufacturing, 2018. **21**: p. 517-528.

[4] Redmon, J., et al. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[5] Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. in *International Conference on Medical image computing and computer-assisted intervention*. 2015. Springer.

[6] Kim, F.H., et al., *High-Resolution X-ray computed tomography (XCT) image data set of additively manufactured cobalt chrome samples produced with varying laser powder bed fusion processing parameters.* 2019, December 13.

[7] Schneider, C.A., W.S. Rasband, and K.W. Eliceiri, *NIH Image to ImageJ: 25 years of image analysis.* Nature methods, 2012. **9**(7): p. 671-675.

[8] Poudel, A. and T.P. Chu, *IntelligentNondestructiveTestingExpertSystemforAircraftCarbonCarbonCompositeBrakesUsingInfraredTh.* Materials Evaluation, 2012. **70**(10).

[9] Pan, Y.-C., T.P. Chu, and P. Filip. *Intelligent Nondestructive Evaluation Expert System for Carbon-Carbon Composites Using Infrared Thermography*. in *Fall Conference 2010*. 2010.

[10] Kim, F., et al., *Investigation of pore structure in cobalt chrome additively manufactured parts using X-ray computed tomography and three-dimensional image analysis.* Additive Manufacturing, 2017. **17**: p. 23-38.

[11] Sezgin, M. and B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation.* Journal of Electronic imaging, 2004. **13**(1): p. 146-166.

[12] Tokime, R.B., X. Maldague, and L. Perron, *Automatic Defect Detection for X-Ray inspection: Identifying defects with deep convolutional network.*

[13] Badrinarayanan, V., A. Kendall, and R. Cipolla, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation.* IEEE transactions on pattern analysis and machine intelligence, 2017. **39**(12): p. 2481-2495.

[14] Kwon, O., et al., *A deep neural network for classification of melt-pool images in metal additive manufacturing.* Journal of Intelligent Manufacturing, 2020. **31**(2): p. 375-386.

[15] Davtalab, O., et al., *Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection.* Journal of Intelligent Manufacturing, 2020: p. 1-14.

[16] Gonzalez-Val, C., et al., *A convolutional approach to quality monitoring for laser manufacturing.* Journal of Intelligent Manufacturing, 2020. **31**(3): p. 789-795.

[17] Alqahtani, N., et al., *Machine learning for predicting properties of porous media from 2d X-ray images.* Journal of Petroleum Science and Engineering, 2020. **184**: p. 106514.

[18] Mutiargo, B., et al., *Evaluation of X-Ray Computed Tomography (CT) Images of Additively Manufactured Components using Deep Learning.* 2019.

[19] Li, X., et al., *Quality analysis in metal additive manufacturing with deep learning.* Journal of Intelligent Manufacturing, 2020. **31**(8): p. 2003-2017.

[20] Lee, J.-S., et al., *Defect Classification of Cross-section of Additive Manufacturing Using Image-Labeling.* Journal of the Korean Society of Manufacturing Process Engineers, 2020. **19**(7): p. 7-15.

[21] Du Rand, F., P. Van Tonder, and A. van der Merwe, *APPLICATION OF MACHINE LEARNING MODELS FOR POWDER BED DEFECT ANALYSIS.*

[22] Ren, S., et al., *Faster r-cnn: Towards real-time object detection with region proposal networks.* arXiv preprint arXiv:1506.01497, 2015.

[23] Eyupoglu, C., *Implementation of Bernsen's locally adaptive binarization method for gray scale images.* The Online Journal of Science and Technology, 2017. **7**(2): p. 68-72.

[24] Culjak, I., et al. *A brief introduction to OpenCV*. in *2012 proceedings of the 35th international convention MIPRO*. 2012. IEEE.

[25] Lin, T. *LabelImg*. 2021; Available from: https://github.com/tzutalin/labelImg.

[26] Siddique, M.J. and K.R. Ahmed, *Deep Learning Technologies to Mitigate Deer-Vehicle Collisions.* Deep Learning and Big Data for Intelligent Transportation: Enabling Technologies and Future Trends: p. 103.

[27] Kharel, S. and K.R. Ahmed. *Potholes Detection Using Deep Learning and Area Estimation Using Image Processing*. in *Proceedings of SAI Intelligent Systems Conference*. 2021. Springer.

[28] Padilla, R., S.L. Netto, and E.A. da Silva. *A survey on performance metrics for object-detection algorithms*. in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. 2020. IEEE.
[29] Kirk, D. and B.S. Center, *NVIDIA CUDA software and GPU parallel computing architecture.* 2008.
[30] Ari, N. and M. Ustazhanov. *Matplotlib in python*. in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*. 2014. IEEE.
[31] Oliphant, T.E., *A guide to NumPy*. Vol. 1. 2006: Trelgol Publishing USA.
[32] Abadi, M., et al., *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.* arXiv preprint arXiv:1603.04467, 2016.
[33] Glenn Jocher, A.S., Jirka Borovec,NanoCode012,ChristopherSTAN, Liu Changyu,Lijun Yu 于力军. *ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration*. 2021 01/05/2021 [cited 2021 01/13/2021]; Available from: https://github.com/ultralytics/yolov5.
[34] Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.
[35] Ho, Y. and S. Wookey, *The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling.* IEEE Access, 2019. **8**: p. 4806-4813.

AUTHORS

Pradip Acharya received master's degree in Mechanical Engineering from Southern Illinois University Carbondale in 2021. He worked in several research and development in the area of non-destructive evaluation and certifying those technologies from federal aviation administration (FAA). Currently, he is working as Packaging R and D Engineer in semiconductor industry. His research interest includes non-destructive evaluation, computer vision, electronic packaging, optical inspection and equipment development.

Dr. Tsuchin "Philip" Chu is a senior Professor in the School of Mechanical, Aerospace, and Materials Engineering. Dr. Chu has conducted research for over 30 years in areas such as non-destructive evaluation (NDE), biomedical engineering, experimental mechanics, computer-aided design, manufacturing, and engineering (CAD/CAM/CAE), finite element analysis (FEA), sensors and instrumentation. He is a pioneer in the area of digital image correlation (DIC) and at the cutting edge of research in NDE and biomechanics.

Dr. Khaled R. Ahmed is currently an assistant professor and Graduate Program director in school of computing at Southern Illinois University, Carbondale. His research interests include deep learning, machine learning, computer vision, high performance computing, and big data.

Subash Kharel graduated with a master's degree in Computer Science from Southern Illinois University in Spring 2021. His research interest lies in the field of Deep Learning and Artificial Intelligence. He is also working as a software developer and has 3 years of experience in designing enterprise level software solutions with incorporating cutting edge technology.