

Automating Visual Inspection with Convolutional Neural Networks

Christopher D. Hollander¹, Sreerupa Das², Suraiya Suliman³

^{1,2,3}*Lockheed Martin Corporation, Orlando, FL, 32825, USA*

christopher.d.hollander@lmco.com

sreerupa.das@lmco.com

suraiya.h.suliman@lmco.com

ABSTRACT

Convolutional Neural Networks (CNNs) have become the recent tool of choice for many visual detection tasks, including object classification, localization, detection, and segmentation. CNNs are neural networks designed around the concept of weight sharing and the convolution operator. CNNs have been proven to work especially well for analyzing grid-like data, e.g. images. One of the key features of a CNN is its ability to automatically detect important features within an image (e.g. edges, patterns, shapes); prior to CNNs, these features had to be manually engineered by subject matter experts.

Inspired by the significant achievements and success that CNNs have experienced in the domain of computer vision, we examine a specific convolutional neural network (CNN) architecture, U-Net, suited for the task of visual defect detection. We identify and discuss situations for the use of this architecture in the specific context of external defect detection on aircraft and experimentally discuss its performance against a dataset of common visual defects.

One requirement of training Convolution Networks on an image analysis task is the need for a large image (training) data set. We address this problem by using synthetically generated images from computer models of jets with varying angles and perspectives with and without induced faults in the generated images. Data augmentation is used to further expand the size of the training set. This paper presents the initial results of using CNNs, specifically U-Net, to detect aerial vehicle surface defects of three categories. We further demonstrate that CNNs trained on synthetic images can then be used to detect faults in real images of jets with visual damages. The results obtained in this research, indicate that our approach has been quite effective in detecting surface anomalies in our tests.

1. VISUAL INSPECTION

Visual inspection is one of the first steps, and a key component, of many traditional maintenance and operational processes soon after an aircraft completes a flight. Being able to get an aircraft off the ground and make it available and ready for the next mission as quickly as possible, i.e., minimize downtime, is of utmost interest to aircraft owners, maintainers, and flight crew. Being able to quickly, correctly, and consistently locate all visual defects, and determine their type and severity, is critical to the functionality of the device and the safety of its operators. Sole reliance on pilot and maintainer walk-around inspection can be time consuming, error prone and expensive. Human fault due to stress, fatigue, distractions, etc. can result in defects being overlooked and undetected. Furthermore, manual detection cannot always be used to discover defects in hard to reach or unsafe areas.

In the case of an Unmanned Aerial Vehicle (UAV), automatic visual inspection may be the only option due to lack of a human pilot to do the walk around and detect visual defects after the vehicle lands. Inspection would have to be completed autonomously based on images collected either using a drone flying around the vehicle or make the UAV land in a hanger fitted with cameras. In either case, interpretation of the images to determine vehicle health, locate defects, and determine type, severity and criticality of the unmanned vehicle must be completed quickly and efficiently.

Automated visual inspection not only facilitates quicker and more consistent inspection, it also enables automating downstream maintenance steps. For example, upon detection of an anomaly or defect, an automated work order could be created for taking necessary maintenance actions (repair or inspection) needed that would otherwise have to be created by a human maintainer upon detection of a problem, thus reducing the mean time to repair and increasing availability. Furthermore, if replacement of a part is necessary, it could even be ordered ahead of time automatically, further lowering the downtime.

Christopher D. Hollander et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. The images used in the paper are ©2019 Lockheed Martin Corporation

In this paper we focus on automating the visual inspection of fixed wing aircrafts with external damages. We approach this problem as an instance of semantic segmentation, with the primary goal being to identify a pixel-by-pixel match of the damaged areas. An example of external damage is shown in Figure 1.



Figure 1. Image of an exterior damage.

2. SEMANTIC SEGMENTATION

2.1. What is Semantic Segmentation?

Semantic Segmentation is one of the key problems in the field of computer vision. It is a process in which an image is partitioned into coherent objects such that each pixel in the image is classified as being part of a person, a car, a bus or any other entity of interest in the image. Semantic segmentation paves the way towards comprehending content and meaning of the image. The importance of semantic segmentation or scene understanding is highlighted by the fact that an increased number of applications seek to infer knowledge from images (or videos), especially in the domains of self-driving cars and medical imaging.

2.2. CNNs for Semantic Segmentation

In the last decade, Convolutional Neural Networks (CNNs) have become the tool of choice for many computer vision problems, including semantic segmentation (Alex Krizhevsky et. al., 2012; Simonyan, Karen & Zisserman, Andrew, 2015; Szegedy, Christian, et. al., 2015; He, Kaiming, et. al., 2016). Semantic segmentation has become increasingly important in the medical sciences, where CNNs have been shown to significantly improve speed and accuracy of interpreting medical images (Meheta, Ra. & Sivaswamy, J., 2017, Shoji Kido et. al., 2018, Grinsven, Mark J.J. et al, 2016).

One approach to building CNN architectures for semantic segmentation is to consider an encoder/decoder framework. A pre-trained classification network like VGG or ResNet is used as an encoder and learns the discriminative features of the data. That encoded layer is then connected to a decoder network that maps the features on to the actual pixels in the image.

In general, both the encoder and decoder are variations of a Fully Convolutional Network (FCN) (Shelhamer, Evan et. al., 2015). FCNs use a mixture of convolutional and pooling layers to extract image features, and deconvolution layers to relate a set of image features to the original pixels. One downside of using an FCN is that by propagating through several alternated convolutional and pooling layers, the resolution of the output feature maps is down sampled, and hence the classifications of FCN typically have low resolution with fuzzy object boundaries. Multiple enhancements to FCN architectures have been proposed such as SegNet (Badrinarayanan, Vijay et al., 2017) and DeepLab-CRF (Chen, Liang-Chih, et. al., 2015). Our work leverages a related architecture known as U-Net (Ronneberger, Fischer, & Brox, 2015).

2.3. Defect Detection

Inspection for anomaly and defect detection is a special case of Semantic Segmentation. In many ways defect detection is a much harder task compared to semantic segmentation on regular shapes such as people, buildings, and cars in an urban scene, where the shapes of objects are regular and well defined. Similarly, objects in medical images are well understood and typically have well defined shapes with some variations. The defects we are looking to identify are highly irregular such as cracks, punctures, and corrosion.

There has been limited research on detecting defects in the last few years, with the main focus centered on the problems of identifying defects in steel strip surface detection (Ren, Qirui, et. al., 2018), defects in tire by analyzing X-ray images (Zhu, Qidan, Ai, Xiaotian, 2018), metallic surface defect detection (Tao, Xian, et. al., 2018), fracture propagation (Miller, Robyn, et. al., 2017), PCB defect detection (Adibhatia, V.A. et.al., 2018), rail surface defect detection (Shang, Lidan, et. al., 2018), railway infrastructure defect recognition (Hyang, Huaxi, et. al., 2018). For majority of these detection tasks, closeup images were used to detect occurrence or absence of unexpected change in grayscale as an indication of defect. The research described in this paper extends the technology to more realistic images where a mounted camera can get a wider view of the vehicle as compared to pilot or maintainer walking around would have seen. This also allows this research to be extended to maintenance of unmanned vehicles where the walk around could be completely automated.

2.4. Use of Synthetic Images

One of the greatest difficulties with training a CNN to solve a semantic segmentation problem is getting enough training data. The training set images typically have to be marked or preprocessed to create masks corresponding to each source image to serve as the target. When using real images, these masks are often generated by hand. This is not feasible for large datasets. A growing solution to this challenge is to use

synthetic data – 2D or 3D models generated on a computer along with the corresponding mask.

Zang, Yang et. al. (2017), investigated the roll of synthetic images for semantic segmentation in urban scenes. One key element of their research was that their choice of urban scenes was based on the regularity of many objects found in urban environments. Unfortunately, damage that occurs to an airframe is rarely regular.

Staar, Benjamin, et.al. (2018) and Tao, Xian, et. al. (2018) have used the publicly available DAGM (DAGM 2007) dataset for training. The DAGM dataset contains synthetically generated images that have some semblance to surface defects. Although labelled, the images were not suitable for our use as our application was more specific to defects on the exterior of air vehicles.

Tremblay, J., et. al. (2018) have used synthetic images for object recognition. They demonstrated the possibility of using inexpensive synthetic data for training neural networks while avoiding the need to collect large amounts of hand annotated real-world data or to generate high-fidelity synthetic scenes. We were inspired by this finding and extended the idea of using synthetic images for defect detection.

Due to challenges of obtaining defect images of jets (either they were unavailable or were considered sensitive), we used synthetically generated images from publicly available computer models of jets with varying angles, lightings and perspectives with and without induced faults for our training. Faults were artificially generated and overlaid over the jet images. One advantage of using synthetic images is that creating masks to indicate where the damage is (essential for training) becomes easy. The masks can be generated by a computer alongside the synthetic defects. Moreover, since many aircraft typically have smooth and streamlined bodies with limited color variations, synthetic images of aircraft appear much closer to real images of aircraft (unlike synthetic images of natural scenery or animals with complex texture and color).

3. CONVOLUTIONAL NEURAL NETWORKS FOR DEFECT DETECTION

To bound the domain our initial investigation of defect detection, this research focused on three distinct surface defects – areas of corrosion, cracks, and punctures. These are the types of defects that would typically be identified during a human inspection. Figure 2 illustrates three samples of these defects.

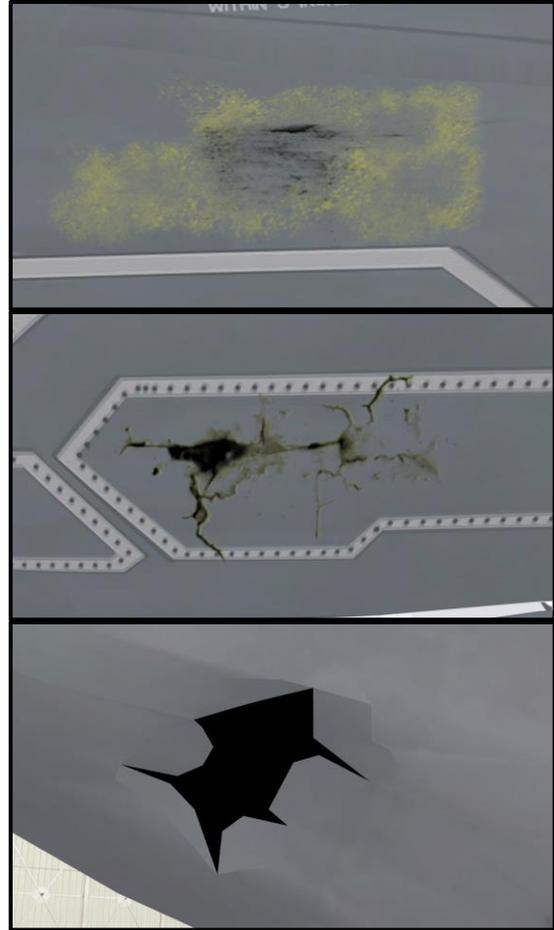


Figure 2. Examples of three types of surface defects focused on in this research: corrosion (top), cracks (middle), puncture (bottom).

3.1. Network Architecture

We used a fully convolutional auto-encoder-decoder network for defect detection and trained the model using a dataset of synthetic fixed-wing aircraft. Our network architecture was inspired by the work on U-Net (Ronneberger, Fischer, & Brox, 2015). Figure 3 displays a diagram of the original U-Net architecture. U-Net is an auto-encoder/decoder convolutional neural network with links between the encoding layer and decoding layers.

We updated and extended the U-Net architecture with the following changes: (1) our input images are 512x288 pixels; (2) we use 3 convolutional layers in each block instead of 2 layers; (3) we use a convolutional layer with a stride of 2 for down-sampling, instead of a max pooling layer; (4) we use a transposed convolutional layer for up-sampling, instead of a basic up-sampling layer; (5) we set the padding so that each standard convolutional layer is the same size, instead of cropping the layers during the copy step; (6) the output layer contains one filter per defect.

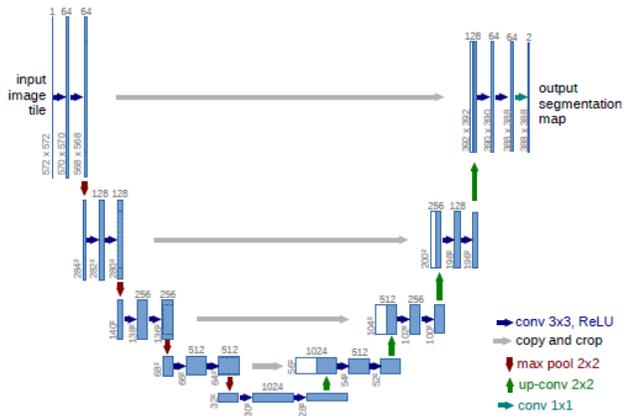


Figure 3. The original U-Net architecture developed by Ronneberger, Olaf et al. (2015).

The output layer of the network is a per-pixel softmax layer, implemented as a 2D convolutional layer with a kernel size of 1 and 4 filters (because there are 3 defect classes plus the background). The cross-entropy function is used as the loss function. The Adam optimization algorithm (Kingma & Ba, 2015), with a learning rate of 0.001, was used to train the network. We assess the performance of our model through a mean intersection over union (IoU) metric (Shelhamer, Long, & Darrell, 2017) that ignores background classification.

3.2 Training Data

There are a limited number of datasets that support defect detection on fixed-wing aircraft. We overcame this issue by leveraging publicly available 3D models and commercial photo manipulation software.

Figure 4 displays an example of our training data paneled as a sequence of 8 inputs. Each input image has a corresponding mask that indicates where a defect is located on the aircraft surface.

To create the input images, a graphic artist added the defects as textures on the 3D model. Multiple frames of the model were then saved at different angles to simulate what an aircraft maintainer might see if they were to walk around the plane.

The first step to create the input masks as to generate a separate mask for each defect on the aircraft. To create these defect masks, each defect was altered to reflect light while lighting on the rest of the model was disabled. This technique allows us to generate an exact mask for any component of interest. These separate masks were then combined into a single image that reflected the total number of defects visible in any given image. Each defect was associated with a class (1, 2, or 3) and the shade of that defect in the input mask updated to reflect its class.

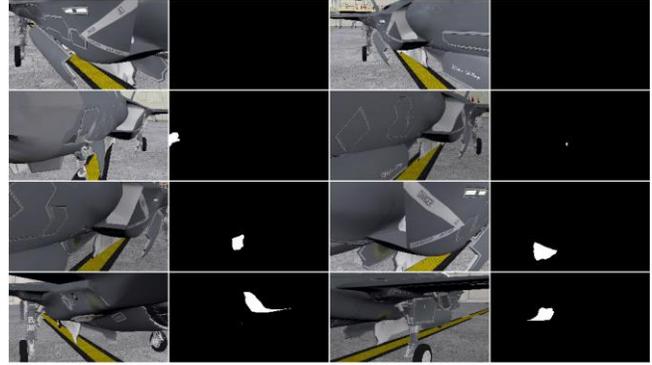


Figure 4. An example of training data fed to the network

The following training images were generated using the synthetic image generation process:

- 361 images of a damaged aircraft model (paint corrosion, surface cracks, and impact punctures)
- 361 image masks of the damaged areas
- 361 images of an undamaged aircraft model

Each input image and associated mask has a native resolution of 1920x1080 pixels. This resolution is reduced to 512x288 pixels during preprocessing

These 361 images and their associated masks were then split such that 20% were used in the test set, 20% in the validation set, and 60% for the training set.

During training, these images were further augmented by rotation, shifting along the x and y axes, shearing, zoom, and horizontal and vertical flips.

4. RESULTS

The convolutional neural network was set to train with a batch size of 4 for 200 epochs at 116 steps per epoch. An early stopping mechanism was employed to halt training once there was no noticeable improvement for 10 consecutive epochs. This resulted in a total training time of 88 epochs.

At the end of training, our network yielded a mean IoU value of 0.836. This value indicates that our predicted masks cover 83.6% of the true defect area in our test set. Furthermore, this value indicates that our system can be expected to perform reasonably well on unseen data.

Figure 5 displays the predicted mask from an image in our test set. The mask is colored red to make it stand out from the aircraft. It can be observed that the prediction accounts for the complex shape of the aircraft, highlighting only the damaged area and ignoring the landing gear panel in the foreground.

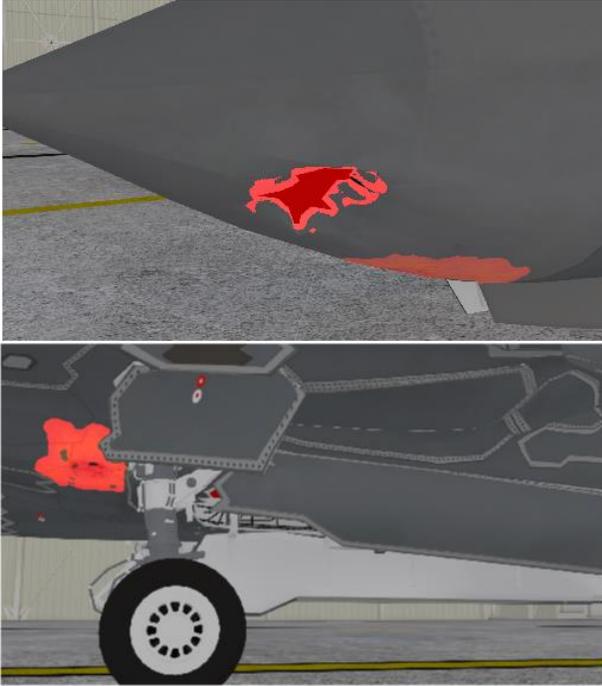


Figure 5. Predicted masks overlaying puncture (above) and corrosion defects (below) on the surface of our test model.

Figure 6 displays the predicted mask from a real image with corrosion photo-shopped on to the nose cone. From this result, we can see that the damage was easily detected, however the hanger door – which is of similar color – was also marked as a defect. This result indicates that our network would benefit from retraining with additional images that include a larger variety of backgrounds and colors.



Figure 6. The output of our model against a real image with simulated paint corrosion on the nosecone. Test images are on the left. Images with the defect prediction overlaid on the test images are on the right. The top row is an image with defect and its prediction. The bottom row is an image with no defect and its prediction.

5. DISCUSSION/BROADER IMPACT

Surface inspection of manned and unmanned aerial vehicles (UAV) for damages is a labor and time intensive process. In

cases where human inspection is not possible due to remote landings of UAVs, damage assessment is curtailed by lack of visual inspections. Automatic visual inspection and damage detection has the potential to transform this labor intensive and error prone task into Just-In-Time (JIT) (Yasuhiro Monden, 1993) decision support. Using the technology described in this paper and image feed from a camera mounted drone or on the hanger where the UAV is parked, an automated walkthrough of the UAV could be performed remotely.

This paper presents the initial results of using CNNs, specifically a modified version of U-Net, to detect aerial vehicle surface defects of three categories. The results - a mean IoU value of 0.836 - indicate that our approach has been quite effective in detecting surface anomalies in our tests. Given the limited synthetic data sets we've used to train the model to achieve our object detection accuracy, we expect that increasing the quality and diversity of training data will increase the effectiveness of the model. Approaches to creating a richer, more realistic set of synthetic data will include using depth camera to reconstruct 3D models of aerial vehicles, generating various lighting effects using ray tracing, adding gaussian and salt and pepper noise, and trying non-uniform image resolutions. We shall also be looking at generating a richer set of damages based on real images collected at our maintenance centers. Automatics detection of surface anomalies can trigger further inspection to classify damages for further action.

The surface defect detection technology that we've developed can be applied to a broad range of domains. We expect that this technology can be readily transferred to land and naval vehicles. Automated damage inspection of a specific kind such as corrosion in the aerial, land and naval domain can have huge implications in the way we maintain and service our equipment

6. CONCLUSIONS AND FUTURE WORK

We've presented an approach to detecting three categories of surface defects on a fixed wing aircraft using the U-Net architecture and synthetically generated training data. To mitigate the fact that we lacked a dataset of real-world damaged aircrafts and their associated damage masks, we generated a set of synthetic data based on a 3D aircraft model. We trained a CNN with 361 images split into training, validation and test sets. We achieved a mean IoU value of 0.836 at a total training time of 88 epochs. The CNN trained on synthetic images generated using simple wireframe rendering showed promising results in detecting damage on real aircraft images.

We expect that training the CNN with more realistic synthetic images and a wider variety of defects will improve the detection performance. Our future work will involve generating test images that are more realistic by using more advanced rendering techniques such as ray tracing and

various lighting models. We shall also explore generating more complex images in realistic environments.

After discussions with aircraft maintenance subject matter experts (SMEs), we've identified surface corrosion as one of the main defects of interest. Corrosion manifests in multiple ways on aircraft exterior such as rust spot, paint bubbles etc. We shall focus our generated data for corrosion detection use cases.

REFERENCES

- Adibhatia, V.A. et.al. (2018). Detecting Defects in PCB using Deep Learning via Convolution Neural Networks, *13th International Microsystems, Packaging, Assembly and Circuits Technology Conference*, pp 202-205.
- Badrinarayanan, Vijay et al. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp 2481-2495.
- Chen, Liang-Chich,, et. al. (2015) Semantic Image Segmentation with Deep Convolution Nets and Fully Connected CRFS, in *International Conference on Learning Representation*.
- DAGM. (2007) Deutsche Arbeitsgemeinschaft für Mustererkennung e.V., *German chapter of the IAPR (International Association for Pattern Recognition)*. Retrieved from <https://resources.mpi-inf.mpg.de/conference/dagm/2007/prizes.html>
- Grinsven, Mark J.J. et al (2016). Fast Convolutional Neural Network Training Using Selective Data Sampling: Application to Hemorrhage Detection in Color Fundus Images, *IEEE Transactions on Medical Imaging*. Volume: 35, Issue: 5_Pages: 1273 – 1284
- He, Kaiming, et. al. (2016). Deep Residual Learning for Image Recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp 770-778.
- He K, Gkioxari G, Dollár P, et al. (2017). “Mask R-CNN,” in *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. (99):1-1.
- Hyang, Huaxi, et. al. (2018). Railway Infrastructure Defects Recognition using Fine-grained Deep Convolutional Neural Networks, *Digital Image Computing: Techniques and Applications*, pp 1-8.
- Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations*. San Diego.
- Krizhevsky, Alex et. al (2012) ImageNet classification with deep convolutional neural networks, in *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, Volume 1, pp 1097-1105
- Meheta, R. & Sivaswamy, J. (2017). M-net: A Convolutional Neural Network for deep brain structure segmentation, in *2017 IEEE 14th International Symposium on Biomedical Imaging*, pp. 437 – 440.
- Miller, Robyn, et. al. (2017). Image Analysis using Convolutional Neural Networks for Modeling 2D Fracture Propagation, *IEEE International Conference on Data Mining Workshop*, pp 978-982.
- Ren, Qirui, et. al. (2018). “Slighter Faster R-CNN for real-time detection of steel strip surface defects”, Chinese Automation Congress, pp 2173-2178.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Springer International Publishing. doi:10.1007/978-3-319-24574-4_28
- Simonyan, Karen & Zisserman, Andrew (2015). Very Deep Convolution Networks for Large-Scale Image Recognition, in *International Conference on Learning Representation*.
- Szegedy, Christian, et. al. (2015). Going Deeper with Convolutions, in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.
- Shelhamer, E., Long, J., & Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 640–651. doi:10.1109/tpami.2016.2572683.
- Staar, Benjamin, et.al. (2018). Anomaly detection with convolutional neural networks for industrial surface inspection, *12th CIRP Conference on Intelligent Computation in Manufacturing Engineering*.
- Shang, Lidan, et. al. (2018). Detection of rail surface defects based on CNN, in *20th International Conference on Advanced Communication Technology*, pp 45-51.
- Shelhamer, E., Long, J., & Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 640–651. doi:10.1109/tpami.2016.2572683
- Shoji Kido et al (2018). Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN), *International Workshop on Advanced Image Technology*, pages 1-4.
- Tao, Xian, et. al. (2018). Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks, *Applied Science*, 8, 1575; doi:10.3390/app8091575.

- Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., Birchfield, S. (2018). Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
doi:10.1109/cvprw.2018.00143.
- Yasuhiro Monden (1993). *Toyota Production System: An Integrated Approach to Just-In-Time*. Springer; 1994 edition (September 12, 1993), ISBN-13: 978-0412582202.
- Zang, Yang et. al. (2017). Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. in *IEEE International Conference on Computer Vision*. pp 2039-2049.
- Zhu, Qidan, Ai, Xiaotian (2018). The Defect Detection Algorithm for Tire X-ray Images Based on Deep Learning, *IEEE 3rd International Conference on Image, Vision and Computing*, pp 138-142.
- Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering system*. Hoboken, NJ: John Wiley & Sons, Inc