DEEP LEARNING ALGORITHMS FOR BATIK DESIGN CLASSIFICATION

LEE XIAO XUAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEEP LEARNING ALGORITHMS FOR BATIK DESIGN CLASSIFICATION

LEE XIAO XUAN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Computer Engineering with Honours

Faculty of Electronic and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Deep Learning Algorithms for Batik Design Classification

Sesi Pengajian : 2023/2024

Saya LEE XIAO XUAN mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	SULIT*	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD*	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.
✓	TIDAK TERHAD	

Disahkan oleh:

(TANDATANGAN PENULIS) (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: 117, Lorong 5, Taman Sri Pinang, 34000 Taiping, Perak.

PROF MADYA DR NURULFAJAR BIN ABD MANAP
Profesor Madya
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK),
Universiti Teknikal Malaysia Melaka

Tarikh : 11 Januari 2024 Tarikh : 24 Januari 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled "Deep Learning Algorithms for Batik Design Classification" is the result of my own work except for quotes as cited in the references.

Signature	:	…………………………………
Author	:	…LEE XIAO XUAN…………
Date	:	…11 JANUARI 2024…………

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Computer Engineering with Honours.

Signature : ....

Supervisor Name : PROF MADYA DR NURULFAJAR BIN ABD MANAP

Date : 24 JANUARY 2024

DEDICATION

First and foremost, we would like to express our gratitude to everyone who was directly or indirectly involved in the preparation of this report. I would like to acknowledge my FYP supervisor, PM. Dr. Nurulfajar bin Abd. Manap who was willing to guide and motivate me to complete the project. I much appreciate his efforts in providing me with comprehensive guidance on how to accomplish the study. My fellow classmates also should be recognized for their support throughout the journey. My biggest thanks go to my family for their unlimited support in making me successfully complete this study.

ABSTRACT

Batik is widely known as one of the unique identifiers among Southeast Asia countries including Malaysia. Batik industry in Malaysia holds an important place in the craft-based industry. However, the batik industry in Malaysia lack of a comprehensive and standardized dataset which is crucial for developing an automated classification system. Besides, the traditional method of batik classification needs trained experts to visually inspect and analyse the intricate pattern. Therefore, this study indicates to compile and construct a new dataset of Malaysia batik for image classification. This study also aims to develop a batik classification system using deep learning algorithms that classify between Indonesia and Malaysia batik. A dataset with 1825 images including 949 images for Indonesia batik and 876 images for Malaysia batik is utilised. CNN models including MobileNet v2, YOLO-v8 and LeNet-5 is selected and the performance of these three CNN models is further investigated based on their accuracy, loss and confusion matrix. As a result, all three models reach a high accuracy which is 97.79% for MobileNet v2, 98.80% for YOLO-v8 and 92.94% for LeNet-5.

ABSTRAK

Batik dikenali sebagai salah satu ikon unik di kalangan negara-negara Asia Tenggara termasuk Malaysia. Industri Batik di Malaysia memegang kedudukan tinggi dalam industri kesenian. Namun, industri batik di Malaysia kekurangan set data yang komprehensif dan standard yang penting untuk membuat sistem pengelasan automatik. Selain itu, kaedah tradisional pengelasan batik memerlukan pakar terlatih untuk memeriksa corak batik yang rumit. Oleh itu, tujuan kajian ini adalah untuk menyusun dan membina set data baru Batik Malaysia untuk pengelasan imej. Kajian ini juga bertujuan untuk membuat sesuatu sistem pengelasan batik dengan menggunakan Deep Learning Algorithms yang mengelas antara batik Indonesia dan Malaysia. Set data dengan 1825 imej termasuk 949 imej untuk batik Indonesia dan 876 imej untuk batik Malaysia digunakan. Model CNN termasuk MobileNet v2, YOLO-v8 dan LeNet-5 dipilih dan prestasi ketiga-tiga model CNN ini dikaji lebih lanjut berdasarkan ketepatan, kerugian dan matriks kekeliruan. Hasilnya, ketiga-tiga model mencapai ketepatan yang tinggi iaitu 97.79% untuk MobileNet v2, 98.80% untuk YOLO-v8 dan 92.94% untuk LeNet-5.

ACKNOWLEDGEMENTS

First and foremost, we would like to express our gratitude to everyone who was directly or indirectly involved in the preparation of this report.

I would like to acknowledge my FYP supervisor, PM. Dr. Nurulfajar bin Abd. Manap who was willing to guide and motivate me to complete the project. I much appreciate his efforts in providing me with comprehensive guidance on how to accomplish the study.

My fellow classmates also should be recognized for their support throughout the journey. They have helped me with my study in different forms, which is very useful for me to complete this project.

My biggest thanks go to my family for their unlimited support in making me successfully complete this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CNN	:	Convolutional Neural Network
DNN	:	Deep Neural Network
IOU	:	Intersection over Union
ReLU	:	Rectified Linear Unit
YOLO	:	You Only Look Once

CHAPTER 1

INTRODUCTION

This chapter will briefly discuss the idea of the project such as problem statement, objectives, project significance, scope of project and the outline of the thesis.

1.1 Project Background

Batik is a traditional Malaysian and Indonesian textile art that applies wax and dye to create intricate patterns and designs. The patterns and design are mostly inspired by nature, mythology and daily life. There is a growing interest in batik for now as a sustainable and eco-friendly textile. However, the process of identifying and classifying the various styles and regions of Batik designs is a time-consuming and challenging task for humans. The traditional method of batik classification needs trained experts to visually inspect and analyse the intricate pattern. Besides, the existing automated batik classification system does not include dataset from Malaysia batik designs. Therefore, the project aims to develop a machine learning system that can accurately identify and classify batik designs which is helpful for preserving this art form that is important in cultural traditions of Malaysia. The project is mainly focusing on Malaysia and Indonesia batik. TensorFlow will be used to design the CNN architecture required in the development of the machine learning system. The performance will be further evaluated by applying suitable evaluation metrics. At the end of this project, an automated classification system for Malaysia batik is expected to be developed based on deep learning algorithms.

1.2 Problem Statement

Batik designs are a valuable cultural heritage that is required to be preserved and promoted. However, the identification and classification task of different Batik styles are time-consuming and tedious. The traditional method of batik classification needs trained experts to visually inspect and analyse the intricate pattern. It makes the scaling up of batik production, promoting batik – related industries such as tourism difficult.

Besides, there is a lack of an accurate and automated system that can helps to identify and classify different batik styles and regions. The existing Batik classification systems are available only for Indonesia batik. The limited dataset of Malaysia batik makes Malaysia batik can only rely on traditional method in batik classification tasks.

The lack of a comprehensive and standardized dataset of Malaysia Batik designs that span multiple styles had also make the training and evaluating of machine learning models for batik classification hard. Hence, the main purpose of this project is to develop a machine learning system that can automatically classified batik designs according to their origin country.

1.3 Research Objectives

The aim of this study is to develop a useful tool for individuals to differentiate Malaysia and Indonesia batik efficiently. Therefore, the following objectives have been established:

1. To compile and construct a new dataset of Malaysia batik for image classification.
2. To develop a batik classification system using deep learning algorithms that classify between Malaysia and Indonesia batik.

1.4 Scope of Research

The scope of research for the batik classification system using CNN and a dataset of Malaysia and Indonesia stamped batik designs includes the development of an automated classification system and the identification and classification of the designs using CNN algorithms such as MobileNet v2, LeNet-5, and YOLO-v8. The research aims to create a classification system that can accurately categorize different types of Malaysia and Indonesia batik designs based on their unique features and characteristics. However, a limitation of the research is the availability of the dataset of Malaysia batik designs, which may impact the accuracy and comprehensiveness of the classification system.

. In conclusion, the scope of research for this project is to develop a batik classification system using CNN to classify Malaysia and Indonesia stamped batik designs. The research will focus on creating an automated classification system and implementing CNN algorithms for accurate identification and classification of the designs.

1.5 Project Significant

The development of this project brings several significant benefits which includes helping art galleries and museums to classify and organize their batik collections based on regions, provide organized and accessible digital records about the regional variations of traditional batik designs. It helps in enhancing the educational value of batik collections and aids the development of educational materials such as textbooks and online courses in institutions.

Besides, by providing digital records of batik designs, this project contributes to preserving this cultural heritage for the future generations. Batik artisans are able to make use of the digital database of batik design elements, which allows them to learn from a wide range of sources to enhance their skills.

1.6 Report Structure

This report has been divided into two parts which is Final Year Project (FYP) 1 and FYP 2. Five main chapter of this report is structured as follows:

Chapter 1 describes the basic idea of the project including background of the project, problem statement, objectives, scope of research and the framework. The motivation of the project is explained.

Chapter 2 includes the literature review of relevant research. It consists of the basic introduction of various types of batik coming from different regions.

Chapter 3 describes the methodology of the project which includes the steps needed to complete the research. The methods to complete the project and to achieve the project objectives are clearly stated.

Chapter 4 includes the results and discussion obtained from the study. It mainly focusses on the data collected and evaluate the performance of each model used.

Chapter 5 consists of the conclusion of this study and includes some suggestions for future improvements for this study.

CHAPTER 2

BACKGROUND STUDY

2.1 Introduction

This chapter includes the literature summary on previous work in batik design classification system based on deep learning. This chapter consists of five sub-chapters which indicates the background study done in different aspect. Section 2.2 is about different types of batik design. Section 2.3 discuss about some batik classification methods used in previous research. Section 2.4 talks about deep learning algorithms while Section 2.5 review about the functionality concept of convolution neural network. Lastly, Section 2.6 review about the CNN classification method available.

2.2 Batik

Batik is a form of traditional textile art that involves applying wax to fabric and dyeing intricate patterns onto the fabric. This wax – resist technique has been found across the world including Indonesia, Malaysia, India and Egypt since the 12th century. In Malaysia, there are four main style of Batik design which are hand – drawn batik, stamped batik, stencilling batik and dip – dye batik. Batik artisans may combine different techniques in order to create unique designs [1].

The sample batik design from Malaysia and Indonesia is shown as below. The Malaysia batik is mostly designed using flora motifs while Indonesia batik has different types of motifs designed from various provinces.

Figure 2.1: Malaysia batik [1].

.

Figure 2.2: Indonesia batik [1].

In comparison with Indonesia batik which is mainly using brown, gold and black colours, Malaysia batik uses bright colours such as pink, purple and green. The Malaysia Batik industry is dominant by stamped batik and hand drawn batik.

2.2.1 Hand Drawn Batik

Hand drawn batik which is also known as Batik *Conteng* or canting batik is basically a free-style hand drawing by the Batik artisan. Hand drawn batik are commonly used to make formal wears. A metal pen tool called canting is used to apply liquid hot wax onto the fabric to create designs. The artisan will then use brushes to paint the dyes within the outlines done. The making process of hand drawn batik is as shown in Figure 2.3 and a sample of hand drawn batik is shown in Figure 2.4.

Figure 2.3: Making process of hand drawn batik [3]

Figure 2.4: Sample of hand drawn batik [32]

2.2.2 Stamped Batik

Stamped batik which is also known as block printed batik and batik *cop* is designed using blocks of wood or copper. For this project, stamped batik is used to represent all batik made using blocks.

Stamped batik is commonly tailored into shirts and dresses for daily wear. It does not have the intricate delicacy compared to hand drawn batik and similar patterns are repeated on a piece of fabric. The designed patterns are stamped onto cloths to create the designs [2]. The stamp is dipped in melted wax and pressed onto the fabric to create a repeating pattern. The fabric is then dip dyed and the wax will be removed. The making process of is as shown in Figure 2.5 and a sample of stamped batik is shown in Figure 2.6.

Figure 2.5: Making process of stamped batik [3]

Figure 2.6: Sample of stamped batik [32]

Stamped batik designs can be further categorised according to its motif. Motif is the main element in stamped batik making. It is referred as a feature used as the base of a batik design which will be repeated to generate a pattern [4]. The most common motif which can be found in Malaysia stamped batik designs are flora, fauna and geometry.

2.3 Batik Classification Method

2.3.1 Introduction

Batik classification had been done by many researchers. They utilised various machine learning methods such as Convolutional Neural Network (CNN) that is called VGG-16 and VGG-19 [5], CNN with Transfer Learning Method [6], CNN with ResNet-18 architecture [7] and Deep Neural Network (DNN) with batch normalisation [8].

2.3.2 Comparison of Past Research

Table 2.1: Batik Classification Performance Comparison

Reference	Approach	Dataset	Performance	Advantages	Disadvantages
I. M. A. Agastya and A. Setyanto [5]	CNN with VGG-19	900 images with 5 classes	89.3%	Using pre-trained CNN to reduce need for manual feature engineering and able to capture complex features from images.	Require significant computational resources and training time
F. A. Putra et al. [6]	CNN with Transfer Learning Method	566 images	96.91%	Reduced risk of overfitting as the model's architecture had already optimized.	Performance easily affected by factors such as lighting conditions and image quality but contain risk of pre-trained model is not relevant to the specific characteristics of new dataset.
D. G. T. Meranggi,	CNN with ResNet-18	598 images	88.88%	Includes data cleaning and	Accuracy affected by

N. Yudistira, and Y. A. Sar [7]	architecture	with 5 types of motifs		normalization steps that helps to improve the consistency of data.	similar visual characteristic and require a large amount of data for training.
I. Nurhaida, V. Ayumi, D. Fitrianah, R. a. M. Zen, H. Noprisson, and H. Wei [8]	DNN with Batch Normalisation	5 classes	85.57%	Use of Batch Normalisation improves convergence speed and reduce overfitting.	Require significant computational resources and training time and have limited interpretability which makes it hard to identify errors occurred.
Negara, Satria, Sanjaya, & Santoso [9]	ResNet-50	300 images	96%	High degree of accuracy. -Better feature learning	High computational complexity.
Joseph S., Hipiny I., Ujir H. et al. [10]	Scale Invariant Features Transform (SIFT)	300 images	99%	Performed well at optimal threshold value of 0.02.	Could not perform well on viewpoint sequence.
Rasyidi M. A. & Bariyah T. [11]	CNN	994 images from 6 categories	94%	High accuracy	Suffer from the vanishing gradient problem.
Rasyidi M. A. & Bariyah T. [11]	CNN with DenseNet	994 images from 6 categories	99%	Higher accuracy than normal CNN.	Unsustainable as each layer takes input not only from the previous layer but from all previous layers.
Iqbal Hussain	ResNet	3540 images	99%	Robust and achieve state-	Require significant

M. A., Khan B., Wang Z. & Ding S.[12]				of-the-art accuracy even when physical properties of the fabric changed.	computational resources and training time.
Rauf M., Jehanzeb M., Ullah U. et al. [13]	Deep Convolution Neural Network (DCNN)	180 images	96.16%	High capacity for non-linear classification and generalisation capabilities.	High complexity
Riski A.,Winata E. & Kamsyakawuni A.[14]	Backpropagation Algorithm	150 images	98%	Accuracy can be increased by increasing number of hidden layers and neurons.	Accuracy depends on number of hidden layers and neurons.
Puarungroj & Boonsirisumpun [15]	MobileNet	4500 images	98.223%	Smaller size of model.	Lower performance than other CNN architecture.
Andrian R., Naufal M., Hermanto B et al. [16]	k-Nearest Neighbour (k-NN)	100 images	97.96%	Makes highly accurate predictions.	Changes in the level of accuracy is unstable.
Ilahi M., Apriyani C., Desiani A. et al. [17]	CNN	1067 images	96%	Good seen with high precision.	No semantic network for node training.
M. D. Firdaus & H. Nugroho[18]	Cosine Similarity	200 images	91% for stamp batik 95% for hand drawn batik	Low complexity	Accuracy affected by image quality and light quality.

D. M. S. Arsa & A. A. N. H. Susila[19]	VGG16+Support Vector Machine (SVM)	300 images with 50 classes	97%	Good performance with limited dataset available.	Unknown performance as it is not being test with larger dataset.

Table 2.1 shows the performance comparison of different researchers with different approaches. The table includes several approaches with high performance in classifying images. These approaches include Scale Invariant Features Transform (SIFT) with 99% accuracy, CNN with DenseNet with 99% accuracy and ResNet with 99% accuracy. These approaches have the advantage of high accuracy with disadvantages, such as the unsustainability of DenseNet due to its high computational complexity and the requirement of a large amount of data for training ResNet.

In terms of computational complexity and training time, some approaches require significant resources, such as CNN with VGG-19 and ResNet, while others, such as Cosine Similarity and SIFT, have low complexity. It impacts the feasibility of the classification task. The interpretability is also different between different approaches. Some approaches, such as SIFT and Cosine Similarity, have low complexity and are easy to interpret, while others, such as CNN with DenseNet and DNN with Batch Normalisation, have limited interpretability, which can make it hard to identify errors that occur during classification.

2.4 Deep Learning

Recently, machine learning has gained popularity in research and has been used in a wide range of applications, including text mining, spam detection, video recommendation, image classification, and multimedia concept retrieval. Deep learning is one of the most often used machine learning methods in these applications [20].

The deep learning process is divided into various steps, which are data collection, data preparation, model selection, model training, model validation, and parameter tuning. To get started, machine learning requires reliable data in order to ensure the machine learning model can find the appropriate patterns. The model's accuracy is determined by the quality of the data provided to the computer. Incorrect or outdated data will produce inaccurate or inappropriate results or predictions [21].

Then, data preparation in terms of accurately labelling the data, cleaning the data by removing undesired data such as unclear images, small sizes, repeated images, and so on. The dataset is then divided into training and testing data. Training dataset is involved in the model learning process while the testing dataset is used to evaluate the model's accuracy after training.

After that, selecting the model is crucial as it determines the output obtained after training with the model algorithm. As a result, it is critical to select a model that is applicable to the current project. The training model's parameters such as image size, batch size, and epoch size need to be properly set during the model training phase. The batch size is the number of samples processed before updating the model, whereas the epoch is the number of times your training data is cycled through.

Finally, when the model is not performing well enough, validate it and tune its parameters. Model validation can be done by evaluating the model's performance with data that has never been seen before. To improve the model's accuracy and efficiency, the number of datasets can be increased and values of the training parameter can be tuned.

Deep Neural Network (DNN) is one of the deep learning algorithms that is designed to learn from large amounts of data and commonly used in image and speech recognition. It has an ability to learn complex features from raw data without the need for manual feature engineering which makes them suitable for tasks where the features are not well-defined or difficult to extract. However, DNNs require significant computational resources and training time, which can be a disadvantage for some applications. Additionally, DNNs have limited interpretability, which can make it hard to identify errors that occur during classification.

Another deep learning algorithm that is widely known is Convolutional Neural Network (CNN). It is normally used in analysing visual data such as images. Principles from linear algebra, particularly convolution operations are applied to extract features and identify patterns within images. The advantage of CNN includes their weight-sharing feature that reduces the number of trainable network parameters which makes the training process more efficient and less prone to overfitting. Besides, CNN is able to automatically extract features from input data which makes it suitable for image recognition tasks than DNN which is more suitable in processing sequential data such as natural language processing or speech recognition.

2.5 Convolution Neural Network

CNN (Convolution Neural Network) is one of the deep learning neural network architectures which is commonly implemented in computer vision projects [22]. It processes data that applies a grid-like topology to analyse data such as images. The CNN architecture has shown excellent performance in a variety of computer vision and machine learning tasks.

CNN algorithm is frequently used in current machine learning applications due to its continuing record-breaking efficiency. These CNNs employ linear algebra to operate. Matrix vector multiplication is the main technique in representing data and weights [23]. It is a mathematical construct that includes three types of layers which are named convolution, pooling and fully connected layer [22]. Convolution and pooling layers carry out feature extraction while fully connected layer maps the extracted features generated from previous layers into final output. The overview of CNN architecture is as shown in Figure 2.7. The details of the layers in CNN architecture will be further discussed in the following subtopics.

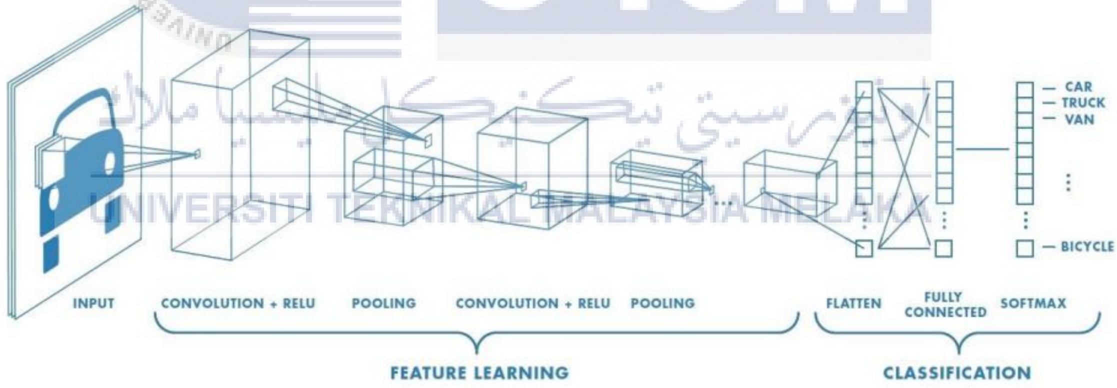

Figure 2.7: CNN Architecture [23]

2.5.1 Convolution Layer

An essential component of the CNN architecture that extracts features is the convolution layer. Convolution is a mathematical operation which applies a small array of numbers, known as a kernel, to a larger array of numbers, known as a tensor.

In CNNs, this technique is utilised for feature extraction. Convolution is done by moving the kernel across the input tensor, calculating the product of each element in the kernel and the corresponding element in the input tensor and summing the results to obtain a value for the output tensor. This procedure is done for every point in the tensor to generate a feature map. It is repeated using multiple kernels to produce an arbitrary number of feature maps representing independent characteristics of the input tensors; different types of kernels can be thought of as different feature extractors.

Two important hyperparameters that determine the convolution operation are the size and number of kernels. Usually, the first one is 3x3, although it may also be 5x5 or 7x7. The depth of the output feature maps is determined by the latter, which is fully arbitrary [22]. While research in [24] indicates that multiple convolutional layers can be used to extract various features from the input image.

CNN utilises multiple convolutional layers. The first convolutional layer extracts some fundamental characteristics, whereas the second convolutional layer obtains advanced features from the fundamental features. As an example, if we are predicting a cat, the first convolutional layer may merely extract some edges or lines, whereas the second convolutional layer may extract some external organs, such as the cat's ears, nose, and eyes. Through this layer-by-layer process, CNN managed to identify the object as a cat. Additionally, the study in [24] had also introduced 5 distinct types of convolutions, each with different advantages, as indicated in the Table 2.2:

Table 2.2: Comparison of different types of convolution [23]

Convolution	Zero-padding	Stride	Groups	Dilation rate	Benefits
Normal convolution	0	1	0	1	Basic and Simple
Convolution with padding	Usually more than 1	1	1	1	Ensure networks reach deep
Strided convolution	Flexible	Usually more than 2	1	1	Like pooling
Grouped convolution	Flexible	Flexible	Usually more than 2	1	Reduce quantity of parameters
Dilated convolution	Flexible	Flexible	1	Usually more than 2	Expand receptive field

2.5.2 Pooling Layer

A pooling layer presents an average down sampling operation, minimising in-plane dimension of the feature maps to introduce translation consistency to minor alterations and distortions and reducing the amount of subsequent training sets. It is worth noticing that none of the pooling layers have learnable parameters, other than filter size, stride, and padding are hyperparameters in pooling operations [22].

As far as we are aware, the convolutional layer successfully extracts beneficial features from the input image. However, having too many features may not always be favourable. The function of pooling layer's is to maintain only the most essential feature data when down sampling the obtained feature maps and compressing their resolution. The pooling layer has the advantages of translation invariance and a fixed kernel, which allows the number of parameters in the whole neural networks to be kept to a minimum [24]. There are four different types of pooling techniques that can be applied to keep the different features of an input layer. Table 2.3 displays the comparison:

Table 2.3: Comparison of different pool

Pooling	Size	Stride	Strategy	Benefits
Max pooling	Fixed	= Size	Fetch max pixel of local region	Preserve texture feature
Average pooling	Fixed	= Size	Calculate mean of local pixels	Preserve background information
Overlapping pooling	Fixed	< Size	Usually fetch max pixel of local region	Better representative ability
Spatial pyramid pooling	Flexible	= Size	Usually fetch max pixel of local region	Overcome various scales and higher accuracy

2.5.3 Fully Connected Layer

In a convolutional neural network, the fully connected layer serves as the overall “classifier”. The original data is mapped to the hidden feature space by convolutional, pooling, and activation function layers, whilst the learned "distributed feature representation" is mapped to sample space by a fully connected layer.

Fully connected layers can also be generated through convolutional operations based on feature maps after convolutional layers, pooling, and activation functions. Fully connected layer creates a one-dimensional vector by applying suitable convolution kernels to the feature maps. The purpose is to minimise the spatial dimension of all attributes from neural networks while also weighing them all, making it easier for the upcoming softmax layer to output classification probability [24]. Fully connected layers are frequently stacked on top of one another, with each intermediate layer voting on phantom “hidden” categories. This allows the network to learn increasingly complex feature combinations for better decision-making with each new layer. Backpropagation is utilised by the deep neural network to obtain the weights for the fully connected layers and the values for the convolution layer. Backpropagation is a method where the neural network calculates how much it needs to modify and adjust by considering the error in the result [23].

2.6 Classification Method

2.6.1 MobileNet

MobileNet is a type of convolutional neural network (CNN) designed for efficient and lightweight image classification [25]. It is known for its high accuracy and low computational power requirement which makes it well-suited for deployment on mobile and embedded devices. The network architecture of MobileNet is designed based on depthwise separable convolutions, which significantly reduces the number of parameters and computations required while maintaining strong performance in image classification tasks.

The MobileNet design can be identified by its use of depthwise separable convolutions, which are made up of a depthwise convolution followed by a pointwise convolution [26]. This architecture enables a significant reduction in the amount of parameters and computations, making it ideal for resource-constrained environments like mobile devices. The depthwise separable convolutions allow the network to learn and extract features from input images in a highly effective manner, resulting in excellent accuracy in image classification tasks.

In the area of image classification, MobileNet is frequently utilised in transfer learning, which involves fine-tuning a pre-trained MobileNet model on a specific dataset to perform a new image classification task. This method takes advantage of the pre-trained MobileNet model's effective feature extraction capabilities and adapts it to the specific characteristics of the new dataset. By fine-tuning the model on the new dataset, it is possible to achieve high accuracy in classifying images based on the dataset's specific classes.

2.6.2 LeNet

LeNet, a Convolutional Neural Network (CNN) architecture, was originally developed for digit recognition and is one of the oldest models, dating back to 1998. The LeNet network is structured with a series of repeated blocks, each containing a convolutional layer followed by a pooling layer. In the implementation used, each block is terminated by a Rectified Linear Unit (ReLU) layer.

As shown in Figure 2.6, it clearly stated that the LeNet's architecture includes seven layers, which are three convolutional layers, two subsampling levels, and two fully linked layers. LeNet's input is a 32x32 grayscale image that passes through the first convolutional layer (C1) with six 28x28 convolution kernels and feature mapping size. The output of C1 is transferred via a subsampling layer (S2) with six 16x16 feature map, followed by a second convolutional layer (C3) with 16 convolution kernels of size 10x10. C3's output is transferred via another subsampling layer (S4) that consists of 16 5x5 feature maps, followed by a fully connected layer (C5) that contains 120 1x1 feature maps. C5's output is then transmitted via another fully connected layer (F6) with 84 units.

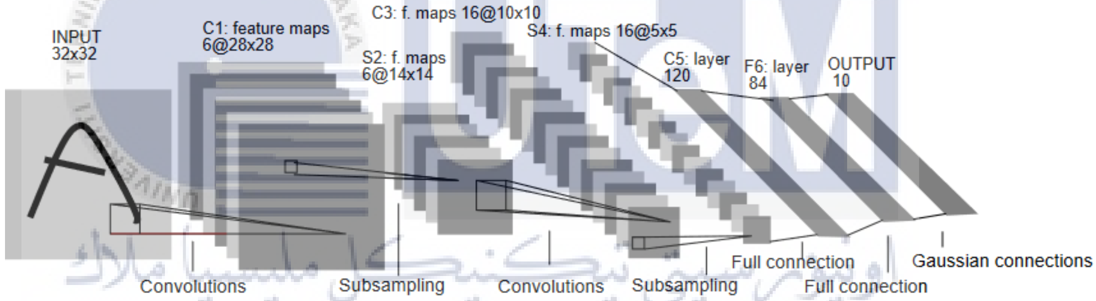

Figure 2.8: Architecture of LeNet [27]

Chapter 2 has covered the literature review done before starting with the development of the batik classification system as stated in the objective of this study. Literature review is done to gain knowledge about the previous research done regarding the proposed title. In the next following chapter, the methods applied in this study will be explained further. The methodology of this study is included and elaborated in detail to give a further understanding of the technique required in the development of the project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter covers the approaches and methods employed to achieve the research's objective. It includes flowcharts which illustrate how the study is conducted. The methodology is implemented consistently until the project is successfully completed. The methodology outlines the general framework of the research of how the objectives will be achieved.

3.2 Development of Batik Classification on Deep Learning

3.2.1 Project Development

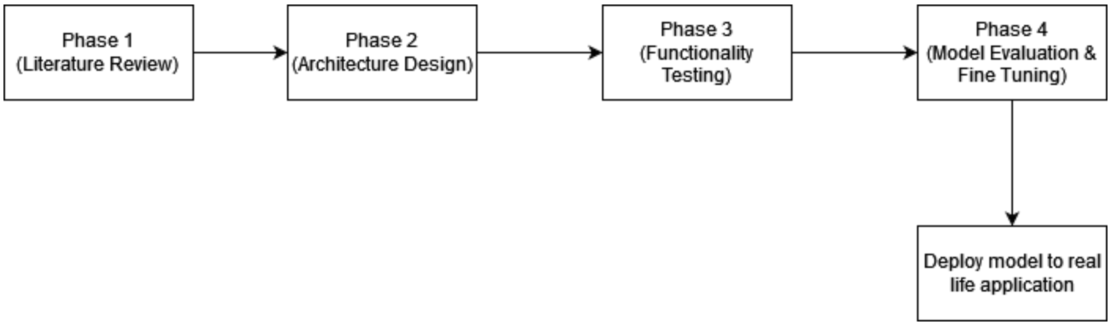

Figure 3.1: Block Diagram on Deep Learning for Batik Classification

Based on Figure 3.1, it is clearly shown that this project is divided into four phases. Phase 1 is the background studies of the project while Phase 2, 3 and 4 is based on the objectives of the project. Phase 2 is mainly focusing on the first objective which is developing a machine learning system using deep learning algorithms such as CNNs and transfer learning technique. While in Phase 3, the focus is on the second objective which is developing an automated classification system that extract features from Malaysia batik designs and classify them into various styles such as flowers, leaves and geometry. In Phase 4, evaluation of the accuracy and performance of the system based on their training and validation accuracy, loss and confusion matrix is done.

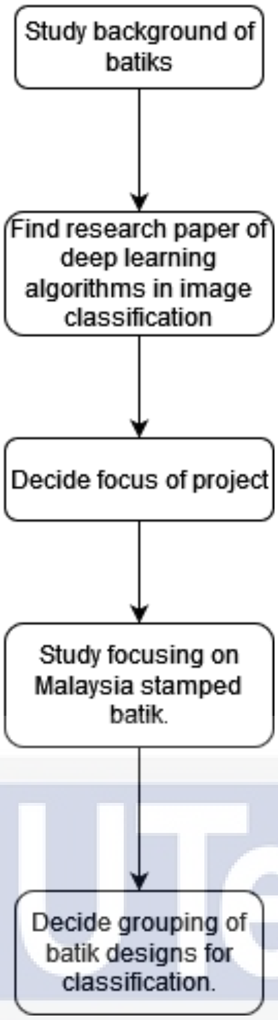

Figure 3.2: Background Studies Flowchart

Based on Figure 3.2, study of the background of batiks from various countries is carried out to have a deeper understanding. Research on deep learning algorithms is carried out to gain more knowledge about the difference, advantages and disadvantages among different deep learning algorithms in order to decide the suitable design method for the development of the project.

After studying about various types of batik, the focus of the project is decided on Malaysia stamped batik. Further research on Malaysia stamped batik is carried to gain more information which can help in identification of the speciality of Malaysia Batik among other batik designs from other countries. It is necessary to design the classification system for Malaysia batik and deciding the grouping of batik designs to helps in developing a digital database of batik designs.

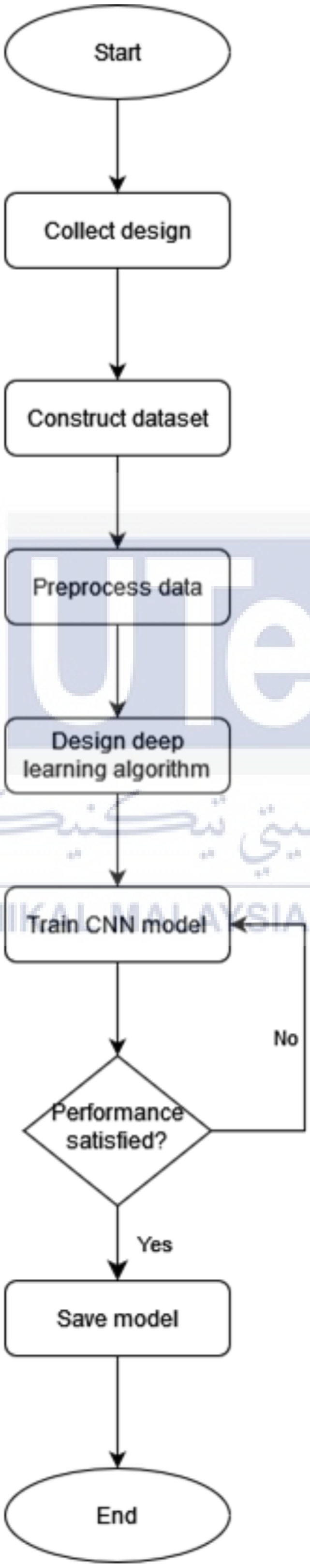

Figure 3.3: Architecture Design Flowchart

As shown in Figure 3.3, Phase 2 starts with the data collection which is collecting various types of Malaysia batik designs. Then, dataset is constructed based on the data collection. The dataset is divided into three categories which are training set, testing set and validation set. The images are then pre-processed to suit into the dataset. The design of deep learning algorithm is carried out based on the expected functionality which is to classify different styles of batik design. After designing the model, it is trained by using the prepared training set. The performance of the model is then evaluated and the hyperparameters are fine-tuned until the desired performance is satisfied. The model is then saved and proceed to Phase 3.

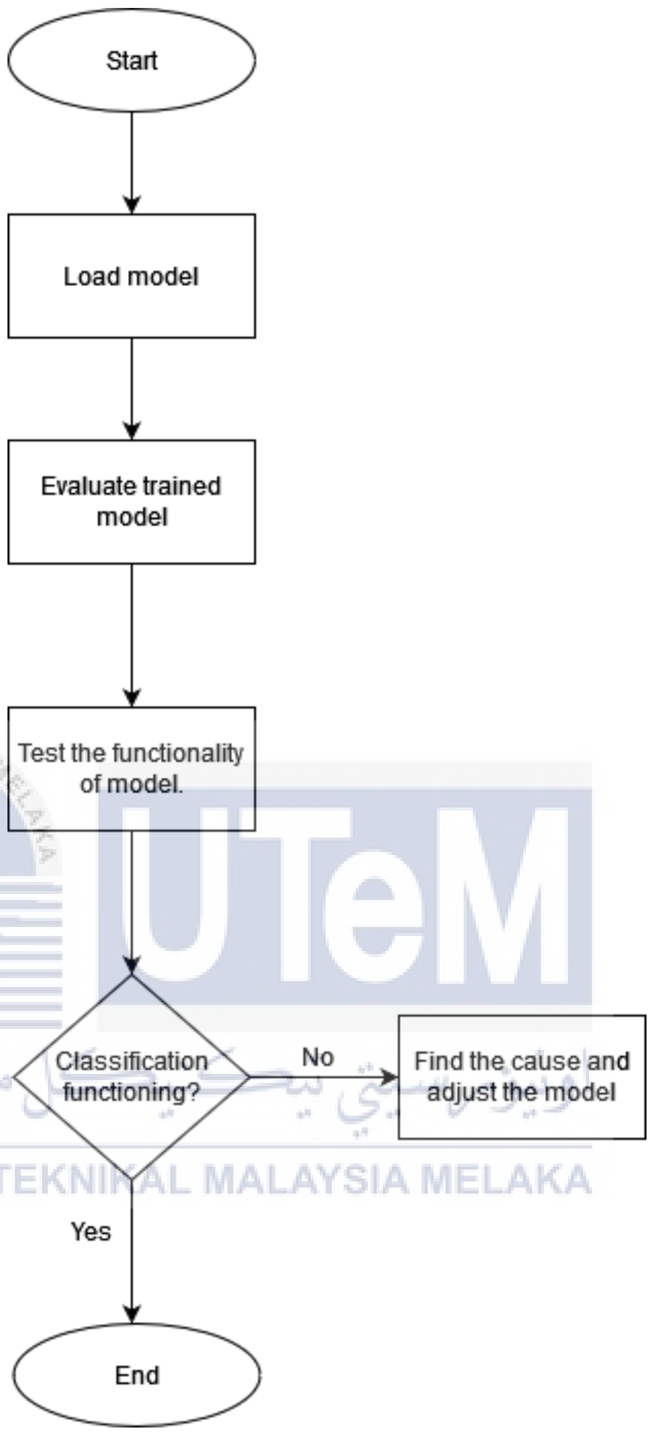

Figure 3.4: Functionality Testing Flowchart

In Phase 3 as shown in Figure 3.4, the trained model completed in Phase 2 is loaded. The model is evaluated by using the testing set and validation set prepared in Phase 1. The functionality of the model is tested to check whether the model can perform the classification of previously unseen data. If the classification is not functioning well, the model will be adjusted until the function is running well.

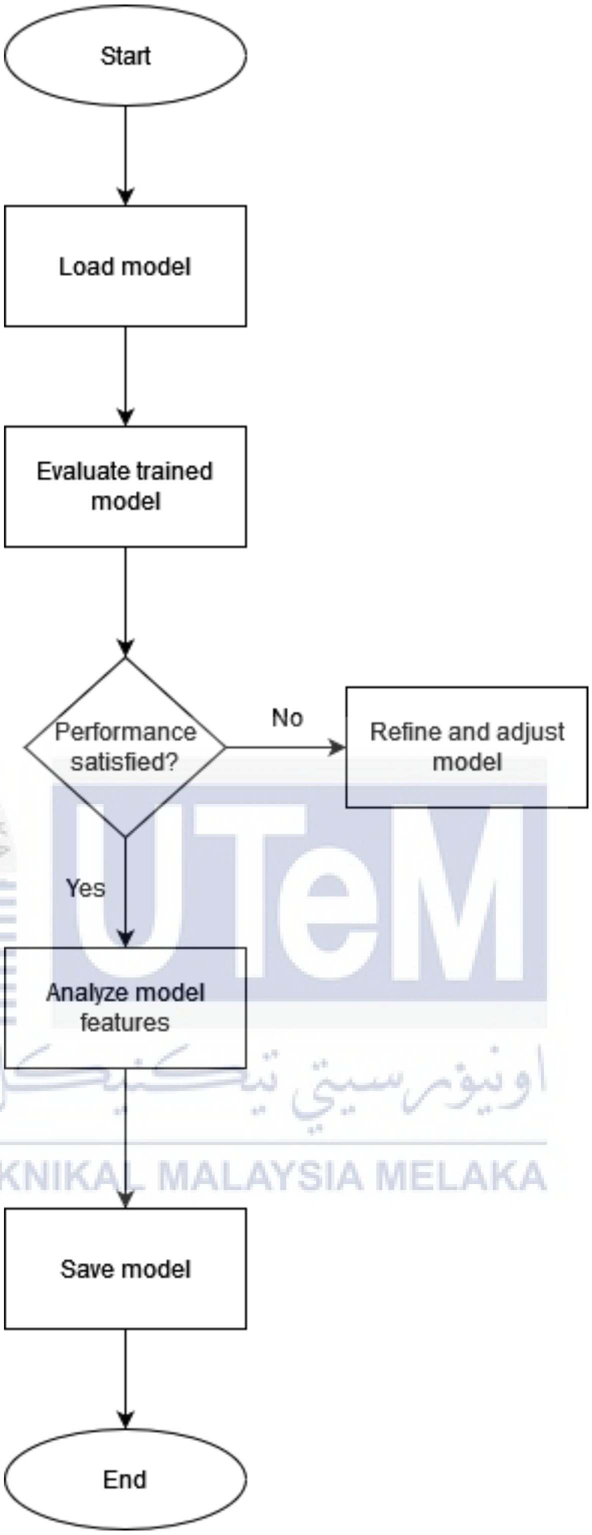

Figure 3.5: Model Evaluation and Fine-Tuning Flowchart

Based on the Phase 4 flowchart shown in Figure 3.5, the next step of the project development is to analyse the performance of the trained model by suitable evaluation metrics such as training and validation accuracy. Refinements and adjustments of the trained model are carried out until the desired performance is satisfied. Then, the model features are analysed to gain insights into the underlying patterns that are used by the model to make classification decisions. After every criterion is satisfied, the model is saved.

3.2.2 Background studies

Background studies on batiks from various countries is carried out to design the classification system. In this project, Malaysia and Indonesia stamped batik is focused. Malaysia and Indonesia stamped batik with various design and style is studied. Study is carried out on deep learning algorithms to understand the difference, advantages and disadvantages among different deep learning algorithms in order to decide the suitable design method for the development of the project.

3.2.3 Data Collection

A large dataset of Batik images from various regions and styles. The dataset includes a variety of images with different orientations, lighting conditions and backgrounds. For the Indonesia batik, it is available in wide variety in online databases. The Indonesia batik designs can be obtained from online datasets. While Malaysia batik, the resources is obtained by gathering images of batik pattern from social media and personal collections.

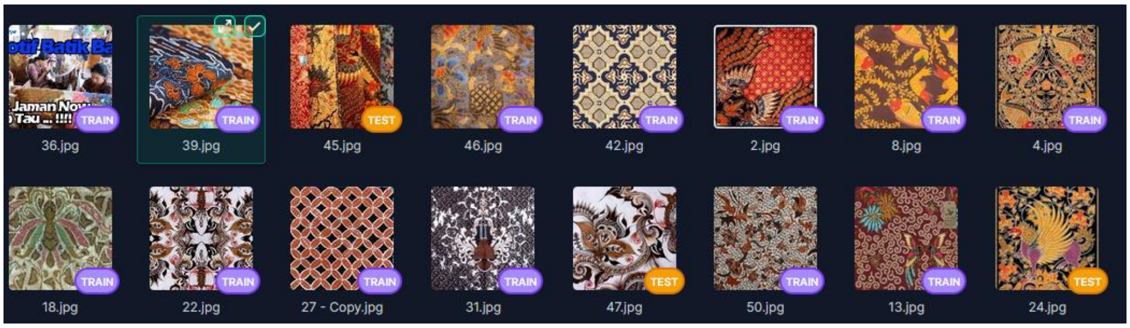

Figure 3.6: Indonesia Batik Dataset

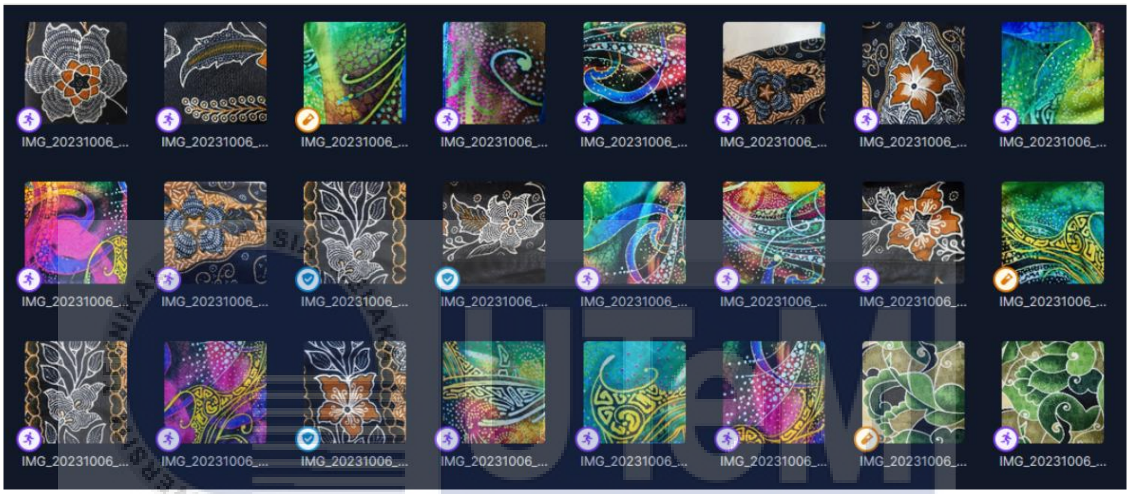

Figure 3.7: Malaysia Batik Dataset

3.2.3.1 Data Preprocessing

The dataset for training is prepared by performing data augmentation, normalization and resizing. Collected images is cleaned and pre-processed to remove duplicates, resize the images and convert images to a standardized format.

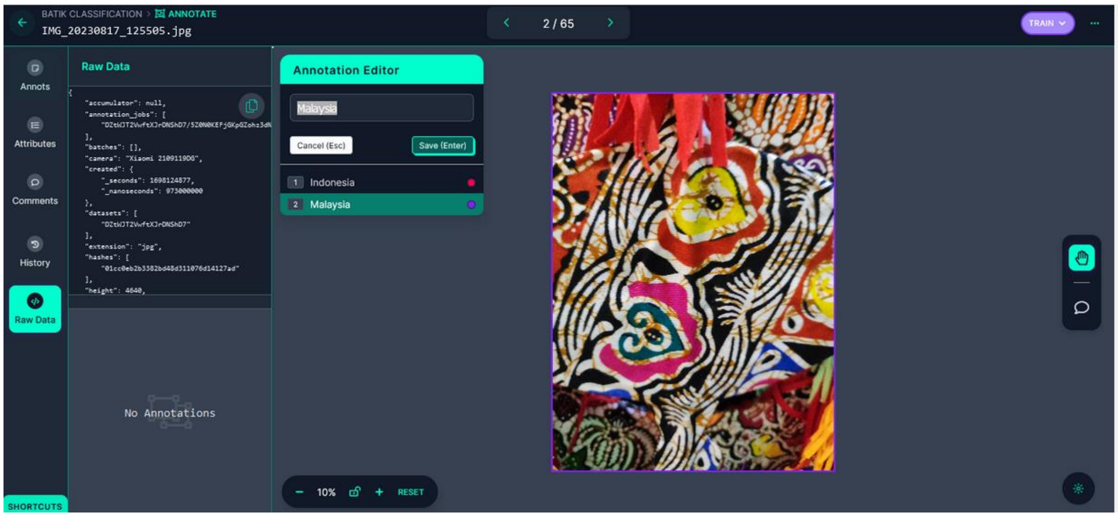

Figure 3.8: Annotation using Roboflow

Data augmentation is carried out such as adding noise or flipping images to increase the model's robustness. The collected image is then splitting into training set and testing set, which are used to evaluate the model's performance.

3.2.4 Model Training

Pre-trained CNN model is selected to design a classification system to extract features from images of the batik designs and classify them into two groups, which is Indonesia and Malaysia batik. Model selection is based on the desired functionality of the classification system. The CNN models selected for this study includes MobileNet v2, YOLO-v8 and LeNet-5.

The selected model is trained on the Batik design dataset that is pre-processed with a suitable optimizer and loss function. The model is trained by applying the train set and is validated using the valid set to check the overfitting of the trained model. The model's performance is then measured by computing the training and validation accuracy, training and validation loss and confusion matrix. Test set is separated to evaluate the performance of the model. This dataset is used to test the model functionality with previously unseen data.

3.2.5 Functionality Testing

The functionality of the trained CNN model is tested on the testing set. The classification function of the model is tested to ensure the model can automatically classify batik designs into various groups. It is done by extracting features from images of Malaysia Batik designs. If the classification function is not functioning well, the finding of root cause is carried out and the model is adjusted to fix the function.

3.2.6 Model Evaluation and Refinement

The trained CNN model is evaluated on the testing set and validated to assess its performance using suitable evaluation metrics such as accuracy, loss and confusion matrix. The CNN model is then refined by fine-tuning its hyperparameters or adjusting the CNN architecture. The hyperparameters of the model such as learning rate and number of layers are adjusted. Adjustment is carried out until the desired performance is achieved. The features learned by the CNN model is analysed to gain insights into the distinguishing characteristics of various Batik styles. It is done by visualizing the filters of CNN layers or by using feature visualisation techniques. Feature visualisation techniques help to gain insight into the model's performance, such as plotting the observations and decision boundaries in two dimensions.

3.2.6.1 Confusion Matrix

Confusion matrix is a specific table layout which functions in visualization of the performance of a deep learning algorithm and summarizes the performance of a classification model by comparing the predicted and actual class labels of a set of test data. The matrix is organized into rows and columns which the rows represent the predicted class and the columns represent the actual class. The diagonal elements of the matrix represent the true positive predictions, while the off-diagonal elements represent the false positive predictions.

The confusion matrix aids in identifying the areas of strength and weaknesses in model's prediction. It can be used to calculate various performance metrics such as precision, recall and F1-score.

Performance metric can be calculated as followed:

1. Precision: The proportion of true positive predictions out of all positive predictions. Precision is calculated as TP / (TP + FP), where TP is the number of true positive predictions and FP is the number of false positive predictions.
2. Recall: The proportion of true positive predictions out of all actual positive instances. Recall is calculated as TP / (TP + FN), where TP is the number of true positive predictions and FN is the number of false negative predictions.
3. F1-score: The harmonic mean of precision and recall. F1-score is calculated as 2 * (precision * recall) / (precision + recall).

3.6 Deep Learning Method

Deep learning, sometimes referred to as deep neural networks or neural learning, is a subset of machine learning. It is an artificial intelligence method that aims to imitate the ways the human brain processes data and creates patterns that can be used to make decisions. It is a form of representation-learning method that consists of multiple layers of data representation, obtained by composing simple but non-linear modules that each transform the representation at one layer (starting with the raw input data) into a representation at a higher, slightly more abstract layer [28]. Deep learning can learn and make decisions on unstructured or unlabelled data without supervision. Deep learning is commonly used in language translation, object detection, speech, visual recognition and many more.

3.6.1 MobileNet

MobileNet is one of the classes of CNN open source by Google. Tensor Flow models are part of the MobileNet family which can be called for mobile and embedded vision systems with constrained resources [29]. It helps to efficiently increase the accuracy of the models while considering the limited resources. This model is a built lightweight deep neural network based on a streamlined architecture that takes advantage of depth-separable convolutions.

The architecture of MobileNet decomposes the standard convolution into a depth convolution and a point convolution. Depth convolution splits the standard convolution into two separate layers, one for filtering and another for combining. Point convolution generates a 1x1 convolution to combine the outputs of depth convolution. As compared to traditional convolutions, depth separable convolution drastically reduces the number of parameters, computation, and model size.

3.6.2 YOLO (You Only Look Once)

YOLO (You Only Look Once) is a method for detecting objects in images using CNN. Unlike other object detection methods, YOLO does not require the use of separate techniques to generate proposals for regions of the image that may contain objects. Instead, it uses a single neural network to analyse the entire image at once and make object detection predictions [30]. YOLO also convert detection problem into regression problem [30]. This algorithm breaks down an image into smaller grid-like sections, or subregions, and then makes predictions about the bounding boxes and class probabilities for each subregion. In other words, it tries to identify objects within each subregion of the image and assigns a probability to each object based on its class. The algorithm divides the image into SxS grid. Two anchor boxes with various aspect ratios are utilised for each of these grids to identify objects of various sizes and scales. The anchor box that contains the object's centre has a positive mark on it.

The YOLO algorithm predicts two values for each anchor box. The first value is the probabilities for each class, and the second is the bounding box coordinates for the rectangle in terms of (x, y, w, h). We noticed that each grid cell predicts two boxes. but only considers the one with the highest Intersection over Union (IOU) with respect to the ground truth. Thus, the model predicts one set of class probabilities per grid cell.

The first YOLO object detector was released in 2016 and was designed by Joseph Redmon et.al. This architecture was substantially quicker than other object detectors when it was released and it quickly became the standard for real-time computer vision applications. Since then, other YOLO versions and modifications have been presented, each with a significant boost in performance and efficiency [31].

3.6.3 LeNet

LeNet is a classic convolutional neural network (CNN) intended for recognition tasks of handwritten digits using the MNIST dataset [27]. Yann LeCun et al. introduced the concept in 1998 and has been widely used in various applications such as character recognition, pedestrian detection and gas identification. LeNet-5, an improved version of LeNet launched in 1998, contains seven layers: two convolutional, two subsampling, and three fully linked layers. The architecture of LeNet-5 impacted future CNN models such as AlexNet and VGG. LeNet-4 is comparable to LeNet-5, with a few modifications of architectural characteristics. The feature map structure consists of four first-level maps, eight subsampling maps connected in pairs, and 16 feature maps.

The features of LeNet-5 include:

1. Input Layer: LeNet-5 takes in a grayscale image of dimensions 32x32 as its input.
2. Convolutional Layers: The initial operation in the network is convolution, utilizing a filter of size 5x5. This operation is performed using 6 filters, leading to a feature map with dimensions 28x28x6.
3. Average Pooling Layers: Following the convolution, an average pooling operation is executed, which reduces the feature map's dimensions by half.
4. Fully Connected Layers: After the convolution and average pooling operations, the network includes two layers that are fully connected.
5. Output Layer: The network concludes with a softmax classifier layer, which categorizes the images into their respective classes. This layer consists of 10 output units, signifying that the network can predict 10 distinct classes.

6. Activation Functions: The network employs a tanh activation function for the hidden layers, while the output layer uses a softmax activation function.

Chapter 3 had explained the methodology and techniques employed to complete this study. The framework of the study is clearly stated and explained followed by the method proposed to complete this study. In the next chapter, the results obtained from the study with the three chosen models including MobileNet v2, YOLO-v8 and LeNet-5 will be illustrated and further analysed. The performance of each approach will also be compared and further discussed.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The research aims to develop a batik classification system to classify Batik designs from different countries using the Convolutional Neural Network (CNN) method. This chapter will discuss the result obtained by applying three different CNN models which is MobileNet v2, YOLO v8 and LeNet-5 on a collection of Batik design photos and explain the performance of the two models in terms of accuracy and efficiency.

4.2 Database

Roboflow is used in developing the classification system. Roboflow is a platform that provides tools for computer vision. It functions in helping users for data collection and management. Tools such as annotation tools provided helps simplify work of preprocessing data including labelling and annotating. Different annotation types are included such as bounding boxes, polygons, key points and segmentation masks. The platform also provides different data augmentation techniques to help user in enhancing the diversity and quality of the dataset. Roboflow integrates seamlessly with popular deep learning frameworks such as TensorFlow and PyTorch. Roboflow Train provides a simple way to scale up GPU training on any custom dataset, removing the guesswork and fine-tuning that comes with managing dependencies.

4.3 Dataset

A total of 1825 batik images are collected from various resources. The images collected are organised by using Roboflow as shown in Figure 4.1. The dataset is divided into two classes, which is Indonesia Batik and Malaysia Batik. Indonesia batik class consists of 949 images while Malaysia batik class consists of 876 images. The details of the images in the dataset is clearly shown in Figure 4.2.

Figure 4.1: Database Interface in Roboflow

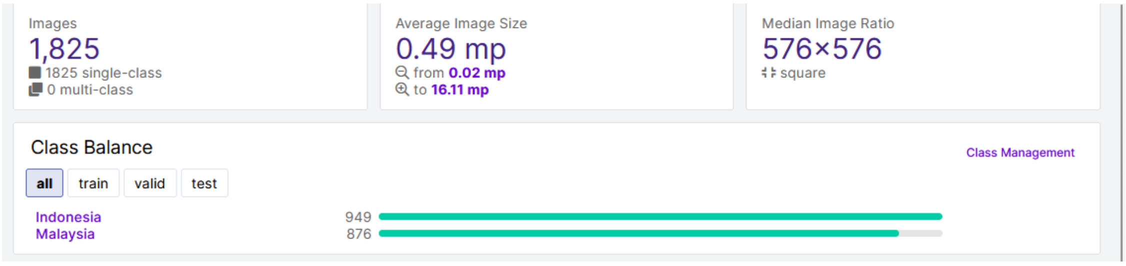

Figure 4.2: Details of dataset in Roboflow

Data augmentation process is then done by using Roboflow. The dataset details are stated clearly in Figure 4.3. A total number of 2507 images are generated after performing data augmentation. 80% of them is allocated for training which is 1995 images, 14% for validation which is 342 images and 7% for testing which is 170 images.

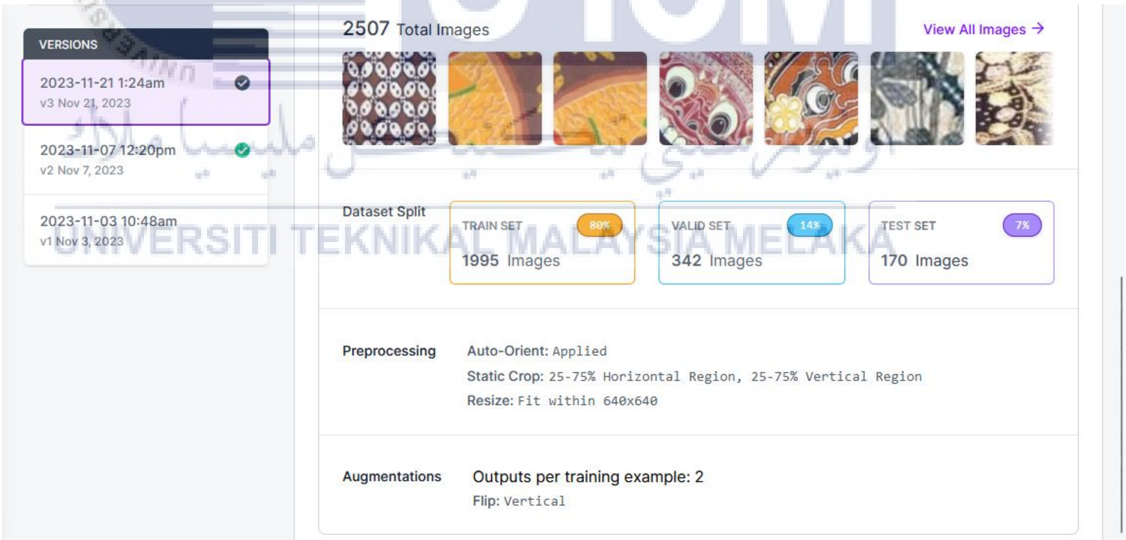

Figure 4.3: Details of dataset after applying data augmentation

The dataset is split into train set, valid set and test set to allow the model to be evaluated according to its performance on the unseen batik designs. The splitting process is done randomly in order to reduce the bias and increase the dependability of the result.

4.4 Performance Evaluation

Three CNN architectures are applied to develop the batik classification system. CNN architectures chosen are MobileNet v2, YOLOv8 and LeNet-5.

4.4.1 MobileNet v2

Figure 4.4 shows the summary of the model used which is MobileNet v2.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_160 (Func  (None, 5, 5, 1280)        2257984
 tional)

 global_average_pooling2d (  (None, 1280)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 1)                 1281

=================================================================
Total params: 2259265 (8.62 MB)
Trainable params: 1281 (5.00 KB)
Non-trainable params: 2257984 (8.61 MB)
_________________________________________________________________
```

Figure 4.4: Model Summary of MobileNet v2

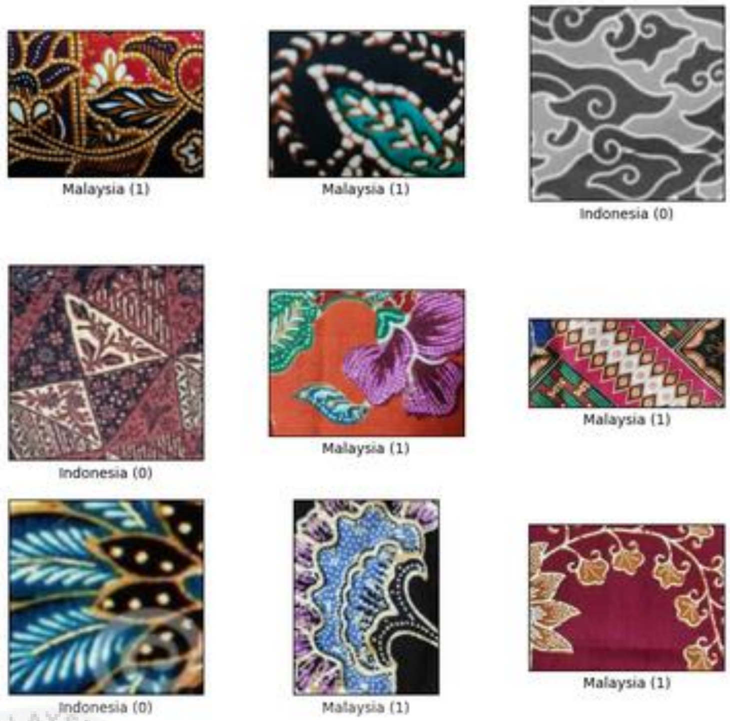

Figure 4.5: Sample of raw data in MobileNet v2

Figure 4.5 shows a few samples of raw data used to train MobileNet v2 model. The purpose of executing the sample data is to make sure that the images is labelled acccording to their respective classes.

```
Epoch 1/10
200/200 [==============================] - 57s 245ms/step - loss: 0.4497 - accuracy: 0.7629 - val_loss: 0.2836 - val_accuracy: 0.8830
Epoch 2/10
200/200 [==============================] - 43s 197ms/step - loss: 0.2115 - accuracy: 0.9243 - val_loss: 0.1681 - val_accuracy: 0.9415
Epoch 3/10
200/200 [==============================] - 40s 182ms/step - loss: 0.1415 - accuracy: 0.9559 - val_loss: 0.1301 - val_accuracy: 0.9591
Epoch 4/10
200/200 [==============================] - 40s 183ms/step - loss: 0.1129 - accuracy: 0.9624 - val_loss: 0.1137 - val_accuracy: 0.9678
Epoch 5/10
200/200 [==============================] - 42s 199ms/step - loss: 0.0979 - accuracy: 0.9654 - val_loss: 0.1055 - val_accuracy: 0.9708
Epoch 6/10
200/200 [==============================] - 45s 211ms/step - loss: 0.0877 - accuracy: 0.9714 - val_loss: 0.1015 - val_accuracy: 0.9678
Epoch 7/10
200/200 [==============================] - 45s 215ms/step - loss: 0.0803 - accuracy: 0.9724 - val_loss: 0.0993 - val_accuracy: 0.9678
Epoch 8/10
200/200 [==============================] - 45s 213ms/step - loss: 0.0747 - accuracy: 0.9739 - val_loss: 0.0967 - val_accuracy: 0.9708
Epoch 9/10
200/200 [==============================] - 41s 184ms/step - loss: 0.0705 - accuracy: 0.9764 - val_loss: 0.0958 - val_accuracy: 0.9708
Epoch 10/10
200/200 [==============================] - 41s 193ms/step - loss: 0.0667 - accuracy: 0.9779 - val_loss: 0.0959 - val_accuracy: 0.9678
```

Figure 4.6: Result of MobileNet v2 Model Training

Figure 4.6 shows the result of the training of MobileNet v2 model. From the figure, it is stated that the number of epochs selected for the MobileNet v2 model training is 10. The MobileNet v2 model demonstrates consistent improvement in both training and validation accuracy over the 10 epochs. The model starts with a training accuracy of 76.29% and a validation accuracy of 88.30% in the first epoch, and gradually improves to a training accuracy of 97.79% and a validation accuracy of 96.78% in the 10th epoch. The training loss decreases from 0.4497 to 0.0667, while the validation loss decreases from 0.2836 to 0.0959 over the 10 epochs. This indicates that the model is effectively learning the features of the training data and generalizing well to the validation data, as the loss decreases and the accuracy increases over the epochs. The results obtained from the model is then visualized into graphs as shown in Figure 4.7 and 4.8.

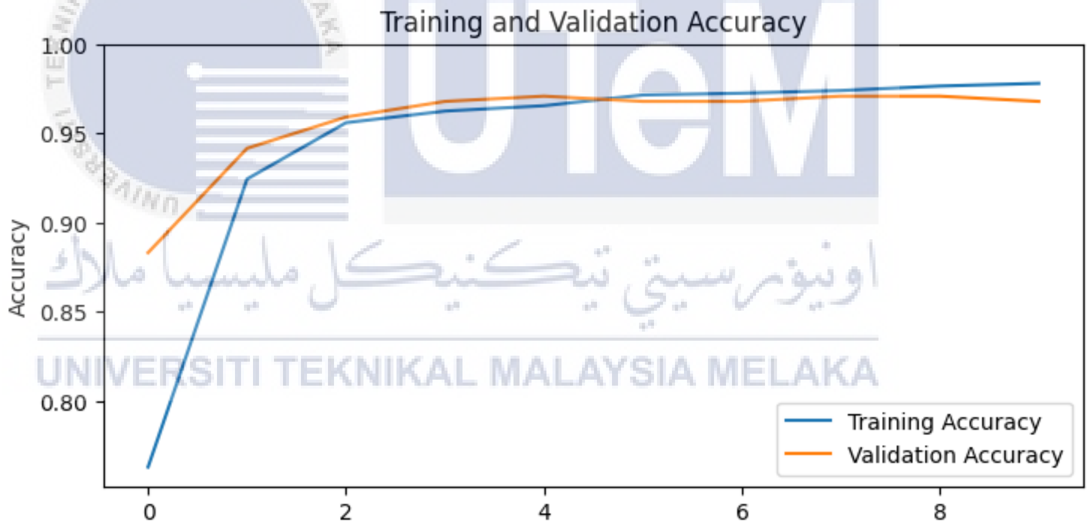

Figure 4.7: Training and Validation Accuracy of MobileNet v2

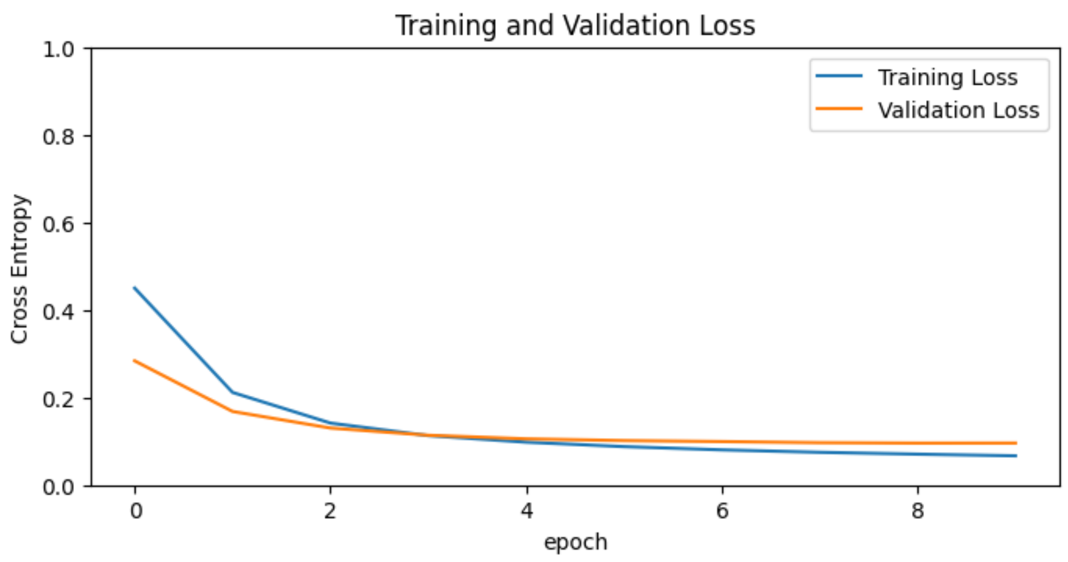

Figure 4.8: Training and Validation Loss of MobileNet v2

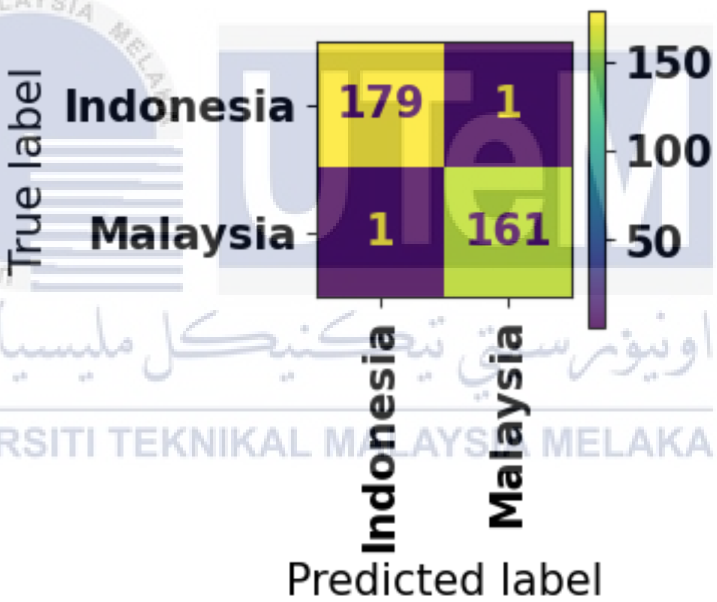

Figure 4.9: Confusion Matrix of MobileNet v2

Precision = 179 / (179 + 1) = 0.9944

Recall = 179 / (179 + 1) = 0.9944

F1-score = 2 * (0.9944 * 0.9944) / (0.9944 + 0.9944) = 0.9944

As shown in Figure 4.9, the bottom right value, 161, represents the true positives which are the cases of the model correctly predicted the positive class. The top left value, 179, represents the true negatives which are the cases of the model correctly predicted the negative class. The top right value, 1, represents the false positives (Type I error) which are the cases of the model incorrectly predicted the positive class. The bottom left value, 1, represents the false negatives (Type II error) which are the cases of the model incorrectly predicted the negative class.

The model has a high number of true positives and true negatives and a very low number of false positives and false negatives. This indicates a high level of accuracy and a low level of confusion between the two classes, as evidenced by the small number of misclassifications.

4.4.2 YOLOv8

e	train/loss	metrics/accuracy_top1	metrics/accuracy_top5	val/loss	lr/pg0	lr/pg1	lr/pg2
1	0.44764	0.96471	1	0.37515	0.0002361	0.0002361	0.0002361
2	0.152	0.96471	1	0.35588	0.00045063	0.00045063	0.00045063
3	0.091	0.96471	1	0.35045	0.0006416	0.0006416	0.0006416
4	0.0997	0.97059	1	0.33889	0.00060797	0.00060797	0.00060797
5	0.08768	0.95882	1	0.34511	0.00060797	0.00060797	0.00060797
6	0.04981	0.97059	1	0.33876	0.00057263	0.00057263	0.00057263
7	0.05232	0.97647	1	0.33005	0.00053728	0.00053728	0.00053728
8	0.03558	0.97647	1	0.33281	0.00050194	0.00050194	0.00050194
9	0.02444	0.98235	1	0.33236	0.0004666	0.0004666	0.0004666
10	0.01583	0.98235	1	0.32945	0.00043126	0.00043126	0.00043126
11	0.01609	0.98235	1	0.337	0.00039591	0.00039591	0.00039591
12	0.02007	0.98824	1	0.328	0.00036057	0.00036057	0.00036057
13	0.01575	0.98824	1	0.33674	0.00032523	0.00032523	0.00032523
14	0.01389	0.98824	1	0.3254	0.00028988	0.00028988	0.00028988
15	0.00706	0.98824	1	0.3248	0.00025454	0.00025454	0.00025454
16	0.01256	0.98824	1	0.33391	0.0002192	0.0002192	0.0002192
17	0.00977	0.98824	1	0.32286	0.00018385	0.00018385	0.00018385
18	0.00929	0.98824	1	0.32389	0.00014851	0.00014851	0.00014851
19	0.00583	0.98824	1	0.32396	0.00011317	0.00011317	0.00011317
20	0.00368	0.98824	1	0.32408	7.78E-05	7.78E-05	7.78E-05

Figure 4.10: Result of YOLOv8 training

Figure 4.10 shows the result of the training process of the YOLOv8 model. As stated in the figure, YOLOv8 model demonstrates consistent improvement in both training and validation accuracy over the 20 epochs. The model starts with a training accuracy of 96.47% and a validation accuracy of 96.47% in the first epoch, and gradually improves to a training accuracy of 98.82% and a validation accuracy of 98.82% in the 20th epoch. The training loss also decreases from 0.44764 to 0.00368, while the validation loss decreases from 0.37515 to 0.32408 over the 20 epochs. This indicates that the model is effectively learning the features of the training data and generalizing well to the validation data, as the loss decreases and the accuracy increases over the epochs. The result obtained from the training process is then plotted into graphs as shown in Figure 4.11.

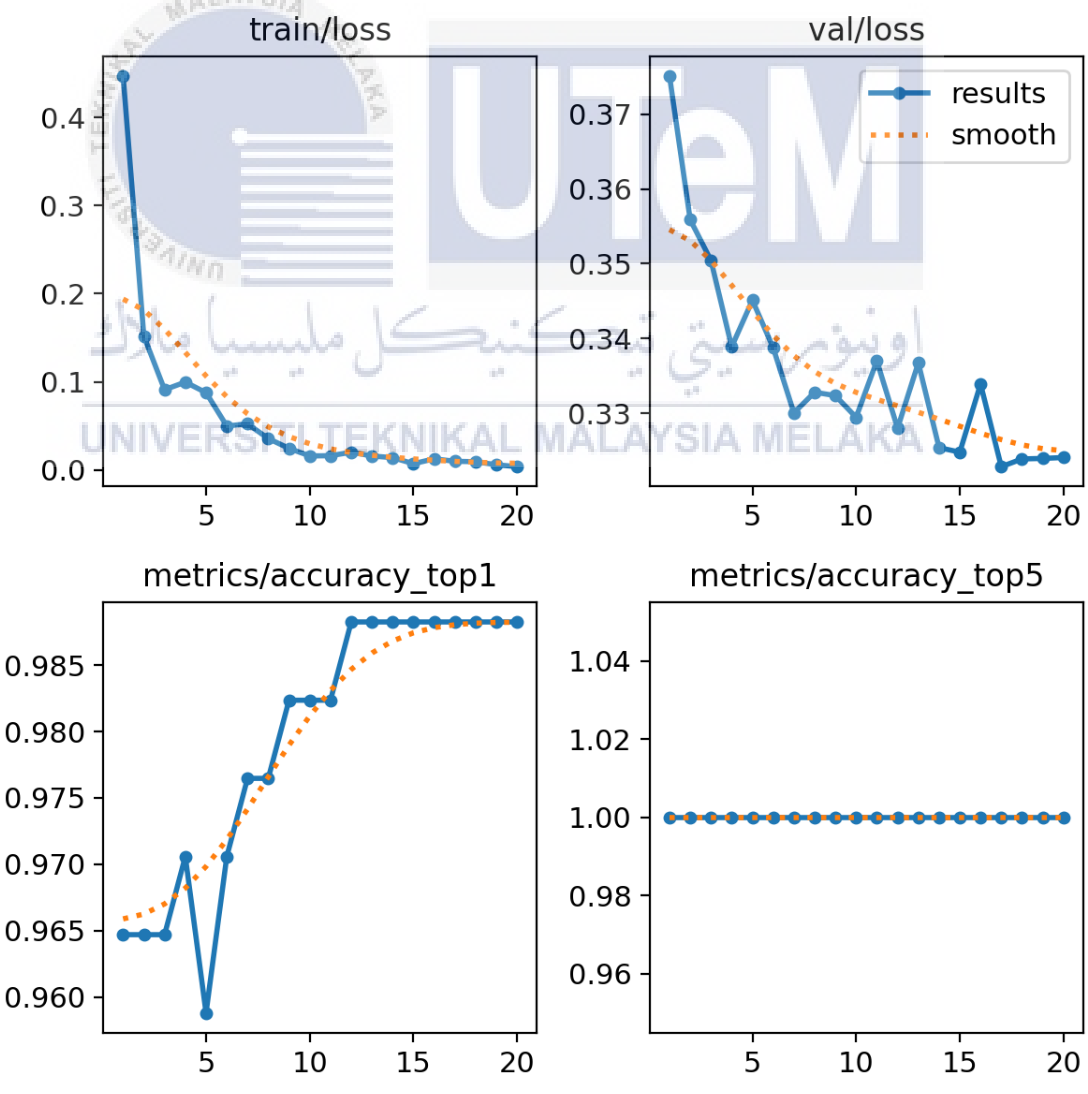

Figure 4.11: Training Result Graph of YOLOv8

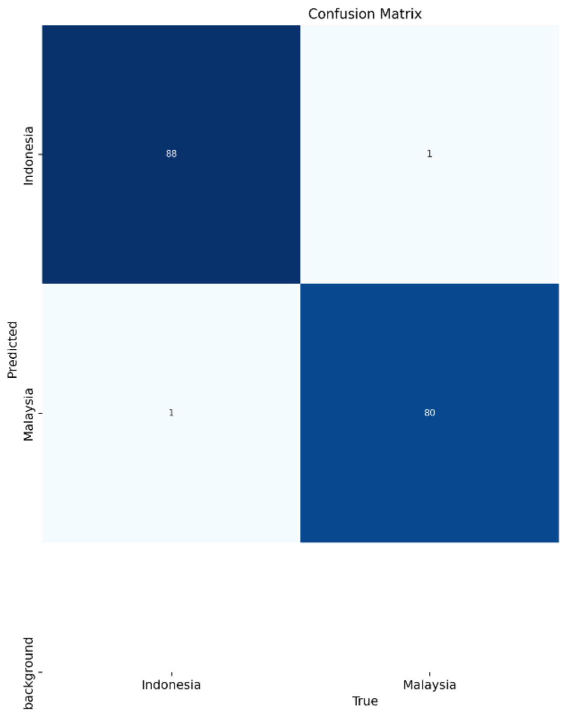

Figure 4.12: Confusion Matrix of YOLOv8

Precision = 80 / (80 + 1) = 80 / 81 = 0.9877

Recall = 80 / (80 + 1) = 80 / 81 = 0.9877

F1-score = 2 * (0.9877 * 0.9877) / (0.9877 + 0.9877) = 2 * (0.9756) / (1.9754) = 1.9512 / 1.9754 = 0.9853

As shown in Figure 4.12, the bottom right value, 80, represents the true positives which are the cases of the model correctly predicted the positive class. The top left value, 88, represents the true negatives which are the cases of the model correctly predicted the negative class. The top right value, 1, represents the false positives (Type I error) which are the cases of the model incorrectly predicted the positive class. The bottom left value, 1, represents the false negatives (Type II error) which are the cases of the model incorrectly predicted the negative class.

The confusion matrix analysis shows that the model has a high number of true positives and true negatives with a very low number of false positives and false negatives. This indicates that the model has a high accuracy in its predictions.

4.4.3 LeNet-5

Figure 4.13 clearly stated the model summary of LeNet-5 used for the development of the batik classification system.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_2 (Conv2D)           (None, 474, 636, 6)       156       
                                                                 
 max_pooling2d_2 (MaxPoolin  (None, 237, 318, 6)       0         
 g2D)                                                            
                                                                 
 conv2d_3 (Conv2D)           (None, 233, 314, 16)      2416      
                                                                 
 max_pooling2d_3 (MaxPoolin  (None, 116, 157, 16)      0         
 g2D)                                                            
                                                                 
 flatten_1 (Flatten)         (None, 291392)            0         
                                                                 
 dense_3 (Dense)             (None, 120)               34967160  
                                                                 
 dense_4 (Dense)             (None, 84)                10164     
                                                                 
 dense_5 (Dense)             (None, 1)                 85        
                                                                 
=================================================================
Total params: 34979981 (133.44 MB)
Trainable params: 34979981 (133.44 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 4.13: Model Summary of LeNet-5

```
Found 1995 images belonging to 2 classes.
Found 342 images belonging to 2 classes.
Found 170 images belonging to 2 classes.
Epoch 1/20
249/249 [==============================] - 143s 550ms/step - loss: 1.4699 - accuracy: 0.7403 - val_loss: 0.6732 - val_accuracy: 0.8036
Epoch 2/20
249/249 [==============================] - 139s 559ms/step - loss: 0.7057 - accuracy: 0.8027 - val_loss: 0.4521 - val_accuracy: 0.8423
Epoch 3/20
249/249 [==============================] - 136s 547ms/step - loss: 0.6322 - accuracy: 0.8007 - val_loss: 0.5215 - val_accuracy: 0.8363
Epoch 4/20
249/249 [==============================] - 135s 542ms/step - loss: 0.5633 - accuracy: 0.8108 - val_loss: 0.5261 - val_accuracy: 0.8304
Epoch 5/20
249/249 [==============================] - 132s 529ms/step - loss: 0.5297 - accuracy: 0.8198 - val_loss: 0.5690 - val_accuracy: 0.7946
Epoch 6/20
249/249 [==============================] - 138s 554ms/step - loss: 0.4676 - accuracy: 0.8319 - val_loss: 0.3623 - val_accuracy: 0.8452
Epoch 7/20
249/249 [==============================] - 133s 532ms/step - loss: 0.4280 - accuracy: 0.8450 - val_loss: 0.3333 - val_accuracy: 0.8780
Epoch 8/20
249/249 [==============================] - 131s 523ms/step - loss: 0.4049 - accuracy: 0.8546 - val_loss: 0.2974 - val_accuracy: 0.8869
Epoch 9/20
249/249 [==============================] - 131s 526ms/step - loss: 0.3967 - accuracy: 0.8490 - val_loss: 0.3070 - val_accuracy: 0.8929
Epoch 10/20
249/249 [==============================] - 129s 520ms/step - loss: 0.3431 - accuracy: 0.8626 - val_loss: 0.2852 - val_accuracy: 0.9018
Epoch 11/20
249/249 [==============================] - 134s 540ms/step - loss: 0.3279 - accuracy: 0.8762 - val_loss: 0.2658 - val_accuracy: 0.9077
Epoch 12/20
249/249 [==============================] - 138s 554ms/step - loss: 0.3341 - accuracy: 0.8732 - val_loss: 0.2600 - val_accuracy: 0.8899

Epoch 13/20
249/249 [==============================] - 135s 541ms/step - loss: 0.3287 - accuracy: 0.8797 - val_loss: 0.2774 - val_accuracy: 0.9107
Epoch 14/20
249/249 [==============================] - 135s 541ms/step - loss: 0.3103 - accuracy: 0.8903 - val_loss: 0.2697 - val_accuracy: 0.9107
Epoch 15/20
249/249 [==============================] - 133s 535ms/step - loss: 0.2973 - accuracy: 0.8873 - val_loss: 0.2584 - val_accuracy: 0.9048
Epoch 16/20
249/249 [==============================] - 132s 529ms/step - loss: 0.2924 - accuracy: 0.8933 - val_loss: 0.2432 - val_accuracy: 0.9048
Epoch 17/20
249/249 [==============================] - 132s 530ms/step - loss: 0.2838 - accuracy: 0.8888 - val_loss: 0.2635 - val_accuracy: 0.8988
Epoch 18/20
249/249 [==============================] - 134s 539ms/step - loss: 0.2780 - accuracy: 0.8988 - val_loss: 0.2341 - val_accuracy: 0.9048
Epoch 19/20
249/249 [==============================] - 133s 536ms/step - loss: 0.2820 - accuracy: 0.8893 - val_loss: 0.2620 - val_accuracy: 0.8899
Epoch 20/20
249/249 [==============================] - 134s 537ms/step - loss: 0.2675 - accuracy: 0.9029 - val_loss: 0.2307 - val_accuracy: 0.8988
22/22 [==============================] - 2s 69ms/step - loss: 0.3286 - accuracy: 0.9000
Test Accuracy: 90.00%
Test Loss: 0.3286
```

Figure 4.14: Result of LeNet-5 Training

The training outcomes of the LeNet-5 model shown in Figure 4.14 illustrate its performance across 20 epochs. The model exhibits a consistent enhancement in both training and validation accuracy, commencing with a training accuracy of 74.03% and a validation accuracy of 80.36% in the initial epoch, and culminating in a training accuracy of 90.29% and a validation accuracy of 89.88% in the 20th epoch. Concurrently, the training loss diminishes from 1.4699 to 0.2675, and the validation loss decreases from 0.6732 to 0.2307 over the 20 epochs, signifying the model's adeptness in learning the training data and effectively generalizing to the validation data, as evidenced by the decreasing loss and increasing accuracy throughout the epochs. The outcomes obtained from the training is plotted into graphs shown in Figure 4.15 and Figure 4.16 for better understanding.

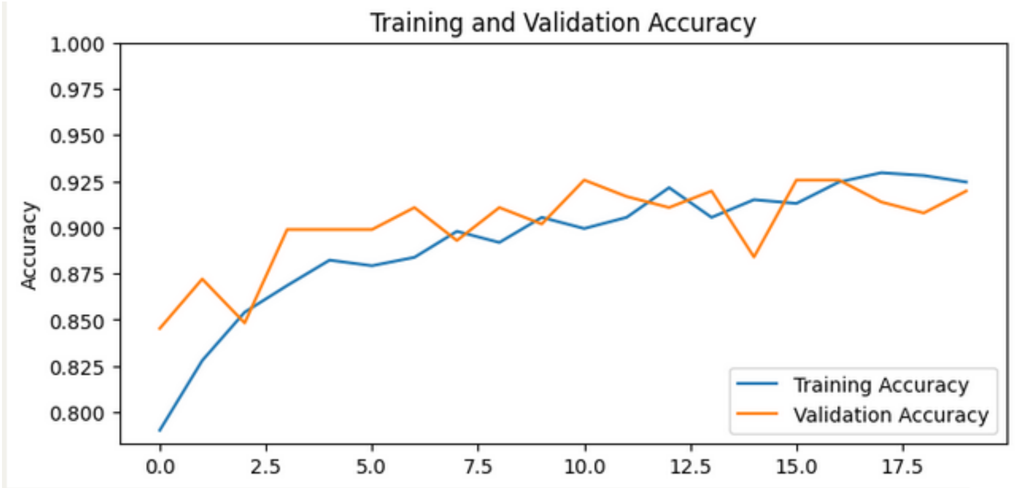

Figure 4.15: Training and Validation Accuracy of LeNet-5

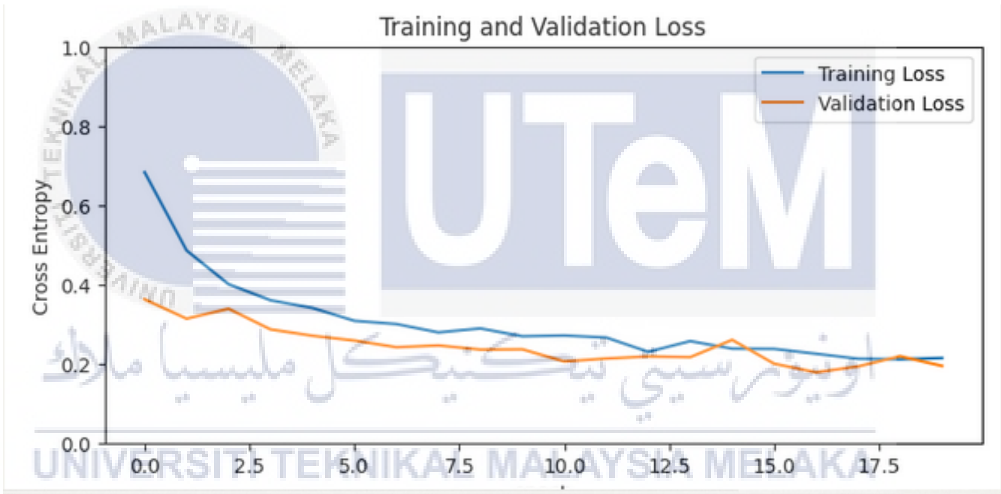

Figure 4.16: Training and Validation Loss of LeNet-5

.

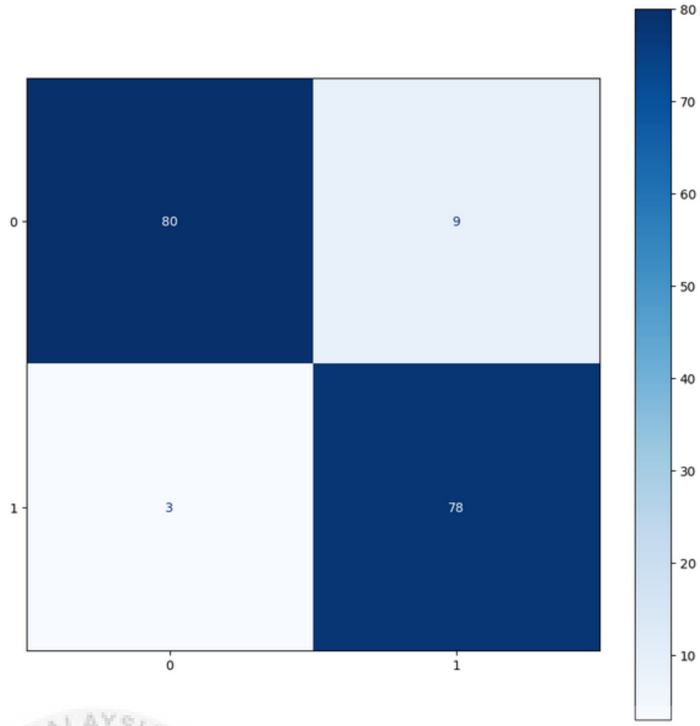

Figure 4.17: Confusion Matrix of LeNet-5

Precision = 80 / (80 + 9) = 80 / 89 = 0.8989

Recall = 80 / (80 + 3) = 80 / 83 = 0.9639

F1-score = 2 * (0.8989 * 0.9639) / (0.8989 + 0.9639) = 2 * (0.9305) / (1.8628) = 1.861 / 1.8628 = 0.997

From the confusion matrix shown in Figure 4.17, the model accurately classified 80 samples from class 1 and 78 samples from class 2. However, it misclassified 9 samples from class 2 as class 1 and 3 samples from class 1 as class 2. This indicates the model's proficiency in classifying both classes, with some misclassifications in each class. It has shown the model overall has a strong performance based on the indicated high accuracy and low misclassification rate.

4.5 Environment and Sustainability

4.5.1 SDG4 Quality Education

This study aims to develop a classification system to classify Indonesia and Malaysia batik, which aids the development of the digital database of batik designs. It helps in preserving the traditional knowledge and craftsmanship by recording them in digital records. The digital database can be used to educate and raise awareness about the art of batik, a significant part of the cultural heritage.

4.5.2 SDG17 Partnerships for the Goals

The development and implementation of the batik classification system involves collaboration between various stakeholders, including tech companies, Batik artisans, government agencies or non-governmental organizations to maximize the effectiveness of the batik classification system. For example, Batik artisans can contribute their knowledge and understanding of batik patterns, which is crucial for accurately classifying different types of batik. Besides, government agencies or non-governmental organizations can support the project in providing funding, facilitating partnerships and also promoting the use of the classification system.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

The general objective of this study is to compile and construct a new dataset of Malaysia batik for image classification and develop a batik classification system using deep learning algorithms that classify between Malaysia and Indonesia batik. A number of 876 images of Malaysia batik is included in the dataset before performing data augmentation. The batik classification system performed well with high accuracy exceeding 90%.

Besides, this study compared three different CNN models which includes MobileNet v2, YOLO-v8 and LeNet-5 in performing the batik classification task. The results for three models show that all three models are able to classify batik patterns according to their countries with high accuracy. YOLO-v8 model demonstrated a higher accuracy compared to other two models. This suggests that YOLO-v8 model is suitable for developing a batik classification system which needs a higher accuracy. While for the MobileNet-v2 model, it shows a slightly lower accuracy compared to the YOLO-v8 model. However, it is suitable for mobile application development due to its lightweight and low requirement of computational power.

5.2 Future Work

The future work of the batik classification system can be done by integrating it into a mobile application. It will help individual users to easily classify and identify batik designs on-the-go. This aim can be achieved by developing a user-friendly interface that allows users to upload images of batik designs and receive a classification result using a mobile application with their devices. The system can also provide information about the batik art design, such as its origin and meaning to enhance the user experience.

The system can also be further improved by expanding the batik dataset. Batik designs from different countries can be included. It allows the model to learn and recognize a wider range of patterns and motifs. This can be achieved by collaborating with batik experts and collectors from various regions to gather a diverse set of images for the dataset.

REFERENCES

[1] J. Rodrigo, '3 Batik Types By Colours, Patterns, & Techniques In Malaysia', Eksentrika: Arts Community • Artist Registry • Asian Arts Magazine. Accessed: Jun. 06, 2023. [Online]. Available: https://www.eksentrika.com/batik-types-colours-patterns-technique-malaysia/

[2] S. I. Syed Shaharuddin *et al.*, 'A Review on the Malaysian and Indonesian Batik Production, Challenges, and Innovations in the 21st Century', *Sage Open*, vol. 11, no. 3, p. 215824402110401, Jul. 2021, doi: 10.1177/21582440211040128.

[3] S. Bin Silah, B. Isa, . Ponimin, and T. Karyono, 'Malaysian and Indonesian Batik: The Best Practices', *KnE Social Sciences*, pp. 47–55, Jun. 2021, doi: 10.18502/kss.v5i6.9177.

[4] S. I. Syed Shaharuddin *et al.*, 'A Review on the Malaysian and Indonesian Batik Production, Challenges, and Innovations in the 21st Century', *Sage Open*, vol. 11, no. 3, p. 215824402110401, Jul. 2021, doi: 10.1177/21582440211040128.

[5] I. M. A. Agastya and A. Setyanto, 'Classification of Indonesian Batik Using Deep Learning Techniques and Data Augmentation', in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, IEEE, Nov. 2018, pp. 27–31. doi: 10.1109/ICITISEE.2018.8720990.

[6] F. A. Putra *et al.*, 'Classification of Batik Authenticity Using Convolutional Neural Network Algorithm with Transfer Learning Method', in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, IEEE, Nov. 2021, pp. 1–6. doi: 10.1109/ICIC54025.2021.9632937.

[7] D. G. T. Meranggi, N. Yudistira, and Y. A. Sari, 'Batik Classification Using Convolutional Neural Network with Data Improvements', *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 1, p. 6, Mar. 2022, doi: 10.30630/joiv.6.1.716.

[8] I. Nurhaida, V. Ayumi, D. Fitrianah, R. A. M. Zen, H. Noprisson, and H. Wei, 'Implementation of deep neural networks (DNN) with batch normalization for batik pattern recognition', *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, p. 2045, Apr. 2020, doi: 10.11591/ijece.v10i2.pp2045-2053.

[9] B. S. Negara, E. Satria, S. Sanjaya, and D. R. Dwi Santoso, 'ResNet-50 for Classifying Indonesian Batik with Data Augmentation', in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, IEEE, Jul. 2021, pp. 1–4. doi: 10.1109/ICOTEN52080.2021.9493488.

[10] S. Joseph, I. Hipiny, H. Ujir, S. F. Samson Juan, and J.-L. Minoi, 'Performance evaluation of SIFT against common image deformations on iban plaited mat motif images', *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, p. 1470, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1470-1477.

[11] M. A. Rasyidi and T. Bariyah, 'Batik pattern recognition using convolutional neural network', *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1430–1437, Aug. 2020, doi: 10.11591/eei.v9i4.2385.

[12] M. A. Iqbal Hussain, B. Khan, Z. Wang, and S. Ding, 'Woven Fabric Pattern Recognition and Classification Based on Deep Convolutional Neural Networks', *Electronics (Basel)*, vol. 9, no. 6, p. 1048, Jun. 2020, doi: 10.3390/electronics9061048.

[13] M. A. Rauf, M. Jehanzeb, U. Ullah, U. Ali, M. Kashif, and M. Abdullah, 'Fabric Weave Pattern Recognition and Classification by Machine Learning', in *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, IEEE, May 2022, pp. 54–59. doi: 10.1109/SMARTTECH54121.2022.00026.

[14] A. Riski, E. B. Winata, and A. Kamsyakawuni, 'Pattern Recognition of Batik Madura Using Backpropagation Algorithm', 2022. doi: 10.2991/acsr.k.220202.044.

[15] W. Puarungroj and N. Boonsirisumpun, 'Convolutional Neural Network Models for HandWoven Fabric Motif Recognition', in *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, IEEE, Jan. 2019, pp. 300–303. doi: 10.1109/ECTI-NCON.2019.8692299.

[16] R. Andrian, M. A. Naufal, B. Hermanto, A. Junaidi, and F. R. Lumbanraja, 'k-Nearest Neighbor (k-NN) Classification for Recognition of the Batik Lampung Motifs', *J Phys Conf Ser*, vol. 1338, no. 1, p. 012061, Oct. 2019, doi: 10.1088/1742-6596/1338/1/012061.

[17] M. W. Ilahi, C. N. Apriyani, A. Desiani, N. Gofar, Y. Andriani, and M. R. Halim, 'Classification of Geometric Batik Motif Typical of Indonesian Using Convolutional Neural Network', *JURNAL TEKNIK INFORMATIKA*, vol. 15, no. 1, pp. 91–100, Jun. 2022, doi: 10.15408/jti.v15i1.24968.

[18] M. D. Firdaus and H. Nugroho, 'Classification of stamps and handmade batik based on pattern recognition', *J Phys Conf Ser*, vol. 1402, no. 6, p. 066053, Dec. 2019, doi: 10.1088/1742-6596/1402/6/066053.

[19] D. M. S. Arsa and A. A. N. H. Susila, 'VGG16 in Batik Classification based on Random Forest', in *2019 International Conference on Information Management and Technology (ICIMTech)*, IEEE, Aug. 2019, pp. 295–299. doi: 10.1109/ICIMTech.2019.8843844.

[20] L. Alzubaidi *et al.*, 'Review of deep learning: concepts, CNN architectures, challenges, applications, future directions', *J Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[21] M. Banoula, 'Machine learning steps: A complete guide', Simplilearn.

[22] R. Yamashita, M. Nishio, R. Do, and K. Togashi, 'Convolutional neural networks: an overview and application in radiology', *Insights Imaging*, vol. 9, Jun. 2018, doi: 10.1007/s13244-018-0639-9.

[23] M. Hussain, J. Bird, and D. Faria, 'A Study on CNN Transfer Learning for Image Classification', Jun. 2018.

[24] J. Wang, H. Zhu, S.-H. Wang, and Y.-D. Zhang, 'A Review of Deep Learning on Medical Image Analysis', *Mobile Networks and Applications*, vol. 26, no. 1/2021, pp. 351–380, 2020.

[25] W. Wang, Y. Hu, T. Zou, H. Liu, J. Wang, and X. Wang, 'A New Image Classification Approach via Improved MobileNet Models with Local Receptive Field Expansion in Shallow Layers', *Comput Intell Neurosci*, vol. 2020, pp. 1–10, Aug. 2020, doi: 10.1155/2020/8817849.

[26] R. O. Ogundokun, S. Misra, A. O. Akinrotimi, and H. Ogul, 'MobileNet-SVM: A Lightweight Deep Transfer Learning Model to Diagnose BCH Scans for IoMT-Based Imaging Sensors', *Sensors*, vol. 23, no. 2, p. 656, Jan. 2023, doi: 10.3390/s23020656.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[28] Y. LeCun, Y. Bengio, and G. Hinton, 'Deep learning', *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[29] T. S. S. J. Srilatha K. Vaidehi, 'Classification of Plastic and Non-Plastic Wastes Using Mobile net and SVM', *Mathematical Statistician and Engineering Applications*, vol. 71, no. 4, pp. 7267–7278, Dec. 2022, doi: 10.17762/msea.v71i4.1344.

[30] S. Srisuk, C. Suwannapong, S. Kitisriworapan, A. Kaewsong, and S. Ongkittikul, 'Performance Evaluation of Real-Time Object Detection Algorithms', in *2019 7th International Electrical Engineering Congress (iEECON)*, IEEE, Mar. 2019, pp. 1–4. doi: 10.1109/iEECON45304.2019.8938687.

[31] J. R. Terven and D.-M. Córdova-Esparza, 'A Comprehensive Review of YOLO: From YOLOv1 and Beyond', 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258823486

[32] R. M. Legino Art and D. UiTM, 'Malaysian Batik Sarongs: A Study of Tradition and Change', 2012.