

SYSTEM DEVELOPMENT OF IMAGE DATA AUGMENTATION USING GENERATIVE ADVERSARIAL NETWORK (GAN)





LIM HOU HUA

FACULTY OF MANUFACTURING ENGINEERING 2022

DECLARATION

I hereby, declared this report entitled "System Development of Image Data Augmentation using Generative Adversarial Network (GAN)" is the result of my own research except as cited in references.

Signature : LIM HOU HUA Author's Name : 28 June 2022 Date UNIVERSITI **TEKNIKAL MALAYSIA MELAKA**

APPROVAL

This report is submitted to the Faculty of Manufacturing Engineering of Universiti Teknikal Malaysia Melaka as a partial fulfilment of the requirement for Degree of Manufacturing Engineering (Hons). The member of the supervisory committee is as follow:

(DR. NUR AIDAWATY BINTI RAFAN) DR. NUR AIDAWATY BINTI RAFAN Timbalan Dekan (Akademik) Fakulti Kejuruteraan Pembuatan Universiti Teknikal Malaysia Melaka UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Sebuah projek dicadangkan untuk menyiasat kegunaan Generative Adversarial Netowork (GAN) untuk menambah data imej demi mengatasi masalah kelas tidak seimbang dalam klasifikasi wafer silikon yang mempunyai kecacatan calar. Masalah kelas tidak seimbang ini disebabkan oleh bilangan sampel rosak yang kecil berbanding bilangan sampel yang elok dalam pembuatan. Mask GAN telah digunakan untuk menghasilkan imej label daripada input kotak sempadan. Seterusnya, imej label dimasukkan kepada Defect Image GAN untuk "melukis" kecacatan pada imej input yang elok. Kebolehlaksanaan system telah dikaji dengan mengukur ketepatan algoritma pembelajaran mesin yang dilatih dengan data janaan GAN. Di samping itu, rangkaian saraf yang dilatih dengan data janaan GAN dan data janaan kaedah konvensional telah dibanding. Sistem yang dicadangkan telah diuji dengan McNemar's Test pada nilai keyakinan 95% dan menghasilkan peningkatan yang ketara berbanding kaedah konvensional.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRACT

A project is proposed to investigate the use of Generative Adversarial Networks (GAN) to augment image data to overcome class imbalance problems in silicon wafer microcrack defect classification. This class imbalance problem is due to the small number of defective products compared to non-defective products in a production environment. A Mask GAN is used to generate images of defect mask label when bounding box information is supplied. The artificial defect masks are used by a Defect Image GAN to "paint" defects onto non-defect images. The system's feasibility is analyzed by evaluating the accuracy of the classifier trained on the augmented data, along with the number of misclassified images of defective products. Two identical classifier networks are trained on traditionally augmented data and GAN-augmented data and be compared. The system is shown with McNemar's test with a confidence of 95%, to produce a significant improvement over conventional methods.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEDICATION

Dedicated to the joy of my family, Dolly.



ACKNOWLEDGEMENT

A thousand thank you to those who motivate, support and guide me along the journey of my final year project. I would first like to thank my final year project supervisor Dr. Nur Aidawaty binti Rafan who is always enthusiastic in providing guidance to our final year project group. Furthermore, I am grateful to have Dr. Zamberi bin Jamaludin, Dr. Shariman bin Abdullah, and Dr. Nazmin bin Mazlan for providing valuable insights in my thesis evaluation. In addition, I thank Mr. Lee Choon Teck and all my acquaintances at ViTrox who inspire me to further my interest in machine vision. Moreover, special thanks to all my friends who are there by my side and going through the challenges of studying during this pandemic period together. Finally, I offer my deepest appreciation to my family without whom I will never be where I am today.

> اونيۈم سيتي تيڪنيڪل مليسيا ملاك UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

ABS	TRAK		i	
ABS	TRACT		ii	
DED	ICATIO	Ν	iii	
ACK	NOWLE	EDGEMENT	iv	
TAB	LE OF C	CONTENTS	v	
LIST	OF TAF	BLES	viii	
LIST	OF FIG	URES	ix	
		BREVIATIONS	vi	
		SKEVIATIONS	AI	
CHA	PTER 1	: INTRODUCTION		
1.1	Projec	t Background	1	
1.2	Proble	m Statement	3	
1.3	Object	tives	4	
1.4	Scope		4	
1.5	Projec	t Rationale	5	
1.6	Projec	اويوم سيخ تيڪنيڪل مليو Methodology	5	
1.7	Thesis	Organization	7	
		JNIVERSITI TEKNIKAL MALAYSIA MELAKA		
	APIEK 2	Classification Backlam	0	
2.1		Classification Problem	8	
	2.1.1	Mashing Learning hased Classification	9	
~ ~	2.1.2 Deep 1	Machine Learning-based Classification	10	
2.2	2 2 1	Network Architecture	11	
	2.2.1	Activation Functions	11	
	2.2.2	Loss Function	12	
	2.2.3	Backpropagation and Optimization	12	
	2.2.1	Regularization and Normalization	13	
2.3	Convo	olution Neural Networks (CNN)	13	
	2.3.1	Convolution Operation	15	
	2.3.2	Common CNN used for Classification	16	
2.4	Image	Data Augmentation Methods	17	

2.5	Genera	ative Adversarial Networks (GAN)	18
	2.5.1	GAN Framework	18
	2.5.2	Deep Convolution GAN (DCGAN)	19
	2.5.3	Conditional GAN	19
	2.5.4	MSG-GAN	20
	2.5.5	Sequential Image Generation using GAN	21
2.6	Deep Learning Methodology		22
	2.6.1	Transfer Learning	22
	2.6.2	Model Validation	22
	2.6.3	Model Selection	23
	2.6.4	Performance Metric	24
	2.6.5	Software	25
2.7	Relate	ed Works	26

CHAPTER 3: METHODOLOGY

3.1	Project	roject Overview		
3.2	Design	of Experiment	30	
3.3	Project	t Software and Tools	31	
	3.3.1	Programming Language, Libraries and Platform	31	
	3.3.2	Transfer Learning	31	
	3.3.3	Data Source	32	
	3.3.4	Neural Network Configuration	33	
	3.3.5	Conventional Data Augmentation Methods	34	
	3.3.6	Performance Evaluation Metric ALAYSIA MELAKA	34	
СНА	PTER 4	: RESULTS AND DISCUSSIONS		
4.1	GAN I	mage Generation	35	
	4.1.1	Mask GAN Output	35	
	4.1.2	Defect GAN Output	36	
4.2	Classif	fier Performance Analysis	37	
	4.2.1	Rotate and Flip Augmented Data	37	
	4.2.2	GAN-Augmented Data	38	
	4.2.3	McNemar's Test	39	
	4.2.4	Loss Curve	40	
СНА	PTER 5	: CONCLUSIONS AND RECOMMENDATIONS		
51	Conclu	ision	/11	

5.1	Conclusion	71
5.2	Recommendations	41

APPI	ENDICES	50
REF	ERENCES	43
5.5	Basic Entrepreneurship	42
5.4	Project Complexity	42
5.3	Sustainability of Project	42



LIST OF TABLES

Table 2.1: Example of Four Common Activation Functions	12
Table 2.2: Common CNN Architectures used in Research	16
Table 2.3: Comparison of Data Augmentation Effect on CNN Performan	nce (Taylor &
Nitschke, 2018)	17
Table 2.4: Performance Metric of Classification Model	24
Table 2.5: Performance Metric of GAN	24
Table 2.6: Summary of Related Works	26
MALAYSIA 4	
Table 3.3: Conventional Data Augmentation Methods Used.	34
Table 3.4: Performance Metrics Used	34
اونيوم سيتي تيكنيكل مليسيا ملاك	

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

Figure 1.1: Architecture of a Typical GAN from (An Intuitive Introduction to Genera	ntive
Adversarial Networks (GANs), 2018)	2
Figure 1.2: Visualization of the Class Imbalance Problem in Classification (Kushw	aha,
2019).	3
Figure 1.3: Overview of Project	5
Figure 1.4: Flow Chart of the Proposed Project	6
Figure 2.1: Samples from insect classification dataset (Wu et al., 2019).	8
Figure 2.2: Workflow of Supervised Learning (A. Zhang et al., 2021)	10
Figure 2.3: Example of Simple Feedforward Neural Network (Goodfellow et al., 2016)	11
Figure 2.4: Example of Convolutional Neural Network Architecture (Vasan et al., 2020).14
Figure 2.5: Moving 3x3 Filter Across Image to Detect Features (Glassner, 2021).	15
Figure 2.6: Overview of GAN Framework (Harada et al., 2019).	18
Figure 2.7: Framework of Conditional GAN (Mirza & Osindero, 2014)	19
Figure 2.8: Overview of MSG-GAN architecture	20
Figure 2.9: Overview of Sequential Image Generation GAN	21
Figure 2.10: Illustration of 10-fold Cross Validation (Berrar, 2018).	23
Figure 2.11: Symbol of Python and Google Colab (Parmar, 2021).	25
Figure 2.12: Symbol of Pytorch (left) and Tensorflow (right) (PyTorch vs TensorFlow, 20	021).
	25
Figure 3.1: Project Planning	29
Figure 3.2: Experiment Tree of the Project	30
Figure 3.3: Flowchart of Transfer Learning and Fine Tuning.	31
Figure 3.4: Microcrack Defect Samples from ViTrox.	32

Figure 3.5: Configuration of proposed system33

Figure 4.1: Output of Mask GAN

35

Figure 4.2: Output of Defect Image GAN	36
Figure 4.3: Confusion Matrix of Classifier trained on Rotate and Flip Augmented Data	37
Figure 4.4: Images that produce False Negative results in Classifier trained on Rotate	and
Flip Augmented Data	37
Figure 4.5: Confusion Matrix of Classifier trained on GAN-Augmented Data	38
Figure 4.6: McNemar's Test Matrix of Classifier trained on GAN-Augmented Data	and
Rotate and Flip Augmented Data	39
Figure 4.7: Loss Curve of Classifier trained on data augmented with both methods	40



LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
DCGAN	Deep Convolutional GAN
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network



CHAPTER 1 INTRODUCTION

1.1 Project Background

This project is performed in collaboration with ViTrox Technologies Sdn. Bhd. ViTrox specializes in developing automated vision inspection system. Machine learning algorithms applied in their machine vision solutions aids their customers to ensure quality in their customer's production.

The presence of microcracks is one of the defects in silicon wafers. Microcrack refers to cracks with a thickness that is on the scale of micrometers. This type of defect is hard to be seen through the naked eye. Hence, computer vision-based solutions are the go-to in detecting microcracks in silicon wafers.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Image-based defect detection is a challenging problem requiring high accuracy algorithms and is often tackled using machine learning techniques. Deep convolutional neural networks are ubiquitous in image classification problems as they can produce results with high accuracy. In addition, convolutional neural networks effectively extract image features while numerous layers in deep neural networks greatly improve network performance by efficiently leveraging large amounts of training data (Krizhevsky et al., 2017).

Data augmentation is the process of manipulating training data to achieve better generalization. Common techniques in image data augmentation are "handcrafted", which include geometric transformations and noise filters (Shorten & Khoshgoftaar, 2019).

GAN is a framework of training two or more neural networks in a contest against each other to generate data that follows the distribution of the training set (Goodfellow et al., 2014). As seen in Figure 1.1, GAN consists of a generator network and a discriminator network. A generator network takes in random noise as input and produces images that mimic the training data. Meanwhile, a discriminator network is trained to distinguish between real and generated data. Iterations of training will result in both networks being more competent in their tasks, eventually producing generated data that closely resemble the training data. Hence, it is easy to see the implication of applying GAN in image data augmentation (Su, 2021).



Figure 1.1: Architecture of a Typical GAN from (An Intuitive Introduction to Generative Adversarial Networks (GANs), 2018)

1.2 Problem Statement

ملال

aun

In defect classification problems, non-defective samples are usually more abundant than defective samples. This phenomenon is also known as the class imbalance problem. As a result, the distribution of the minority class will be misrepresented, and the number of false negatives will increase (Buda et al., 2018). As seen in Figure 1.2, classifiers trained on imbalanced data tend to misclassify the minority class.



Figure 1.2: Visualization of the Class Imbalance Problem in Classification (Kushwaha, 2019).

au 9

Furthermore, misclassification of defective products as being non-defective is critically detrimental in defect detection applications. Data augmentation can be used to balance the data distribution while increasing data size. While the traditional "handcrafted" data augmentation technique is viable, it is specific to the image source and has to be manually tuned for each different source of the image. Hence, GAN is proposed as a generalizable solution to augment the minority samples and balance the data distribution.

1.3 Objectives

The project objectives are as follows:

- (a.) To produce a generalizable image data augmentation system for defect detection problems using GAN.
- (b.)To improve the accuracy of neural network-based defect detection algorithm by data augmentation using GAN.
- (c.) To analyze the impact of GAN-based data augmentation and traditional data augmentation on the performance of classification networks.



- a. The proposed system will be evaluated on an image-based defect detection problem.
- b. Transfer learning will be used in training the classification network and GAN to avoid training the networks from scratch.
- c. Image under different conditions will be used to evaluate the generalizability of the proposed system.
- d. Traditional data augmentation techniques, including noise injection and geometric transformation, will be used as a benchmark for comparison in evaluating the performance of GAN-based data augmentation.

e. The number of misclassified defective images by the classifier trained with augmented data will be used as a metric for the performance of the GAN.

1.5 Project Rationale

The project rationale are as follows:

- a. Data with imbalance class problem tend to affect machine learning model performance. This project aims to use data augmentation to overcome class imbalance by artificially generating samples of the minority class.
- b. Conventional data augmentation methods are chosen based on the characteristics of the input samples. GAN is proposed to automate the data augmentation step by learning image characteristics and generating artificial samples that are similar to input data.

1.6 Project Methodology

The project consists of using GAN as an alternate method of data augmentation. The role of GAN within the overall machine learning framework is shown in Figure 1.3.



Figure 1.3: Overview of Project

Furthermore, the overall methodology of the project is summarized in the flowchart in Figure 1.4.



Figure 1.4: Flow Chart of the Proposed Project

After a preliminary background study is performed, an experiment will be conducted to evaluate the proposed system. First, image data collected and preprocessing such as resizing of the image will be done. The data will then be split into training and testing sets. Further, the training data will be be used to train the GAN-based data augmentation system. Afterwards, two classifiers will be trained, with traditionally augmented data and GANaugmented data. Finally, the rate of false negatives by these two classifiers will be compared to show the viability of GAN-based data augmentation.

1.7 Thesis Organization

This thesis is organized according to the 5 main chapters. Chapter 1 describes the project background, problem statement, objectives, scope, and rationale. Furthermore, literature on GAN and machine learning frameworks are reviewed in Chapter 2. In addition, project planning and experiment procedure is detailed in Chapter 3. Moreover, Chapter 4 analyzes results obtained from the experiments. Finally, the project is concluded in Chapter 5.



CHAPTER 2

LITERATURE REVIEW

2.1 Image Classification Problem

Image classification is a class of machine learning problem that involves taking in an image as input and determining the class that the image belongs to (Russakovsky et al., 2015). For example, insect image classification involves determining the type of insect in each image within the dataset, as shown in Figure 2.1.



Figure 2.1: Samples from insect classification dataset (Wu et al., 2019).

2.1.1 Image Classification Pipeline

Luo et al. (2020) stated that image classification can be divided into the following steps:

- 1. Image capture
- 2. Image preprocessing
- 3. Feature extraction
- 4. Feature selection
- 5. Feature classification

AALAYS/A

Image capture : Image acquisition system including camera and lighting system.

Image Preprocessing : Image enhancement such as denoising and segmentation.

Feature Extraction : Features such as lines and edges are extracted from the image.

Feature Selection : Analysis is performed to choose features that represent labels to be classified.

Feature Classification : Chosen features are used to determine the label of input images.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.1.2 Machine Learning-based Classification

Feature extraction and feature classification steps shown in the previous section may consist of either hand-crafted or machine learning-based algorithms. Machine learning can be broadly divided to supervised and unsupervised learning. Referring to Figure 2.2, supervised learning involves extracting relationship information between a set of training data and labels. This means that both feature extraction and feature classification is fully automated through the machine learning algorithm.



2.2 Deep Neural Networks

According to (Goodfellow et al., 2016), deep neural networks are end-to-end machine learning algorithms. Neural network parameters are optimized with training data through a loss function in order to maximize its accuracy.

2.2.1 Network Architecture

Referring to Figure 2.3, Neural Networks are feedforward computational graphs consisting of multiple layers. Each layer in the graph has a set of weights and biases that are altered through the optimization process.



Figure 2.3: Example of Simple Feedforward Neural Network (Goodfellow et al., 2016)

On the left, the architecture of neural network is drawn in detail of each of the neuron unit. Each arrow represents the multiplication operation with a weight while each node represents the summation of the value with a bias. On the right, each layer is represented with a node. This representation is used for larger neural networks where it would be too space consuming to draw every single unit.

2.2.2 Activation Functions

Activation functions are functions that introduce non linearity to the deep neural networks. At the end of each layer, the feature variables are passed through activation functions. Examples of activation functions reviewed by Zaheer & Shaziya (2018) is as shown in Table 2.1.

	1	-	1	
Function	Sigmoid	Tanh	ReLu	ELU
Function	v h	y 1	v 1	y 1
Snape				
Output Range	0 to 1	-1 to 1	0 to ∞	-∞to∞
	2	0		

Table 2.1: Example of Four Common Activation Functions

2.2.3 Loss Function

Loss functions is a representation of how well the Deep Neural Network is performing. In the training process, this function is minimized by tuning the network parameters. For classification problems, probabilistic loss functions are used. For example, Ruby & Yendapalli (2020) applied the Binary Cross Entropy loss function to classify whether the input is a flower and achieved 95.63% accuracy in their experiment.

2.2.4 Backpropagation and Optimization

Non linearity introduced by activation functions pose a challenge in computing derivatives, which is solved through backpropagation. The derivatives are computed layerby-layer, from output to input, using a chain rule.

In general, the optimization algorithm for Deep Neural Network training is based on gradient descent methods. Gradient descent methods make use of derivatives in a function in order to compute the parameters that minimizes it. Specifically, weights and biases are iteratively calculated using the negative of the derivative in order to step towards a minimum.

Huda et al. (2019) compared two optimizers, Adam and Stochastic Gradient Descent (SGD) in their work. It is found that Adam optimizer converges more quickly and reaches a lower loss value. However, it is prone to overfitting as the model achieves a lower training loss but did not perform better on validation data.

2.2.5 Regularization and Normalization

Regularization of network weights is a needed process to improve network generalizability. The weights in a neural network can be interpreted as degree of emphasis of a certain feature in determining the output. This means that without regularization, the neural network may overemphasize some features while overlooking others.

Batch normalization is one of the methods in network regularization. For each batch of training data, weights of a given layer are scaled to be zero mean and unit variance. Batch normalization of network weights ensure the difference of weights between layers are not too large. Hence, derivatives of loss function at each layer will not vary too much and overshooting of minimization steps are reduced. For example, Chen et al.(2018) applied batch normalization on their vehicle detection algorithm and achieved 1.9% improvement on mean average precision compared to the model trained without normalization.

2.3 Convolution Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a family of neural networks that utilizes convolution operation in computing layer outputs. The use of convolution operations allows better feature extraction which greatly improves performance of the neural network. For example, Vasan et al. (2020) applied convolutional neural network to classify malware by converting software binary into a 2D image.



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.3.1 Convolution Operation

Convolution in a 2D image is a signal processing operation in which the output at a certain pixel position is the weighted sum of surrounding pixel values of the input. This operation is ubiquitous in image feature extraction algorithms.

By applying convolution in deep learning, instead of a vector, layer outputs are treated as feature maps of the input image. The use of feature maps allows the neural network to retain positional information of each pixel in the image. Furthermore, the automatic optimization of convolution weights allows for hidden patterns in images to be discovered. Figure 2.5 illustrates how features can be filtered from an image by convolving a 3x3 kernel across it.



2.3.2 Common CNN used for Classification

Various state-of-the-art CNN architectures are used in deep learning research. These architectures are easily accessible through deep learning library such as Tensorflow and Pytorch. For example, Ghaderzadeh & Asadi (2021) have identified ResNet, DenseNet, MobileNet, VGG, and Inception among the most popular CNN models used in research of CNN-based COVID 19 diagnosis. Table 2.2 shows the common CNN models, with time per inference step and accuracy evaluated against ImageNet dataset (Martín Abadi et al., 2015).

Table 2.2: Common CNN Architectures used in Research

Architecture	Original Author	Parameters	Time (ms) per inference step	Accuracy
ResNet50	He et al. (2015)	25,636,712	4.55	0.921
DenseNet121	Huang et al. (2018)	8,062,504	5.38	0.923
MobileNetV2	Sandler et al. (2019)	3,538,984	3.83	0.901
VGG16	Simonyan &	138,357,544	4.16	0.901
	Zisserman (2015)			
InceptionV3	Szegedy et al. (2015)	23.851.784	6.86	0.937



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.4 Image Data Augmentation Methods

WALAYSIA

Data augmentation is a method to artificially increase training data in order to improve the performance of deep learning classifier. According to Shorten & Khoshgoftaar, (2019) conventional image data augmentation methods include geometric transformation and color space manipulation. Moreover, an emerging method of augmenting data is through the use of generative networks.

Taylor & Nitschke (2018) compared different data augmentation schemes and the results are shown in Table 2.3 below. Image Data Augmentation methods are not necessarily mutually exclusive. In fact, the combination of different Image Data Augmentation methods will likely provide improvement over using only an individual method (Shijie et al., 2017).

Table 2.3: Comparison of Data Augmentation Effect on CNN Performance (Taylor & Nitschke, 2018)

Method	Example Image	Description	Accuracy
Baseline		The image without augmentation.	64.50 ± 0.65%
Flipping		Image is flipped along its vertical axis.	67.36 ± 1.38%
Rotating		The image is rotated along its center.	69.41 ± 0.48%
Cropping		The image is cropped, resulting in many images of smaller resolution.	$79.10 \pm 0.80\%$
Color Jittering		Color properties such as Hue, Saturation and Brightness are altered.	$67.18 \pm 0.42\%$
Edge Enhancement		Sobel filter is used to detect edges, which are then intensified.	$66.49 \pm 0.84\%$
Fancy PCA		Random noise is added to Principal Components of an Image.	67.54 ± 1.01%

2.5 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) falls under generative models in machine learning. Generative models are approaches in statistical classification which aims to produce samples that mimic true data distribution. In order words, the distribution of training samples is modeled which allows for artificial samples to be generated.

2.5.1 GAN Framework

Referring to Figure 2.6, GAN consists of two neural networks, the generator and discriminator. The generator network in a GAN takes in random noise as input and produces a sample from its trained distribution. This generator network is trained under an adversarial training process along with a discriminator network (Goodfellow et al., 2014). The objective of training the discriminator network is to distinguish between samples generated from generator network and the actual training data. On the other hand, the generator network is trained to produce samples which are hard for the discriminator network to discern.



Figure 2.6: Overview of GAN Framework (Harada et al., 2019).

2.5.2 Deep Convolution GAN (DCGAN)

DCGAN is the application of GAN framework in generating images. Convolutional layers are used within the architecture of the generator network and discriminator network. In this case, convolutional layers provide learned feature that aid in representing images effectively (Radford et al., 2016).

2.5.3 Conditional GAN

Conditional GAN is a variation of GAN that allows generation of samples that are conditioned with the class information of the training data. Conditional GAN allows selective generation of samples. For example, the conditional GAN can be trained with data consisting of defective and non-defective samples, while selectively output only defective generated samples. This is achieved by modifying the input for both the generator and discriminator networks to include the class label information (Mirza & Osindero, 2014). Referring to Figure 2.7, conditional input y is supplied to discriminator and generator, in addition to the original input x (image to be discriminated) and input z (random variable).



Figure 2.7: Framework of Conditional GAN (Mirza & Osindero, 2014)

Multi-Scale Gradient GAN (MSG-GAN) (Karnewar & Wang, 2019) is a state-ofthe-art GAN framework that incorporates various existing implementations in addition to novel techniques. The resulting GAN is capable of generating high resolution output with good fidelity.



Progressive synthesis network are shown to improve training speed and output quality (Karras et al., 2018). Moreover, MSG-GAN incorporates progressive network in the generator due to the capability to control image features at different level of resolutions.

The main idea of MSG-GAN is to control fluctuations in discriminator loss during training. As the loss function is applied at different level of resolutions, GAN training will be more stable and risk of training failure such as modal collapse is reduced.

2.5.5 Sequential Image Generation using GAN

Sequential Image Generation GAN (Turkoglu et al., 2019) is a GAN architecture that enables foreground images to be "painted" onto background image sequentially. As shown in Figure 2.9, numerous foreground objects can be created on a background image by passing the image iteratively through the GAN.



2.6 Deep Learning Methodology

2.6.1 Transfer Learning

Transfer learning is the adaptation of model trained on large dataset onto a new, smaller dataset (Weiss et al., 2016). This is achieved by using the network parameters of the trained model to train on a new dataset. The benefit of using pre-trained model is faster convergence in training and less data is required to train the network for good performance. Moreover, transfer learning not only applies to CNN but it can also be used in training GAN (Wang et al., 2018).

2.6.2 Model Validation

Holdout test set refers to a subset of input data that is not used in training (Pal & Patel, 2020). This set of data is withheld and used to evaluate the model at the end of the model tuning process. Holdout test set ensures the validity of the experiment as the model is not altered anymore after evaluation on holdout test.

20

Validation is performed at each training step of a deep learning model in order to assess its performance. Cross validation is a method to obtain an accurate estimate of a model performance (Berrar, 2018). Particularly, k-fold cross validation is a cross validation method with low bias.

In k-fold cross validation, data is separated into k number of sets. At each validation step, one set is withheld while the rest is used to train the model. The performance of the model is then evaluated on the withheld data. This is repeated k times, using different sets as the validation data. After the model is trained and evaluated on all k validation sets, the performance value is averaged out to obtain the estimate of true model performance.



Figure 2.10: Illustration of 10-fold Cross Validation (Berrar, 2018).

2.6.3 Model Selection

Various statistical methods are available in the process of machine learning model selection and comparison. Model comparison revolves around statistical testing for difference in proportion. Particularly, McNemar's Test is typically used for comparing two models (Raschka, 2020). When many machine learning models are involved, multiple hypotheses testing such as F-test is required.

2.6.4 Performance Metric

In binary classification problem, performance of the model revolves around the ratio of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) (Flach & Kull, 2015). These metrics are summarized in Table 2.4 below.

Metric	Formula	Description
Accuracy	TP + TN	Used to show the likelihood of the model
	$\overline{TP + FP + TN + FN}$	producing the correct output.
Precision	ТР	Used to show the performance of the model
	$\overline{TP + FP}$	when False Positives are concerned.
Recall	TP	Used to show the performance of the model
	$\overline{TP + FN}$	when False Negatives are concerned.
F1 Score	$Precision \times Recall$	Harmonic mean between precision and recall.
	$2 \times \frac{1}{Precision + Recall}$	Shows the weighted average of precision and
	AVSIN	recall.

Table 2.4: Performance Metric of Classification Model

On the other hand, two popular performance metrics for GANs are Fréchet Inception Distance (FID) and Inception Score (IS) (Chong & Forsyth, 2020). GAN performance metrics are summarized in Table 2.5.

Table 2.5: Performance Metric of GAN Formula Description Metric Fréchet Represents difference of training and generated $||M_t + \operatorname{Tr} \left(C_t + C_g \right)$ Inception data, along with difference in measure of Distance dispersion in these data. $-2(C_tC_g)^{\frac{1}{2}}$ Generated images that more closely resemble training data numerically will produce a lower Where Fréchet Inception Distance, which is more M is mean matrix, desirable. C is covariance matrix. t is training data, and g is generated data Represents variety in the images generated and Inception Score $\exp \left| \mathbb{E}_{z \sim p(z)} \left[\mathbb{D} \left(p(y|g(z)) || p(y) \right) \right] \right|$ how distinctive each image is. Assesses GAN based on probability distributions of a classifier Where output. y is a label, p(y|x) is probability of the label to be Generated images that are more easily computed on image x using distinguished and classified will result in a Inceptionv3 model, higher score, which is more desirable. p(y) is the marginal class distribution, and $\mathbb{D}(p || q)$ is the Kullback Leibler divergence between probability p and q,

24

2.6.5 Software

Python 3 is a high level programming language that is and commonly used in machine learning research (Raschka, 2015). Python is dynamically typed and supports multiple programming paradigm including object-oriented programming.

In addition, Google Colaboratory or Google Colab is a cloud-based machine learning research platform using Python (Bisong, 2019). Colab enables deep learning model to be trained on the cloud and is easily accessible to researchers. The symbols for Python and Google Colab are shown in Figure 2.11.



Besides, deep learning libraries are used to enable tensor computations required for deep learning. Figure 2.12 shows two of the most popular deep learning libraries which are TensorFlow (Martín Abadi et al., 2015) and Pytorch (Paszke et al., 2019). Both of these libraries include functions to specify network architecture and contain optimizer algorithms to perform training. There is little difference between Tensorflow and Pytorch and the choice of library is mostly depending on the researcher.



Figure 2.12: Symbol of Pytorch (left) and Tensorflow (right) (PyTorch vs TensorFlow, 2021).

2.7 Related Works

Title	Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection	Defect Image Sample Generation with GAN for Improving Defect Recognition	Defect Enhancement Generative Adversarial Network for Enlarging Data Set of Microcrack Defect
Author	Zhang et al. (2021)	Niu et al. (2020)	Lin et al. (2019)
Use Case	Applied on CODEBRIM dataset which consists of surface defects in concrete bridges.	Applied on commutator cylinder surface defect dataset.	Applied on captured images of magnetic ring that contain microcrack defects.
Classifier	ResNet34 DenseNet121	VGG16	Lenet5 Alexnet SVM
GAN	 Novel Defect-GAN introduced in the paper Generator networks consist of two autoencoders The two autoencoders mimic defacement and restoration process 	 Novel SDGAN introduced in the paper D2 Adversarial loss and Cycle consistency loss is proposed One generator network produces defect image output while the other produces a defect free output 	 Novel DEGAN introduced in the paper Based on DCGAN using convolutional layers Applies discriminator model from BEGAN which contains an autoencoder structure Introduces a novel reconstruction error as loss function
Compared Against	StackGAN++ StyleGAN v2 StarGAN	CNN without data augmentation WGAN D2GAN CycleGAN	Comparison is made between the classifiers trained on DEGAN augmented data and classifiers trained only on original training data.
Performance metrics	Accuracy Fréchet Inception Distance	Error rate Fréchet Inception Distance Training time	Accuracy Sensitivity Specificity
Findings	• The proposed Defect-GAN using defacement and restoration process generates data that can train a more accurate classifier	 CNN trained on GAN augmented data performs better than CNN trained on only original training data. Models trained on GAN augmented data is more robust to 'dirty' data outliers The proposed SDGAN generates high quality data that can capture variations in lighting condition 	 The proposed DEGAN is capable of expanding training data and produce better classifier performance Classifier trained on GAN augmented data produces better sensitivity and specificity characteristics

Table 2.6: Summary of Related Works

CHAPTER 3 METHODOLOGY

3.1 **Project Overview**

Figure 3.1 shows the steps that are conducted throughout the project. Background study is conducted on theory related to the project. In addition, project planning is performed through the use of flow chart and Gantt chart. Next, textbooks and journals relating to architecture of classifier and GAN neural networks are studied. Furthermore, similar works to the project are searched and compared.

Following literature search, project methodology including experiment design is completed. Data collection is performed by referring to open-source surface defect data in addition to data from collaboration company. Image data is preprocessed through resizing and cropping to the appropriate resolution and is split into train and test sets.

Train data set is used for training GANs and classifier networks while test data set is used to evaluate the classifier performance. The GANs are trained until the outputs are visually similar to training data. The trained GANs are then be used to augment the training data. Meanwhile, the training data is also separately augmented using conventional methods. The GAN augmented data and conventionally augmented data are used to separately train the CNN classifier. With both the GAN and conventionally augmented data, the CNN classifier is trained until the loss function converges and does not improve further. The performance of the CNN classifier on the test data set is recorded. Further, McNemar's Test is used to compare the performance of CNN classifier trained on GAN and conventionally augmented data.

One possible error that might occur is improper hyperparameter of GAN. If classifier performance on GAN generated data is significantly worse than conventional data augmentation, GAN hyperparameters such as input resolution will be tuned. In addition, "dirty" data such as inconsistent input sample characteristics will also affect performance. If classifier performance is poor, the problematic data set will be inspected and cleaned before running the experiment on it again.





Figure 3.1: Project Planning

3.2 Design of Experiment

The experiment tree of the project is shown in Figure 3.2 below. One variable is varied which is augmentation technique. GAN-based data augmentation is compared to conventional data augmentation to determine the viability of the proposed system. True Positive, True Negative, False Positive, and False Negative (shortened as TP, TN, FP and FN) are used to evaluate the performance of model trained on augmented image data.



3.3 **Project Software and Tools**

3.3.1 Programming Language, Libraries and Platform

Python is used as the programming language for the project. Python is popular in machine learning research and the CNN and GAN source code used is programmed in this language. Furthermore, Tensorflow library is used as the deep learning library. Additionally, supporting library including NumPy for matrix data manipulation, and Python Image Library and OpenCV for Image manipulation are used.

3.3.2 Transfer Learning

Transfer Learning and fine tuning are used to speed up training process of the CNN classifier. The transfer learning procedure is shown in Figure 3.3 below. First, the weights of a pre-trained model is imported and frozen meaning it will not be tuned in the training processed. Next, new layers are appended on the base model and the model is trained until convergence. Afterwards, the base model is unfreezed and trained again.



Figure 3.3: Flowchart of Transfer Learning and Fine Tuning.

3.3.3 Data Source

The project is applied on microcrack defect detection. The proposed system is tested on microcrack defect dataset from ViTrox Technologies Sdn. Bhd., the collaborating company. Samples from the defect data used in the experiment are as Figure 3.4 below. Note that only line shaped defects are chosen as train set. This is done to illustrate whether the proposed system can generalize defect shapes into those from the test set.



3.3.4 Neural Network Configuration

The proposed system, as shown in Figure 3.5, consists of two GANs that takes in bounding box input vector and non-defect input image to generate defect images. Mask GAN is trained on defect mask labels and generates defect mask from bounding box information. On the other hand, Defect Image GAN is trained on mask-image pair to generate defect images from defect mask and non-defect images. The architecture used Mask GAN is MSG-GAN (Karnewar & Wang, 2019) while Defect Image GAN uses Sequential Image GAN (Turkoglu et al., 2019). Furthermore, the classifier architecture used is ResNet (He et al., 2015).



UNIVER Figure 3.5: Configuration of proposed system

3.3.5 Conventional Data Augmentation Methods

Conventional data augmentation methods used are shown in Table 3.1 below.

Data Augmentation Method	Parameters
Rotation	0°, 90°
Flipping	Horizontal, Vertical

Table 3.1: Conventional Data Augmentation Methods Used.

3.3.6 Performance Evaluation Metric

Performance Evaluation Metric for the project are listed in Table 3.2 below. True Positive, True Negative, False Positive, and False Negative metrics are summarized into a confusion matrix.

EKN1	Table 3.2: Performance Metrics Used
Metric	Justification
True Positive	Number of correctly predicted defect images
True Negative	Number of correctly predicted nondefect images
False Positive	Number of incorrectly predicted nondefect images
False Negative	Number of incorrectly predicted defect images
Visual Inspection	Qualitative inspection of GAN output

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 GAN Image Generation

4.1.1 Mask GAN Output

The output of Mask GAN is shown in Figure 4.1 below. Each row represents images that are generated with the same bounding box input. Significant variation between each image can be seen which means that the GAN can generate defects of varying shape within the same bounding box.



Figure 4.1: Output of Mask GAN

4.1.2 Defect GAN Output

The output of Defect Image GAN is shown in Figure 4.2Figure 4.1 below. The first column is the actual images from the dataset and the following columns are images generated with different latent variables. In each row, the same defect mask is provided to the Defect Image GAN. There are no observable variation in the output images and the defects are not textured. Hence, Defect Image GAN can only produce defect images by painting a single tone at specified image mask.



Figure 4.2: Output of Defect Image GAN

4.2 Classifier Performance Analysis

4.2.1 Rotate and Flip Augmented Data

The result of classifier trained on rotate and flip augmented data is shown in Figure 4.3 below. The number of correct predictions is the sum of diagonal cell values, which is 14. 6 defect images are predicted as pass, meaning that there are 6 false negatives that are produced by the classifier.



Figure 4.3: Confusion Matrix of Classifier trained on Rotate and Flip Augmented Data

The poor performance of classifier trained on rotate and flip augmented data is explained by inspecting images which produces incorrect predictions. As shown in Figure 4.4, the false negatives are produced by images that are disimmilar to the training data. Rotate and flip data augmentation is incapable of producing defect shapes that are different from the original data.



Figure 4.4: Images that produce False Negative results in Classifier trained on Rotate and Flip Augmented Data

4.2.2 GAN-Augmented Data

The result of classifier trained on rotate and flip augmented data is shown in Figure 4.5 below. The number of correct predictions is the sum of diagonal cell values, which is 20. In this case, the classifier trained on GAN augmented data achieves an accuracy of 100%. The 100% accuracy does not actually reflect real-life performance. Instead, the exceptionally good performance is due to the small quantity of data available for evaluation. Nevertheless, GAN augmented data is shown to minimize false negatives, provided that bounding box information that reflect real world data is given to the GAN-based data augmentation system.



4.2.3 McNemar's Test

McNemar's Test is used to evaluate the significance of difference between performance in classifier trained on rotate and flip augmented data and classifier trained GAN augmented data. The correct and incorrect predictions of each model is formulated into a matrix as shown in Figure 4.6 below.



Figure 4.6: McNemar's Test Matrix of Classifier trained on GAN-Augmented Data and Rotate and Flip Augmented Data

A null hypothesis is formed, stating that the probability of occurrence in the offdiagonal cells are the same.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

 $H_0: \pi = 0.5$

 $H_1: \pi \neq 0.5$

Significance level, α: 0.05 (95% confidence)

p-value: 0.03125

At a confidence level of 95%, p-value is smaller than significance level of 0.05. Therefore, the null hypothesis is rejected. There is a significant difference between GANbased data augmentation and Rotate and flip data augmentation.

4.2.4 Loss Curve

Further analysis of classifier performance between each data augmentation method can be analyzed by inspecting the loss curve, as shown in Figure 4.7. Classifier trained on rotate and flip augmented data produces a smoother loss curve than classifier trained on GAN augmented data. The smooth loss curve is produced when there is little variation among the training data. Conversely, GAN augmented data is varied and therefore the classifier requires more iterations to converge. Hence, the loss curve illustrates that GAN-based data augmentation increases variation in data. The significant increase in performance of classifier proves that meaningful variation of data is produced by GAN-based data augmentation.



Figure 4.7: Loss Curve of Classifier trained on data augmented with both methods

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

In this project, a data augmentation system using GAN is developed. The GAN-based image data augmentation system takes in bounding boxes information and produces images with defects on the specified locations. The developed system is compared with conventional data augmentation method, specifically random rotation and flipping, by training a classifier on the augmented data. The classifiers trained on both data augmentation methods are evaluated using confusion matrix and compared using McNemar's Test. As a result, classifier trained on GAN-augmented data is shown to be better than classifier trained on rotate and flip augmented data at a confidence level of 95%. In short, the developed GAN-based data augmentation system is capable of creating user-customized defect dataset which improves the performance of classifier trained on the data.

5.2 Recommendations | TEKNIKAL MALAYSIA MELAKA

As shown in Chapter 4, images generated from the defect image GAN has little variation in texture within defects. One suggestion to improve the system is to tune this GAN so that different textures of defects can be produced. Introducing variation in defect texture will lead to more realistic generated images that reflect actual images occurring within production lines.

ونيۆم سيتى تيكنيكل مليسيا ملا

In addition, GAN based data augmentation system can be configured to support multiple defect types. Both the Mask GAN and Defect Image GAN can be conditioned on defect class labels and produce different type of defects when different label inputs are given. This will remove the limitation of defect type for the system, and reduce the training time by avoiding having to train the whole system separately for each type of defect needed.

5.3 Sustainability of Project

The GAN-based data augmentation system aims to improve classifiers performance in detecting defect. By detecting defect early within the production line, steps can be taken to isolate and fine tune problematic areas in the production system. This reduces the chances of defect occurring and reduces the material and labor cost of reworking. Hence, the completion of this project is another step towards improving quality of production line and reducing wastes.

5.4 **Project Complexity**

5.5

This project involves complex engineering knowledge including statistical methods. Moreover, conflicting requirements such as balancing training time of GAN and its performance are required. In addition, the application of GAN in data augmentation is novel and explored with experiments in this project. Furthermore, this project has real life implications in improving defect detection in a production environment.

اونيوم سيتي تيڪنيڪل مليسيا ملاك Basic Entrepreneurship_{KNIKAL} MALAYSIA MELAKA

The GAN-based data augmentation system is produced under collaboration with ViTrox Corporation Berhad. The aim of this project is to improve current defect detection algorithms in the company. The application of GAN-based data augmentation will improve defect detection and lead to increased customer satisfaction and product quality.

REFERENCES

An intuitive introduction to Generative Adversarial Networks (GANs). (2018, January 7). FreeCodeCamp.Org. https://www.freecodecamp.org/news/an-intuitiveintroduction-to-generative-adversarial-networks-gans-7a2264a81394/

Berrar, D. (2018). Cross-Validation. https://doi.org/10.1016/B978-0-12-809633-8.20349-X

- Bisong, E. (2019). Google Colaboratory. In E. Bisong (Ed.), Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners (pp. 59–64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7
- Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks: The Official Journal of the International Neural Network Society*, 106, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011
- Chen, L., Ye, F., Ruan, Y., Fan, H., & Chen, Q. (2018). An algorithm for highway vehicle detection based on convolutional neural network. *EURASIP Journal on Image and Video Processing*, 2018(1), 109. https://doi.org/10.1186/s13640-018-0350-2
- Chong, M. J., & Forsyth, D. (2020). Effectively Unbiased FID and Inception Score and Where to Find Them. 6070–6079. https://openaccess.thecvf.com/content_CVPR_2020/html/Chong_Effectively_Unbi ased_FID_and_Inception_Score_and_Where_to_Find_CVPR_2020_paper.html
- Flach, P. A., & Kull, M. (2015). Precision-Recall-Gain Curves: PR Analysis Done Right. *NIPS*, 15.

Ghaderzadeh, M., & Asadi, F. (2021). Deep Learning in the Detection and Diagnosis of COVID-19 Using Radiology Modalities: A Systematic Review. *Journal of Healthcare Engineering*, 2021, e6677314. https://doi.org/10.1155/2021/6677314

Glassner, A. S. (2021). Deep Learning: A Visual Approach. No Starch Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville,
A., & Bengio, Y. (2014). Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27. https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf
3-Abstract.html

- Harada, S., Hayashi, H., & Uchida, S. (2019). Biosignal Generation and Latent Variable Analysis with Recurrent Generative Adversarial Networks.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. ArXiv:1512.03385 [Cs]. http://arxiv.org/abs/1512.03385
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2018). Densely Connected Convolutional Networks. ArXiv:1608.06993 [Cs]. http://arxiv.org/abs/1608.06993
- Huda, N. S., Mubarok, M. S., & Adiwijaya. (2019). A Multi-label Classification on Topics of Quranic Verses (English Translation) Using Backpropagation Neural Network with Stochastic Gradient Descent and Adam Optimizer. 2019 7th International Conference on Information and Communication Technology (ICoICT), 1–5. https://doi.org/10.1109/ICoICT.2019.8835362
- Karnewar, A., & Wang, O. (2019). *MSG-GAN: Multi-Scale Gradients for Generative* Adversarial Networks. https://doi.org/10.48550/ARXIV.1903.06048

- Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. ArXiv:1710.10196 [Cs, Stat]. http://arxiv.org/abs/1710.10196
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. https://doi.org/10.1145/3065386
- kushwaha, abhishek. (2019, August 24). Solving Class imbalance problem in CNN. AI
 Graduate. https://medium.com/x8-the-ai-community/solving-class-imbalance problem-in-cnn-9c7a5231c478
- Lin, S., He, Z., & Sun, L. (2019). Defect Enhancement Generative Adversarial Network for Enlarging Data Set of Microcrack Defect. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2946062
- Luo, Q., Fang, X., Su, J., Zhou, J., Zhou, B., Yang, C., Liu, L., Gui, W., & Tian, L. (2020).
 Automated Visual Defect Classification for Flat Steel Surface: A Survey. *IEEE Transactions on Instrumentation and Measurement*.
 https://doi.org/10.1109/TIM.2020.3030167
- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/
- Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. *ArXiv:1411.1784 [Cs, Stat]*. http://arxiv.org/abs/1411.1784

- Niu, S., Li, B., Wang, X., & Lin, H. (2020). Defect Image Sample Generation With GAN for Improving Defect Recognition. *IEEE Transactions on Automation Science and Engineering*, PP, 1–12. https://doi.org/10.1109/TASE.2020.2967415
- Pal, K., & Patel, Biraj. V. (2020). Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 83–87. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00016
- Parmar, M. (2021, March 30). Structure your code better in Google Colab with Text and Code Cells. *Medium*. https://miteshparmar1.medium.com/structure-your-codebetter-in-google-colab-with-text-and-code-cells-b6fa73feec20
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019).
 PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 32 (pp. 8024–8035).
 Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
- PyTorch vs TensorFlow: Comparing deep learning frameworks. (2021, April 22). Blog | Imaginary Cloud. https://www.imaginarycloud.com/blog/pytorch-vs-tensorflow/
- Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ArXiv:1511.06434 [Cs]. http://arxiv.org/abs/1511.06434

Raschka, S. (2015). Python Machine Learning. Packt Publishing Ltd.

- Raschka, S. (2020). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning* (arXiv:1811.12808). arXiv. https://doi.org/10.48550/arXiv.1811.12808
- Ruby, U., & Yendapalli, V. (2020). Binary cross entropy with deep learning technique for Image classification. *International Journal of Advanced Trends in Computer Science* and Engineering, 9. https://doi.org/10.30534/ijatcse/2020/175942020
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A.,
 Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale
 Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3),
 211–252. https://doi.org/10.1007/s11263-015-0816-y
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019). MobileNetV2: Inverted Residuals and Linear Bottlenecks. ArXiv:1801.04381 [Cs]. http://arxiv.org/abs/1801.04381
- Shijie, J., Ping, W., Peiyi, J., & Siping, H. (2017). Research on data augmentation for image classification based on convolution neural networks. 2017 Chinese Automation Congress (CAC), 4165–4170. https://doi.org/10.1109/CAC.2017.8243510
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. https://doi.org/10.1186/s40537-019-0197-0
- Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv:1409.1556 [Cs]. http://arxiv.org/abs/1409.1556
- Su, X. (2021). A Survey on Data Augmentation Methods Based on GAN in Computer Vision. Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery. https://doi.org/10.1007/978-3-030-70665-4_92
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. ArXiv:1512.00567 [Cs]. http://arxiv.org/abs/1512.00567

- Taylor, L., & Nitschke, G. (2018). Improving Deep Learning with Generic Data Augmentation. 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 1542–1547. https://doi.org/10.1109/SSCI.2018.8628742
- Turkoglu, M. O., Thong, W., Spreeuwers, L., & Kicanaoglu, B. (2019). A Layer-Based Sequential Framework for Scene Generation with GANs. https://doi.org/10.48550/ARXIV.1902.00671
- Vasan, D., Alazab, M., Wassan, S., Naeem, H., Safaei, B., & Zheng, Q. (2020). IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. *Computer Networks*, 171, 107138. https://doi.org/10.1016/j.comnet.2020.107138
- Wang, Y., Wu, C., Herranz, L., Weijer, J. van de, Gonzalez-Garcia, A., & Raducanu, B. (2018). Transferring GANs: Generating images from limited data. ECCV. https://doi.org/10.1007/978-3-030-01231-1_14
- Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(1), 9. https://doi.org/10.1186/s40537-016-0043-6
- Wu, X., Zhan, C., Lai, Y.-K., Cheng, M.-M., & Yang, J. (2019). IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8779–8788. https://doi.org/10.1109/CVPR.2019.00899
- Zaheer, R., & Shaziya, H. (2018). GPU-based empirical evaluation of activation functions in convolutional neural networks. 2018 2nd International Conference on Inventive Systems and Control (ICISC), 769–773. https://doi.org/10.1109/ICISC.2018.8398903
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). *Dive into Deep Learning* (release 0.16.1). https://d2l.ai

Zhang, G., Cui, K., Hung, T.-Y., & Lu, S. (2021). Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). https://doi.org/10.1109/WACV48630.2021.00257



APPENDICES

Gantt Chart of Proposed Project

 .		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~																					 	-											
Thesis			1	<u>.</u>											Wonths (weeks 1-4)																				
Tasks	Oct-21			ct-21		Nov-21				De	c-21	:-21		Jai	n-22			Fe	b-22	.2		Mar-	lar-22			Apr-22			May-22			Jun-22			2
Background Study		1										1						-				_													
Project Planning			1																			_	1												
Study Classifier Networks																																			
Study GAN Architectures		-									-																								
Study Similar Works			2													-																			
Define Experiment Design				67.											1		1				N.,														
Collect Data				6	1)																														
Finals and Semester Break																																			
Conduct Experiment			1			1							je.																						
Compile Project Report				1								16					6						_			\mathbf{r}_{i}									
Revise Project Report				X	Y.					1.000	1	-			1	1				~	(\cdot)	5		~10	0	2	~	7							
Project Presentation							1		100											1	1	10 m					10.0								
Submit Project Report		_																																	
		U	N		VI	EF	2.5	31	П	Т	E	K	NI	K	A	L	N	A	L	A)	(S	SIA		VIE	1		V	()					_		