DESIGN AND DEVELOPMENT OF A PRACTICAL MACROSCOPIC WOOD IDENTIFICATION SYSTEM USING DEEP LEARNING

TANG XIN JIE

MASTER OF ENGINEERING SCIENCE

LEE KONG CHIAN FACULTY OF ENGINEERING
SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN
APRIL 2019

DESIGN AND DEVELOPMENT OF A PRACTICAL MACROSCOPIC WOOD IDENTIFICATION SYSTEM USING DEEP LEARNING

By

TANG XIN JIE

A dissertation submitted to the
Lee Kong Chian Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Engineering Science
April 2019

ABSTRACT

Design and Development of a Practical Macroscopic Wood Identification System Using Deep Learning

Tang Xin Jie

Wood serves as raw material for countless industries due to its unique material characteristics. As such, different wood types are graded and valued accordingly based on their commercial value as raw material. Hence, wood identification is needed to ensure the correct wood type for usage. Macroscopic level wood identification that has been practiced by wood anatomists for decades can identify wood up to genus level for any commercial timber group. However, this knowledge is difficult to transfer to the industry non-experts. In this research, a rapid and robust macroscopic wood identification system is proposed using deep learning method with off-the-shelf smart-phone and retrofitted macro-lens as image acquisition device. Trained deep learning model is deployed as a cloud service accessible via Internet. This research collects and verifies data by wood anatomists on 100 Malaysian Tropical Timber types using the image acquisition device. A new Convolution Neural Network BlazeNet designed by the author, achieved better accuracy when benchmarked against SqueezeNet in this research. A cloud based wood identification system was deployed accompanied by an iOS application, Mywood-ID.

ACKNOWLEDGEMENT

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor and project principal researcher, Dr Tay Yong Haur for his invaluable advice, guidance and his enormous patience throughout the development of the research. Besides that, I appreciate the input from my co-supervisor Dr. Sugumaran a/l Nallusamy on thesis writing matter.

In addition, I would also like to express my gratitude to the project funder Forest Research Institute of Malaysia (FRIM) who has fully funded this project and provided greatest support throughout the project in particularly, our collaborators, Mr Lim Seng Choon and Dr Nordahlia Abdullah Siam from FRIM for their contributions to this research. Besides, thank for the support from my loving parents and friends who had helped and given me encouragement.

Lastly, I would like to extend my gratitude to my institute Universiti Tunku Abdul Rahman for providing me comfortable workspace and equipment.

APPROVAL SHEET

This dissertation entitled "**DESIGN AND DEVELOPMENT OF A PRACTICAL MACROSCOPIC WOOD IDENTIFICATION SYSTEM USING DEEP LEARNING**" was prepared by TANG XIN JIE and submitted as partial fulfillment of the requirements for the degree of Master of Engineering Science at Universiti Tunku Abdul Rahman.

Approved by:

____

(Dr. Sugumaran a/l Nallusamy)
Date: . . . .
Assistant Professor/Supervisor
Department of Internet Engineering and Computer Science
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

____

(Prof. Dr. Junichi Kishigami)
Date: . . . .
Adjunct Professor/Co-Supervisor
Department of Internet Engineering and Computer Science
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

FACULTY OF ENGINEERING SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date:

SUBMISSION OF DISSERTATION

It is hereby certified that ***Tang Xin Jie*** (ID No: ***16UEM01225***) has completed this dissertation entitled "*Design And Development of A Practical Macroscopic Wood Identification System Using Deep Learning*" under the supervision of Dr Sugumaran a/l Nallusamy (Supervisor) from the Department of Internet Engineering and Computer Science, Lee Kong Chian Faculty of Engineering and Science, and Prof. Dr Junichi Kishigami (Co-Supervisor)* from the Department of Internet Engineering and Computer Science, Lee Kong Chian Faculty of Engineering and Science.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Tang Xin Jie*)

*Delete whichever not applicable

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name TANG XIN JIE

Date

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figures **Page**

LIST OF SYMBOLS / ABBREVIATIONS

ConvNet	Convolution Neural Networks
FRIM	Forest Research Institute Malaysia
CITES	Convention on International Trade of Endangered Species of Wild Fauna and Flora
GLCM	Gray-level Co-occurrence Matrix
LBP	Local Binary Pattern
CLBP	Complete Local Binary Pattern
k-NN	K Nearest Neighbours
ANN	Artificial Neural Network
MLP	Multi-layer Perceptron
SGD	Stochastic Gradient Descent
GPU	Graphical Processing Unit
SqNet	SqueezeNet Convolution Neural Networks architecture
ResNet	Residual Network
BzNet	BlazeNet Convolution Neural Networks architecture
BzNet Colour	BlazeNet Colour Conversion Convolution Neural Networks architecture
BzNet Gray	BlazeNet Grayscale Convolution Neural Networks architecture
ReLU	Rectifier Linear Unit Activation Function
FLOPS	Floating Point Operations per Second
MB	Megabytes
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

CHAPTER 1

INTRODUCTION

1.1 Background

Wood, yielded by tree has been applied in many industries ranging from construction, paper making, furniture manufacturing and others. Different wood types are graded commercially based on material characteristics, demand from industries and rarity respectively. When trading is involved, the seller should declare the wood types and on the other hand, buyer has to verify the wood types to prevent fraudulent in trading. Therefore, there is need for wood identification to identify wood types.

Wood trading also gives rise to illegal logging and deforestation, endangering the world. Wood identification provides one of the most valuable supports in combating illegal logging and timber regulation. Wood identification in macroscopic as well as microscopic level are well established among wood anatomists through active research that spans over decades. Moreover, macroscopic level wood identification can identify wood up to genus level using 10-15x hand lens by Gasson (2011) and Koch *et al.* (2015). It might not able to identify up to species level as specified by CITES, however, macroscopic wood identification can serve as a first line of identification to fight against illegal logging and timber trade for law enforcement authority. The United Nations Office on Drugs and Crime published a practical guideline on Best Practice Guide for Forensic Timber Identification in 2016 shows the importance of wood identification as stated in Anonymous (2016).

Macroscopic level wood identification solves one of the problems faced by wood industry on identifying wood types and more crucially the commercial value. However, this knowledge is not easily transferable to the non-expert especially front line law enforcer as it takes special training to obtain proficiency in the knowledge. Generally, front line field screening of woods are done with hand lenses and keys, atlases of woods, or field manuals as in Henderson (1964), Ilic *et al.* (1991) and on International Trade in Endangered Species of Wild Fauna and Flora (2002). Such keys works on the basis of wood structure observed macroscopically shows abundant, characteristic distinctions typically allowing genus level identification. For more confident specificity identification, experts have to resolve to microscopic characters identification in the laboratory. In short, experts with hand lenses are still the state-of-the-art in the field.

Having stated that, macroscopic wood identification requires professional training in order for one to reach proficient level. The time and cost in establishing and maintaining this human-based domain knowledge, and the inconsistency of accuracy among those exercising such knowledge, signifies that this approach is limited in terms of scaling up to meet the ever-rising demands of wood experts. Forest Research Institute Malaysia (FRIM) offers charged professional wood identification services by Wood Anatomy Laboratory. A macroscopic hand lens with 10-15x magnification is used to examine the features on transverse cross section of wood which is the standard practices by wood anatomists as described in Anonymous (2009).

Computer vision has the potential to provide a practical and cost effective way to complement human-based domain knowledge for field identification of trading wood. One of the primary advantages of this potential is the reproducibility of robust identifications, independent of individual human training. Robustness and reproducibility are the keys in developing such system. Hence, sufficient images of the woods in question are required for training classifiers and such images should be captured in the field. Therefore, this motivated the author to create a practical automated wood types identification system that can transfer simple know-how in wood identification to industry.

In computer vision terms, the problem of image-based wood identification is one of texture-based image classification when the transverse cross section of wood is magnified to 10 times or higher. Computer vision techniques are used to extract the textural features from wood images. These features are then used to train a classifier using machine learning method. In recent years, deep learning methods are widely used in the field of pattern recognition with state-of-art performance either in supervised or unsupervised learning. Generally, deep learning required large datasets to train in order to achieve great performance. Therefore, this stresses the importance of large dataset in order to leverage the prowess of deep learning. Convolution Neural Network (ConvNet) which is a type of supervised deep learning method is an end-to-end method which consists of feature extractor and neural network classifier in an all-in-one architecture. Due to the end-to-end system wide design, ConvNet can be optimized to achieve great accuracy and outperform existing computer vision techniques.

In short introduction, this project as initiated by FRIM aims to build a mobile wood types identification system which works on macroscopic transverse cross section wood images. The overall idea to is replicate human wood anatomist with deep learning model hosted in the cloud as a service as illustrated in Figure 1.1.

Build upon the works of Siew *et al.* (2017) and Lee (2016), the author aims to improve system as in terms of classification accuracy and evaluate and deploy the system as a service. At current stage of the project, a server is hosted in the cloud with a deep learning classifier to serve mobile application as client. A simple overview of the architecture is shown in Figure 1.1, the front end application serves as a user interface to capture and upload macroscopic wood cross section images to the server. After classification, the server will then return the probabilities of wood types based on the uploaded wood type image. The mobile application is intended

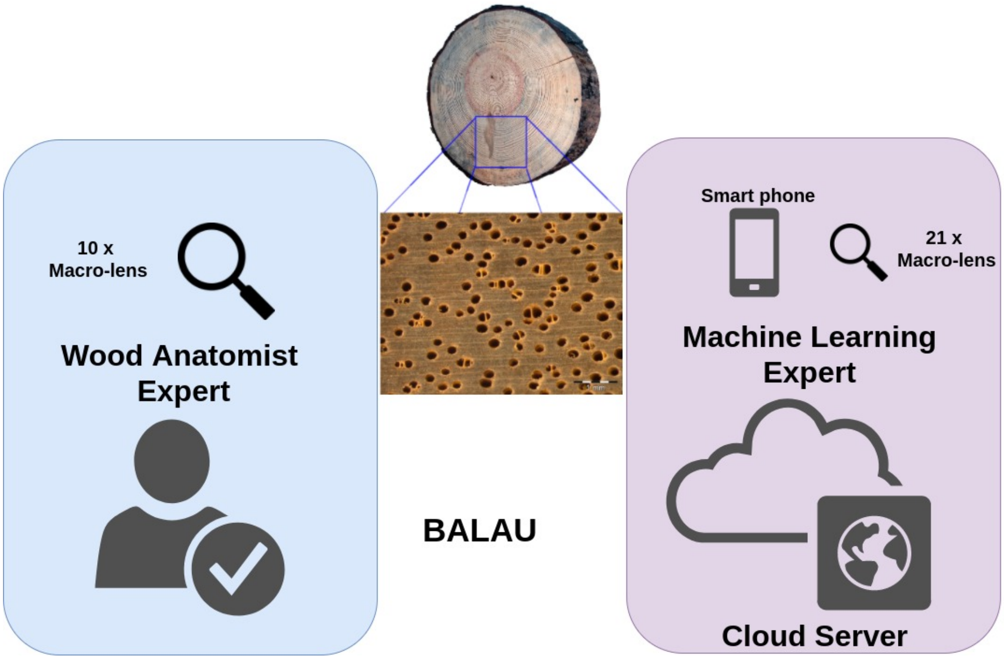


FIGURE 1.1: Overall project overview.

to be installed in smart-phone coupled with macroscopic lens. As for the server, a trained ConvNet classifier will be hosted to perform classification.

1.2 Problem Statement

In current practices, macroscopic level wood types identification relies on highly trained human experts and the availability of such expertise cannot cope with the demand of international interest in wood identification.

Wood types information is critical for the industry especially custom enforcer to tackle illegal wood log exportation. Current practice of human expert identification is slow, time consuming and non-reproducible as discussed in 1.1. Hence, this project aims to build a mobile wood types identification system that can cut down the identification time with close to human or better level of accuracy and reproducible, with great mobility and accessibility where users can use the system everywhere.

The main performance assessment criteria are accuracy in identifying wood types, the time of identification, the mobility of the system and the practicality of the system. Note that the system is built upon identifying macroscopic cross section wood type images as this replicates how wood anatomist identifies wood types. Several computer based wood types identification systems are developed such as the intelligence system developed by Khalid *et al.* (2008). The design used an industrial monochrome camera setup as image acquisition device to capture high quality monochrome macroscopic cross section wood types images in laboratory. This intelligence system lacks the mobility and accessibility as it limits the image acquisition process in the laboratory only.

Based on the work of Siew *et al.* (2017), deep learning method ConvNet outperformed the computer vision based methods as well as the state-of-the-art ConvNet method in terms of accuracy in classification. With the inclusion of dropout layer and patches prediction, Siew was able to leverage the disadvantage of lack of training data. This shows that deep learning method is suitable for wood types classification problem. However, the work was only focused on Forest Species Database Macroscopic (FSD-M) de Paula Filho *et al.* (2009) dataset which consists of high quality Brazilian wood types images which are acquired in the laboratory with industrial standard camera.

Further down the road, Lee (2016) used mobile smart-phone as the image acquisition device with retrofitted macroscopic lens. This method of image acquisition differs from laboratory practices where carefully prepared wood samples are captured using macroscopic camera under controlled environment. Using the trained classifier on laboratory image, the performance was disappointing as great accuracy could not be reproduced using smart-phone acquired image. The discrepancy of image quality between laboratory and smart-phone images rendered the laboratory image dataset insignificant for the new system. However, this quality discrepancy did not hinder the usage of smart-phone due to its mobility and accessibility. A detailed discussion of works by Siew *et al.* (2017) and Lee (2016) are written in Section 2.3.

As discussed in Section 1.1, deep learning method is data driven in order to achieve great overall performance. Since laboratory images were not transferable to the new system, there is a need to collect a huge dataset that consists of images of macroscopic transverse cross section of wood that are prepared using on-the-field method. The wood samples preparation method which will be discussed in Section 4.1.3 is the common in-situ practice by human expert in the field. The key idea is to collect data that closely resembles the field-deployable practices to ensure practicality.

Having said that, this image digitization process requires wood samples and human efforts from FRIM. To facilitate the data collection process, the author proposes to build a web based data collection mobile application with the aim of building a large-scale professionally annotated and verified wood types images. As a result, a digitized wood types database or digital Xylarium would be created at the end of data collection.

With large-scale data collected, the next challenge will be the architecture of ConvNet which is the core brain of system. An optimized ConvNet architecture can improve overall classification accuracy while maintaining small footprint of ConvNet. Therefore, the study into the disign and optimization of ConvNet to suit this problem is proposed by the author.

Based on the discussion above, the need to create a practical wood type identification system is summarized as follows:

1. There is a need to improve the classification performance by improving the ConvNet architecture.

2. There is a need to collect large scale data using the current mobile smartphone setup to train the classifier.

1.3 Research Question

The goal of this project is to develop a practical mobile wood identification system using deep learning method. The output of this project is expected to improve the classification accuracy of the wood type classifier. The general research question is as follows:

How to build a practical mobile macroscopic wood identification system using Convolution Neural Network?

The following are sub questions of the main research question:

1. What are the current issues faced by the mobile wood type identification system?
2. How to improve the accuracy and performance of the wood type identification system?
3. How to design a practical mobile wood type identification system and wood feature highlighting system using deep learning method?
4. How to validate the significance of proposed method on building a practical mobile wood type identification system?

1.4 Objective

This section discuss the main objectives for this study. The objectives are as follows:

1. To study and analyze the current issues related to mobile wood identification system using deep learning method.
2. To develop a practical wood identification service to identify macroscopic wood types by optimizing ConvNet architecture for classification.

3. To collect large-scale dataset for classifier and compile a digitized wood types database that consists of Malaysian Timber group.

4. To evaluate the performance of wood types classification.

1.5 Scope of Work

This section describes the scope of work for the project. The scope involves:

1. The development of data collection and verification application to facilitate data collection process. The data collector application is designed to be easy to use with simple annotation with continuous image capturing function. On the other hand, the verifier application is designed for human expert to verify the annotation of the collected data.

2. Designing and maintaining the wood type database $(DigitalXylarium)$ based on the data collected.

3. Compilation of dataset for training, evaluation and testing of classifier.

4. Optimize and improve the classification accuracy and the performance of wood type classifier.

5. Develop and improve the web based system architecture to ensure the scalability of the system as a service.

1.6 Significance of Study

The outcome of this research is expected to develop a practical mobile macroscopic wood type identification application that can be used to identify wood types with high accuracy, high mobility and practical to use. It should replicate or close to human accuracy in wood types identification within the seconds of time and can perform on-the-spot identification.

A new image acquisition method is introduced in this project which consists of mobile smart-phone and commercially available 21x magnification macroscopic lens. As opposed to customized imaging device, this setup offers lower cost of ownership to acquire macroscopic cross sectional of wood images.

Besides that, a digital Xylarium is created at the end of this project. Through the process the data collection for dataset compilation, at the same time, wood samples are digitized and stored in digital database for preservation and for future references. Digitization of the Xylarium provides a pathway to preserve invaluable wood samples and serves as a new communication medium to the research community which is beneficial to FRIM.

1.7 List of Publications

1. Tang, X.J., Tay, Y.H., Siam, N.A. and Lim, S.C., 2017. Rapid and Robust Automated Macroscopic Wood Identification System using Smartphone with Macro-lens.arXiv preprint arXiv:1709.08154.

2. Tang, X.J., Tay, Y.H., Siam, N.A. and Lim, S.C., 2018.MyWood-ID: Automated Macroscopic Wood Identification System using Smartphone and macro-lens.ACM International Conference Proceeding Series, pp. 37-43.

Publication 1 was submitted to The 9th Pacific Regional Wood Anatomy Conference (PRWAC 2017).

1.8 Dissertation Outline

The structure of the thesis is as follows:

- Chapter 2 reviews prior works in the related domain and their influence on the proposed design.
- Chapter 3 discusses theoretical background of the method used in this dissertation.
- Chapter 4 discusses the detailed methodology used in the research.
- The experimental results and analysis on datasets will be detailed in Chapter 5.
- Finally, the conclusion of the research and future works will be discussed in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

As discussed in Chapter 1 background study, the literature review will focus on macroscopic wood types identification problem in particularly identification on macroscopic transverse cross section of wood samples. When magnified to macroscopic level, wood structure observed shows abundant, characteristic distinctions in textural form as shown in Figure 2.1. At such, this type of identification is categorized under texture recognition problems due to the unique textural features found on the surface of macroscopic cross section of varying wood types. For texture recognition, there are two type of approaches for this problem namely handcrafted feature extraction using computer vision technique coupled with machine learning method for classification and deep learning method which is an end-to-end method.

This chapter covers review on computer vision methods and deep learning method in wood type identification in section 2.1 and section 2.2 respectively. Next, review of current and related work is detailed in section 2.3. The review of deep learning method covering convolution neural network is detailed in Chapter 3 for a in depth technical review.

2.1 Computer Vision Methods

For computer vision method, researchers design a feature extractor to extract the useful information that can describe or represent the texture of images. Encoded feature vectors are then used to train a machine learning classifier. For wood texture

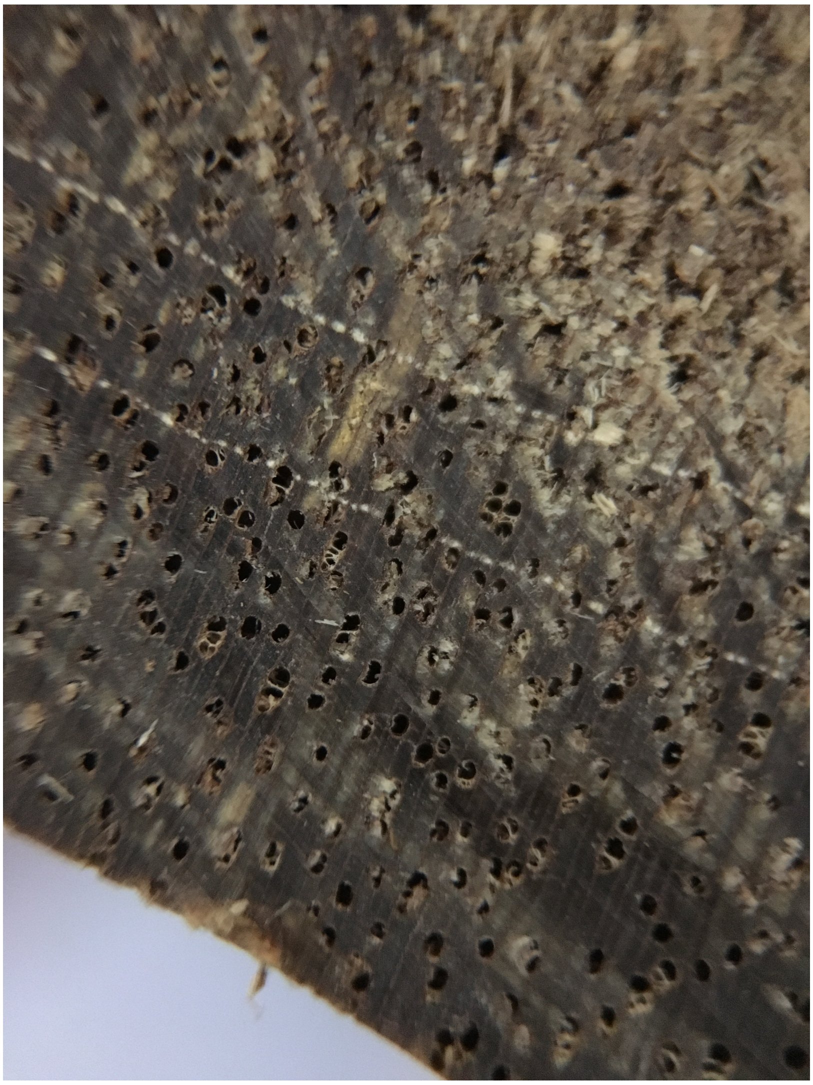


FIGURE 2.1: Example of textures observed at macroscopic level on transverse cross section of Dark Red Meranti captured using smart-phone with macro-lens.

classification, the following authors applied a few computer vision feature extractor for this problem. Tou *et al.* (2007) applied Gabor Filters and Gray-level Co-occurrence Matrix (GLCM). Besides that, Khalid *et al.* (2008) also applied GLCM in wood type classification. Nasirzadeh *et al.* (2010) used the Local Binary Pattern (LBP) to extract the intensity of the pixels from the images and classify them using k-nearest neighbour (k-NN) methods into different forest species. Besides that, Cavalin *et al.* (2013) proposed the use of a combined classifier, which combines feature extractor such as color-based feature, GCLM, LBP and Complete Local Binary Pattern (CLBP).

Computer vision approaches are a two-part system which consists of feature extractor and machine learning classifier as shown in the top part of Figure 2.2. The aim of the feature extractor is to extract interesting features from the images. The handcrafted features are then encoded or vectorized to represent the input image. The representations are then be used as training data for machine learning classifier for classification purpose. Commonly used machine learning classifiers are support vector machine (SVM), neural network, and clustering methods.

Researchers face two challenges which are the designs of feature extractor and machine learning classifier algorithm. These methods can perform theoretically with great accuracy but it may require great effort to design and optimize the two-part system. Using different feature extraction method might extract more unique and representative features but the next question lies ahead. Can the classifier able to classify them well? Classifier plays an important role in the overall identification problem and tuning and optimizing it is a great challenge.

Since the goal of the system is to have a high accuracy in identifying wood types, computer vision approaches impose high technical difficulties as researcher have to optimize the feature extractor as well as the classifier. Good performance can only be achieved when a good feature extractor meets a good classifier.

2.2 Deep Learning Methods

On the other hand, deep learning method is described as an end-to-end method where the feature extractor is combined with the classifier as a one-part architecture as shown in the bottom part of Figure 2.2. As such, deep learning method especially Convolution Neural Networks (ConvNets) have the ability to learn feature extraction and classification through supervised training from training data. As opposed to computer vision method, the designing work and the optimization of the system is now narrowed down to the design of convolution network architecture.

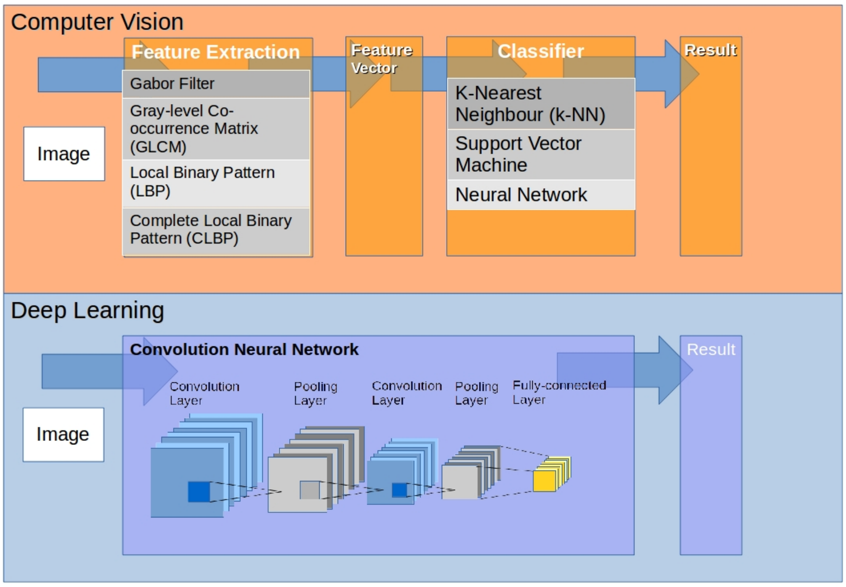


FIGURE 2.2: Comparison of Computer Vision Method (Top) and Deep Learning (Bottom) Design Pipeline

ConvNet approach excels in image classification problem. Hafemann *et al.* (2014) proposed using ConvNet on wood texture identification on Forest Species Database Macroscopic (FSD-M) dataset which consists of Brazilian wood types published by de Paula Filho *et al.* (2009). This paper showed that ConvNet outperformed the existing computer vision approaches on microscopic wood images with 97.32% accuracy by a margin of 4.32 % to the best computer vision method. A detailed comparison on classification between computer vision methods and ConvNet is shown in the paper. However, the method did not outperform the work by Cavalin *et al.* (2013) which combine multiple feature extractors for classification. Using multiple feature extractors increase the representation of the input images, hence it can improve the accuracy on classification. However, this approach increases the complexity of the system.

Table 2.1 extracted from Siew *et al.* (2017) shows the comparison in terms of accuracy in classification on computer vision based methods and deep learning methods. The work by Siew *et al.* (2017) using deep learning method registered the best accuracy. Deep learning method is easy to design and it outperforms the

TABLE 2.1: Classification on Macroscopic images dataset from Siew *et al.* (2017)

Features and Algorithm	Accuracy
Color-based features (SVM)	87.53%
Gabor Filters (SVM)	87.66%
ConvNet + Locally Connected Layers	95.77%
CLBP	96.22%
Multiple Classifiers (SVM)	97.77%
Proposed method (Improved ConvNet)	**98.16%**

existing computer vision based method.

2.3 Current and related work

Current works on wood types identification by Siew *et al.* (2017) and Lee (2016) shows good result using deep learning method. The work by Siew *et al.* (2017) showed that using Dropout Layer proposed by Srivastava *et al.* (2014) improved the classification accuracy of wood types on FSD-M dataset as shown in Table 2.1. Siew pointed out the issue of lack of training data adversely affected the accuracy of classification during cross validation because of over-fitting issue in ConvNet. To counter this problem, patches prediction as well as the inclusion of dropout layer were introduced to improve the generalization of model. Small patches were extracted from a input image and used for training. This significantly increases the size of training dataset.

For classification, a number of patches were extracted from the input image and classified by ConvNet. The patches predictions were then fused using Vote Rule to obtain the final prediction of wood type. This work shown that the macroscopic texture on wood types was local and abundantly repetitive among classes of wood types. However, this work focused on publicly available open-sourced FSD-M dataset which consists of high quality laboratory prepared macroscopic wood images. The method of laboratory image acquisition was described in de Paula Filho *et al.* (2009). The wood samples were carefully prepared to expose refreshed macroscopic transverse wood structure. Next, these features were then captured using macroscopic camera with controlled illumination.

Works by Lee (2016), as a continuation of Siew *et al.* (2017) work, on the other hand explores the practicality of developing and deploying ConvNet wood types classifier into a web based mobile service with the introduction of a mobile smart-phone image acquisition method. Mobile smart-phone with retrofitted macroscopic lens solved the mobility and accessibility problem as it lifted the constraint of location of identification where previously only limited in the laboratory with human wood experts.

Lee (2016) started his work by testing smart-phone acquired images with the classifier trained on laboratory images. The performance was however, disappointing as previous great accuracy was not reproducible with smart-phone images. Upon investigation, laboratory trained classifier could not cope with huge discrepancy of quality in images captured using smart-phone setup. In contrast to laboratory images, wood samples prepared on the field were rough with a simple cut using razor blade as depicted in Section 4.1.3. Hence, the wood structures exposed on macroscopic level were less visible. Besides that, structured illumination is not readily accessible in the field. As a result, smart-phone captured images that supposed to represent the images in-field demonstrated great variance in terms of exposure and most importantly the macroscopic visible wood structures. These discrepancies of quality of images obtained by mobile smart-phone camera as shown in Figure 2.3 hindered the accuracy of classifications substantially using laboratory trained classifier.

TABLE 2.2: Best classification accuracy from Lee (2016)

Top-N	Accuracy
Top-1	78%
Top-3	92%
Top-5	95%

Lee (2016) work showed that laboratory trained classifier were not transferable to adapt real world field images in a straightforward way. Besides that, Lee stressed out the problem of insufficient real world field data as captured using the smart-phone setup. Huge variation in image quality posed significant hindrance to laboratory trained classifier. Lee performed data augmentation to increase the data size of training data with hope to counter the deficit of insufficient training.

The best classification accuracy is shown in Table 2.2. However the best accuracy was still behind the achievement by Siew *et al.* (2017). This further strengthened the notion of inability of laboratory trained classifier in coping with real world on-field data. Besides that, there were limitation of data augmentation method to replicate the real world data in terms of variation. One way to remedy this hindrance is to collect large scale real world field images with much natural variations.

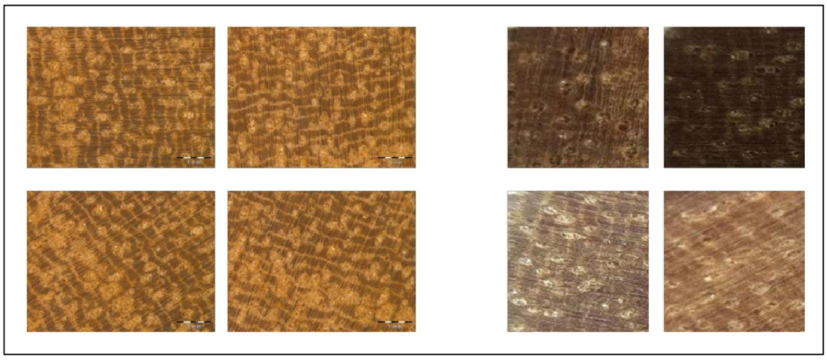


FIGURE 2.3: Laboratory (Left) and field (Right) images of Tembusu sample from Lee (2016)

In short summary, Siew explored the possibility of using ConvNet as opposed to computer vision based method for wood type identification. Built on his work, Lee explored the practicality of using mobile smart-phone with macroscopic lens as imaging device to perform real world wood type identification. Since deep learning is a data driven algorithm, both authors agreed on the lack of training data hindered the performance of classification.

2.3.1 Xylotron

XyloTron by Hermanson *et al.* (2013) from USDA Forest Products Laboratory provides a field deployable wood identification system that comprises of a customized camera with a computer running machine vision algorithm program as shown in Figure 2.4. These systems and approaches prove that computer-aided machine vision system is viable in macroscopic wood anatomy identification. A common characteristic between Khalid *et al.* (2008) system and XyloTron system is a specialized camera module with structured light source. This is to ensure the consistency of the image acquisition process to capture well-lit macroscopic wood image so that the machine vision model can identify the timber types correctly. However, this setup brought some drawbacks to the scalability of the application.

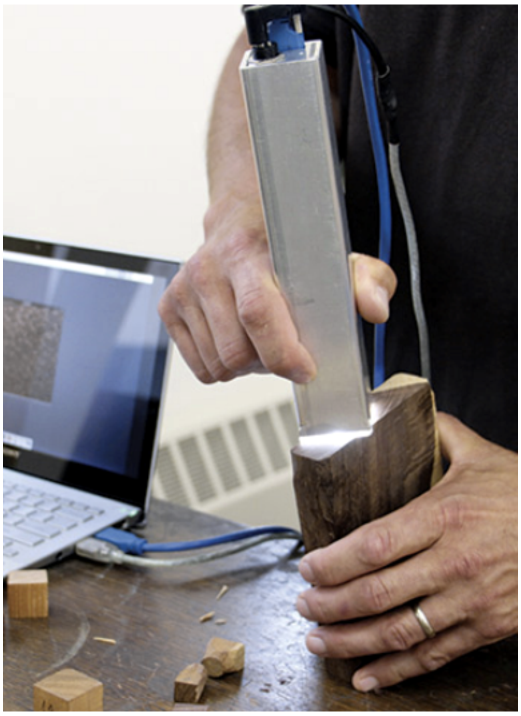


FIGURE 2.4: A demonstration of Xylotron device on wood identification

Firstly, Khalid *et al.* (2008) camera module was not meant to be portable as it was designed for laboratory. XyloTrons camera module solved the portability problem by custom fabricating a camera module with imaging sensors, controlled illumination light source and some connection ports to communicate with an external computer running the machine vision program. This setup is field-deploy ready but it involves some works to deploy it to frontline like the assembly of the camera module and installation of software.

2.3.2 Species Level Identification

Work by Ravindran *et al.* (2018) used ConvNet on classification of CITES-listed and other neotropical *Meliaceae* wood images. In this work, the author focused on 10-class species-level model on *Meliaceae* family as well as 6-class genus-level model. Patches of wood images similar to Siew *et al.* (2017) and image-level images were used to train a ConvNet with great performance achieving over 90% accuracy. The significance of this work showed that ConvNet was able to classify wood images down to species level albeit with lower accuracy than genus-level classification.

2.4 Residual Network and DenseNet

Residual Net (ResNet) by He *et al.* (2016) from Microsoft Research, winner of 2015 ImageNet Competition, introduced a new way of connection for ConvNet layer that offer skipping between layers. The intuition behind ResNet was that a layer should be able to fully represent its input and output under ideal condition. However, this is not true as the network goes deeper with layers where vanishing gradient problem arises. Hence, ResNet introduced the learning of residual function to mitigate this problem. As a result, ResNet architecture allows very deep ConvNet with 153 layers.

ResNet opened new door in skipping module when layers are allowed to bypass next layer and take a shortcut to next few layers. New architecture such as Highway Net by Srivastava *et al.* (2015), and DenseNet by Huang *et al.* (2017) were introduced.

DenseNet was built with Dense Block where any layer within the block are all directly connected to all subsequent layers. In other words, all subsequent layers receive feature maps from previous layers using concatenation methods. Dense connections between layers in Dense Block permits better information flows from back to front or from top to down in the network. This makes DenseNet easier to train as error can be back-propagated easily with better information flow. Besides that, the passing down of feature maps from preceding layers to subsequent layers can reduce redundancy in parameters and in turn reduces the size of network.

2.5 Summary of Review

As a summary of review, the lack of training data and the negligence of global context knowledge of classifier in wood types image reduced the performance of classifier based on the work of Lee (2016). Hence, the author proposes to collect large-scale real world data to enlarge the training dataset and remove patches generation. With great amount of data, the whole image is used for training so that the global context of the wood images is learned in order to improve the accuracy of the classifier and the practicality of the mobile wood identification system.

CHAPTER 3

THEORETICAL BACKGROUND

This chapter will discuss some fundamental concepts of Deep Learning that covers Artificial Neural Network, Convolution Neural Network (ConvNet) and some of the ConvNet design.

3.1 Artificial Neural Network

Artificial Neural Network (ANN), as its name implied, it is built with a network of artificial neurons that interconnect to each other. The idea behind ANN is to model biological nervous system, in particularly brain. In general consensus, ANN is deemed as a non-linear modeling tool that serves as a function approximation that maps the relationship from input to output data.

3.1.1 Artificial Neuron

Artificial Neuron, the fundamental building block of ANN, is modeled after a biological neuron that processes inputs as shown in Figure 3.1. It is modeled after a simple mathematical function as shown in Equation 3.1.

$$t = \sigma\left(\overrightarrow{w} \cdot \overrightarrow{x} + b\right) \tag{3.1}$$

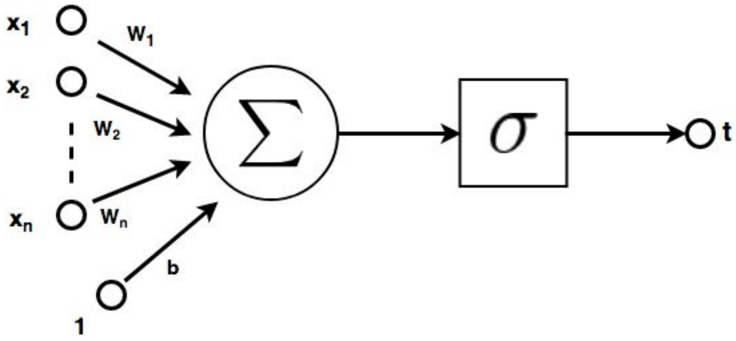


FIGURE 3.1: Artificial Neuron Concept

Based on Equation 3.1, $\overrightarrow{w}$ consists of *n* array of weights $\{w_1, w_2, \ldots, w_n\}$ and $\overrightarrow{x}$, contains *n* array of inputs denoted $\{x_1, x_2, \ldots, x_n\}$. Dot product of vector $\overrightarrow{w}$ and vector $\overrightarrow{x}$ is then summed with scalar bias, *b*. The output is passed to activation function denoted σ to get neuron output of *t*. The non-linearity of ANN is achieved by the activation function in most cases. Commonly used activation functions are defined as follows.

Activation Functions:

1. Hyperbolic tangent, *tanh*
2. Sigmoid
3. Softmax
4. Rectifier Linear Unit, ReLU

For this research, ReLU, defined in Equation 3.2 is chosen as the activation function for the deep learning ConvNet architecture for its high computation speed as verified by Krizhevsky *et al.* (2012).

$$\sigma(x) = max(0,\ x) \tag{3.2}$$

3.1.2 Multi-layer Perceptron

A neuron as discussed in 3.1.1 is simple function approximation tool with limited learning capacity. Adding more neurons and connecting them in layer fashion creates a more powerful modeling tool. Enter multi-layer perceptron, (MLP) as shown in Figure 3.2. With backpropagation algorithm, MLP can solve non-linear problems.

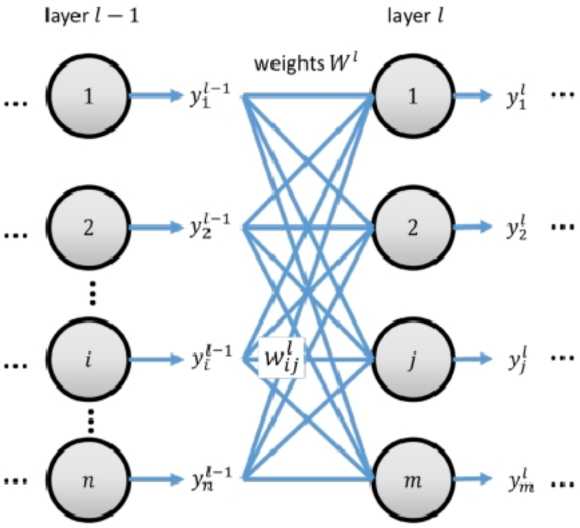


FIGURE 3.2: A model of multi-layer perceptrons

A simple and classic MLP consists of 3 layers which are input layer, a middle layer as known as hidden layer and output layer. The number of hidden layer determines the ability of a model in finding the relationship between inputs and output at higher degree.

In details, the *i* th output from layer $l - 1$ will multiply to their respective weight w_{ij}. Next, *i* th output are summed with respect to *j* th together with their biases in *j* th neurons. Lastly, the output of summation is then pass to activation function in layer *l*. The output is then pass to the next layer until it reaches the output layer.

3.1.3 Forward Propagation

Forward propagation is a process of computing the output of each node from the first hidden layer to the last output layer of multi-layer perceptrons. These computation refers applying activation functions on each node.

In this research, ReLU is chosen as the activation function for hidden layer. As for the output layer, softmax function is used instead which is defined in Equation 3.4. Softmax function is designed to solve multiclass classification problem. Based on Equation 3.3, assume that *L* th layer is the final layer and z_i^L is the output of a neuron in that layer. W_i^L and b_i^L are the weight matrix and biases of last layer, y_i^L is the output of *i* th neuron in the second to last layer. This equation is merely a neuronal function as defined in Equation 3.1. In Equation 3.4, y_i^L is the output of a neuron in output layer. The summation part in the denominator of Equation 3.4 represents the summation of every output of neurons in output layer. In a way, softmax function can be deemed as normalized output probability.

$$z_i^L = W_i^L y^{L-1} + b_i^L \tag{3.3}$$

$$y_i^L = \frac{e^{z_i^L}}{\sum_i e^{z_i^L}} \tag{3.4}$$

In short, the forward propagation is completed when the final outputs from output layer is obtained.

3.1.4 Training Objective

In machine learning, the core idea of training a model lies in the loss function denoted $\mathcal{L}$. In supervised training, $\mathcal{L}$ calculates the scalar distance of prediction output $\overrightarrow{y}^L$ and target output $\overrightarrow{t}$, denoted $\mathcal{L}\left(\overrightarrow{y}^L, \overrightarrow{t}\right)$, Each training samples is associated with a target, denoted $\left(\overrightarrow{x}, \overrightarrow{t}\right)$. In simple terms, this loss function measures the error made by the model with respect to the target.

In the case of supervised training, the training objective focuses on minimizing the sum of loss function applied to all training samples. Common loss functions used in Deep Learning community are defined as follows:

Loss Functions:

1. Square Error, Equation 3.5
2. Cross Entropy Error, Equation 3.6

$$\mathcal{L}\left(\overrightarrow{y}^L, \overrightarrow{t}\right) = \frac{1}{2}\sum_c \left(y_c^L - t_c\right)^2 \tag{3.5}$$

$$\mathcal{L}\left(\overrightarrow{y}^L, \overrightarrow{t}\right) = -\sum_c \left(t_c \log y_c^L\right)^2 \tag{3.6}$$

In Equation 3.5 and Equation 3.6, y_c^L refers to output of neuron *c* in the last layer of MLP, and t_c is the identity function applied to the ground truth *t* defined in Equation 3.7.

$$f\left(x\right) = \begin{cases} 1, & t = c \\ 0, & otherwise \end{cases} \tag{3.7}$$

3.1.5 Back Propagation

MLP permits higher capacity in modeling complex relationship between inputs and outputs by increasing higher number of hidden layer. However, there is a problem in training the model effectively. Rumelhart *et al.* (1986) proposed back propagation algorithm in a MLP. This algorithm provides an effective method to update parameters in the model iteratively and enables the model to generalize to any non-linear activation function.

The working principle of back propagation depends on calculating the derivatives of loss function or error with respect to model weights. This error derivatives is propagated back from the last layer to the previous layer, all the way to the first input layer, one layer at a time. It travels the opposite direction of forward propagation and hence the name back propagation. The relationship of error with respect to weights gives the model an idea of how the weights affect the error in the loss function. Therefore, the model now can adjust weights more effectively that can lead to minimization of error and this is called Gradient-based learning.

3.1.6 Training Algorithm

Back propagation in Section 3.1.5 gives the model an idea on how weights affecting the final loss. Training algorithm solves the problem of updating the model's parameters efficiently and effectively to achieve faster convergence of error. The most popular training algorithm used is Stochastic Gradient Descent (SGD) algorithm. The generalized algorithm as defined by Duda *et al.* (1973) is described below in high level.

```
Algorithm: Stochastic Gradient Descent
Begin
while not Convergence Criteria() do
    x_batch, y_batch ← next batch_size examples from (X, y)
    network_state ← ForwardProp(W, x_batch)
    $W_{grad}$ ← BackProp(network_state, y_batch)
    Δ W ← $-\sigma W_{grad}$
    W ← W + $\Delta W$
end
```

Algorithm 1: Stochastic Gradient Descent

The inputs are as follows:

1. *W*, weights and biases of model
2. (*X*, *y*), dataset and batch size, the number of training sample in each iteration
3. σ, learning rate

For simplicity in notation, model parameter of weights and biases is denoted as *W*. In practical implementations, each layer is defined in 2-dimensional matrix of weights and 1-dimensional biases.

In simple terms, SGD iterates over a mini-batch of dataset at each iteration instead of the full batch of dataset. The iteration starts by sampling a mini-batch of dataset determined the batch size, performs forward propagation followed by back propagation to obtain the derivatives of loss function. Then, these derivatives are used to update the weights and the iteration is finished. Next mini-batch is generated and the process repeats until convergence criteria is met.

Most common convergence criteria are as follows:

1. Maximum epoch, number of times the whole training dataset is iterated over
2. Desired value of cost function is met
3. Loss function shows no improvement after few epochs

In practice, one potential drawback of SGD is that the occurrence of oscillations in gradient due to not all examples are taken into account in each weights update. This may cause slow convergence and potential fatal explosion of gradient, i.e divergence of gradient. One workaround strategy is to introduce momentum terms to weight updates equation as shown in Equation 3.8.

$$\Delta w_{ij}^{(j)} = -\sigma \frac{\delta E(t)}{\delta w_i} + \beta \Delta w_{ij}^{(t-1)} \tag{3.8}$$

The idea to take the running average of derivatives by joining the previous update $\Delta w_{ij}^{(t-1)}$ into current update iteration. β represents momentum usually in the range of 0.5 - 1.0 controls the amount of previous weight updates in the current iteration. This can help smoothen the oscillation of gradient introduced by SGD.

3.2 Deep Learning

Deep learning is a branch of machine learning that consists of model with huge number of hierarchical hidden layer, hence the name deep. The branch of research sees great implementation in solving pattern recognition, optimization and graphical representation problem. The abundance of data and the accessible to high performance graphical processing unit (GPU) drives the development of Deep Learning at exponential pace.

The success of back propagation implementation revitalized the machine learning research back in 1980s. At this time, machine learning model such as MLP with one hidden layer, Support Vector Machine, and etc. were proposed. However, these models were considered shallow due to low number of hidden layer or node.

Fast forward to 2006, Hinton *et al.* (2006) success in training a Restricted Boltzmann Machine, RBM shift the focus of machine learning research to Deep

Learning. RBM is a generative model that learns the pattern of input data in unsupervised training. The weights learned are then used to initialized a neural network for classification or feature extraction purposes. At the same time, auto-encoders by Bengio *et al.* (2007) identified that training can be done at each layer in a greedy layer-wise format.

In the interest of this research, Convolution Neural Network (ConvNet) is widely applied in image classification problem, popularized by Krizhevsky *et al.* (2012) in the ImageNet competition.

3.2.1 Convolution Neural Network

ConvNet was first proposed by LeCun *et al.* (1989) for handwritten zip code recognition. ConvNet, is a type of feed forward neural networks that is built with additional convolution and pooling layers which have these three characteristics: local receptive fields, shared weights and spatial or temporal sub-sampling LeCun *et al.* (1989). At that time, ConvNet is the first truly successful in training multi-layer network. This is due to the weights sharing feature that reduces the number of parameters as opposed to MLP with the same number of hidden layer.

However, the development of ConvNet was not popularized since then due to limitation of computing power and lack of big scale dataset. Breakthrough came from AlexNet by Krizhevsky *et al.* (2012) and large scale dataset like Imagenet by Deng *et al.* (2009). AlexNet shown that the ability of 7-layer ConvNet in classifying 1000 classes of images.

Since then, ConvNet has become the go-to tools in solving problems for image classification, object detection and face recognition with good performance. The trend of development of ConvNet focused on building larger and deeper ConvNet architecture to improve the accuracy of classification. However, deeper ConvNet has substantially more parameters to train which translate to more training time and possible over-fitting issue due to limited dataset.

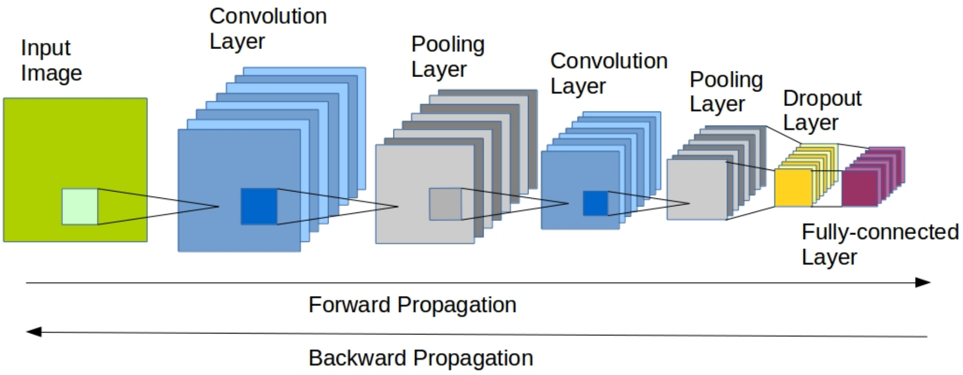


FIGURE 3.3: A generic ConvNet design composed of convolution layer, pooling layer and fully connected layer.

Figure 3.3 shows a generic design of ConvNet solving image classification problem which consists of convolution layer, pooling layer, fully connected layer. These layers are discussed in the following sections.

3.2.1.1 Convolution Layer

Convolution layer uses the convolution technique in computer vision to convolve learn-able filters with input images to produce feature maps. Each neuron in the filter is not fully connected to the previous layer. The main benefits of using convolution layers is that, it reduces the computation complexity and time compared to fully connected layers. In short, these data-driven convolution filters are tasked to extract interesting features.

The formulation of 2-dimensional Convolution layer is defined in Equation 3.9.

$$y_{rc}^{l} = \sigma\left(\sum_{i=1}^{F_r}\sum_{j=1}^{F_c} y_{(r+i-1)(c+j-1)}^{l-1} w_{ij}^{l} + b^{l}\right) \tag{3.9}$$

where

- y^l_{rc}, the output unit at row, r, column c
- F_r and F_c are the number of rows and columns in the 2-dimensional filter
- w^l_{ij}, value of filter at position i, j
- $y^{l-1}_{(r+i-1)(c+j-1)}$, value of input to this layer at position $r + i - 1$ and $c + j - 1$
- b^t, bias term

3.2.1.2 Pooling Layer

Features that are extracted by convolution layer is ready to be passed to classification layer at least in classical machine learning approach. However, these features remained huge in number which in turn results high number of parameter if connected in fully connected fashion. A down-sampling method is needed to reduce the number of parameter. Enter pooling layer.

Pooling layer performs down-sampling of data and it is usually connected after convolution layer. The main intuition of pooling layer is from the observation of the static property of image, which means that a feature extracted by one neuron might be extracted by another neuron. In other words, same extracted feature can be aggregated by statistical method such as averaging the feature. This aggregation method is called pooling, the most commonly used pooling methods are listed as follows:

- Average Pooling
- Max Pooling

$$y^l_{rc} = max_{i,j\epsilon[0,1,\ldots,m]} y^{l-1}_{(r+i-1)(c+j-1)} \tag{3.10}$$

where

- y^l_{rc}, the output unit at row, *r*, column c
- $y^{l-1}_{(r+i-1)(c+j-1)}$, value of input to this layer at position $r + i - 1$ and $c + j - 1$

The formulation of max pooling is defined in Equation 3.10. It reduces the parameters from the input layer and summarizes the feature responses. Together with convolution layer and fully connected layer, the input data is transformed from a 3-dimensional image to a 1-dimensional vector consists of the probabilities of each class.

3.2.1.3 Fire Module

Figure 3.3 shows a generic and simple design of ConvNet. SqueezeNet from Iandola *et al.* (2016), on the other hand proposed a ConvNet that is 50 times smaller than AlexNet in terms of parameters while achieving similar accuracy in classification.

The core of SqueezeNet is the Fire Module that comprises of 1x1 squeeze convolution layer followed by 1x1 and 3x3 expand convolution layer. 8 Fire Modules were used in SqueezeNet model with max pooling layer and global average layer.

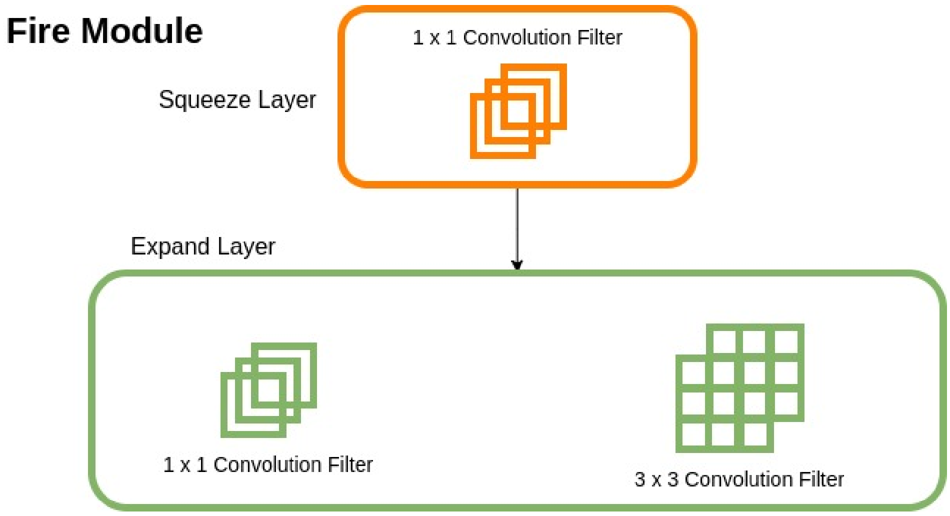


FIGURE 3.4: Fire Module of SqueezeNet comprises Squeeze Layer and Expand Layer.

Figure 3.4 shows the Fire Module as defined in SqNet. This module consists of 2 layers namely Squeeze Layer and Expand Layer. Only $1x1$ convolution filter Is used in Squeeze Layer with reduced number of output from previous layer. Next, Expand Layer which is connected after Squeeze Layer consists of $1x1$ convolution filter and $3x3$ convolution filters. There is a total of 8 Fire Modules as defined in SqNet.

The core idea of reducing parameters for SqNet is achieved through Squeeze Layer in Fire module where the number of output feature maps are reduced in comparison to the input feature maps. The usage of $1x1$ convolution filter with rectifier activation function (ReLU) selects a limited number of the best feature maps from input. Next, the Expand layer increase the number of feature maps with 1 x 1 and 3 x 3 convolution filters. For $3x3$ filters, it is more like conventional convolution filters that learn the abstraction of the previous Squeeze layer. Note that padding of 1 is used in 3 x 3 filters to ensure the output feature maps of Expand layer have the same dimensions.

CHAPTER 4

METHODOLOGY

This chapter discusses the methodology of this research. This research used deep learning model in particularly ConvNet to recognize 100 Malaysian Tropical timber types. Supervised learning was applied where data in the form of RGB colour images with accurate and verified annotation were used to train the deep learning model. Hence, the main research focused on the development and design of ConvNet classifier. Therefore, the procedure of the research is defined as follows:

1. Collect and verify data
2. Compile dataset for training and validation
3. Design ConvNet architecture
4. Train and validate classifier
5. Evaluate the performance of classifier

Each procedure above is discussed in detail in the following sections.

4.1 Data Collection and Verification

This section discusses the type of data collected and the verification process of data.

4.1.1 Macroscopic Cross-sectional Wood Types Images

The data collected for macroscopic wood identification was cross-sectional macroscopic wood images in colour RGB format. The cross section of wood exposes unique macroscopic feature which allows human anatomist to identify the wood type up to genus level with relatively high accuracy without resolving down to microscopic level. A hand lens with macroscopic magnification magnifies the wood feature for human identification. This practice is well established among wood anatomist as described in Gasson (2011).

4.1.2 Image Acquisition Device

The main function of an image acquisition device is to mimic human anatomist on identifying wood type at macroscopic level. Hence, it comprised of a smart-phone with retrofitted macroscopic lens of 20 x magnification as depicted in Figure 4.1. The goal of the data collection was to collect real-world field images captured by the image acquisition device. The key specification of image acquisition device is tabulated in Table 4.1.

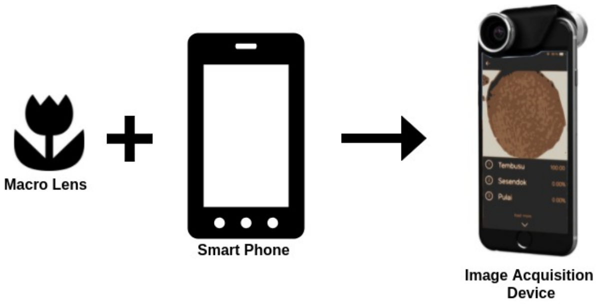
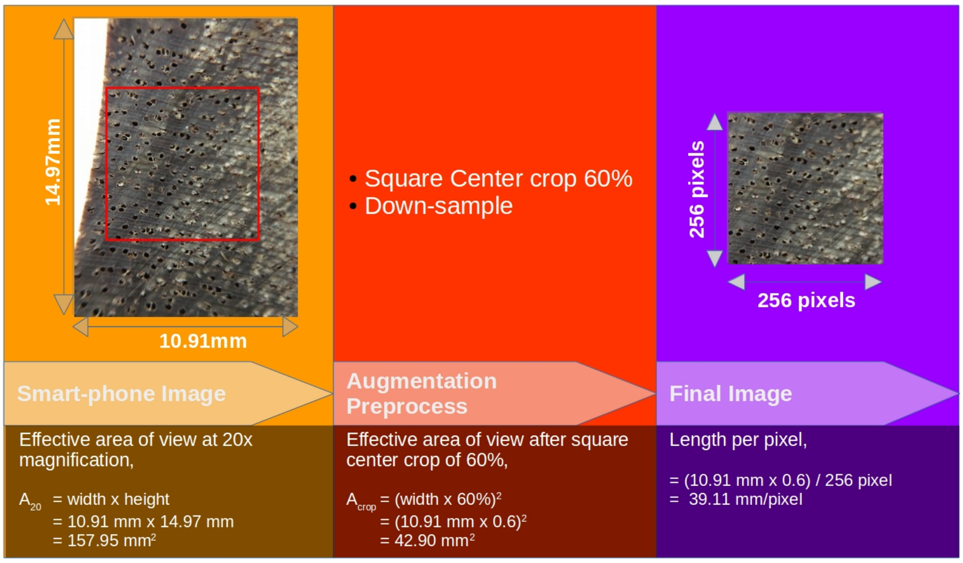


FIGURE 4.1: Image Acquisition Device comprised of smart-phone and 20x macro lens.

Smart-phone comes with varying camera resolutions ranging from 8 Megapixels to 20 Megapixels and above. However, the optical focal length of these camera

TABLE 4.1: Table of image acquisition device specification

Device	iPhone 6	iPhone 6s
Manufacturer	Apple	Apple
Camera Resolution (megapixel)	8.0	12.0
Focal Length (mm)	28.0	28.0
Macro lens magnification (times)	20	20
Effective area of view after magnification (mm^2)	157.95	157.95



Effective area of view at 20x magnification,

A_{20} = width x height
= 10.91 mm x 14.97 mm
= 157.95 mm^2

Effective area of view after square center crop of 60%,

A_{crop} = (width x 60%)^2
= (10.91 mm x 0.6)^2
= 42.90 mm^2

Length per pixel,

= (10.91 mm x 0.6) / 256 pixel
= 39.11 mm/pixel

FIGURE 4.2: Image captured by smart-phone with macro lens and the flow of preprocessing.

commonly falls in the range of averagely 28mm. This observation allows us to assume that the actual area of views after 20 times magnification of different smartphone camera are marginally similar. In fact, this assumption held true as we used smart-phone with different camera module of 8 Megapixel and 12 Megapixel for development. As per Fig. 4.16, the actual area of tissue captured after magnification is $157.95mm^2$ with 2% variation between different camera module. A squared center portion of the image were cropped and down-sampled to 256 pixel $\times$ 256 pixel after preprocessing which represents the actual area of tissue is $42.90mm^2$.

4.1.3 Wood Specimen Preparation Prior to Data Collection

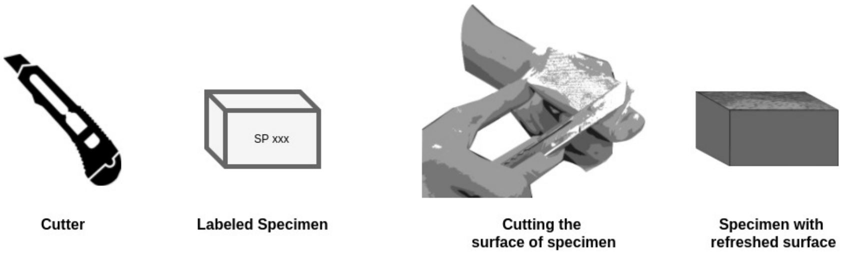

SP xxx
Cutter
Labeled Specimen
Cutting the surface of specimen
Specimen with refreshed surface

FIGURE 4.3: The process of preparation of wood specimen to expose macroscopic feature prior to data collection.

The flow of data collection starts as follows. Firstly, the wood specimens were prepared as shown in Figure 4.3. The cross-section of the wood specimen was cut using a razor cutter to refresh the surface of the specimen. This is a standard practice among wood anatomist to expose a clean and refreshed wood features for identification. With a proper refreshed surface, the specimen was ready for data collection.

4.1.4 Data Collection and Verification

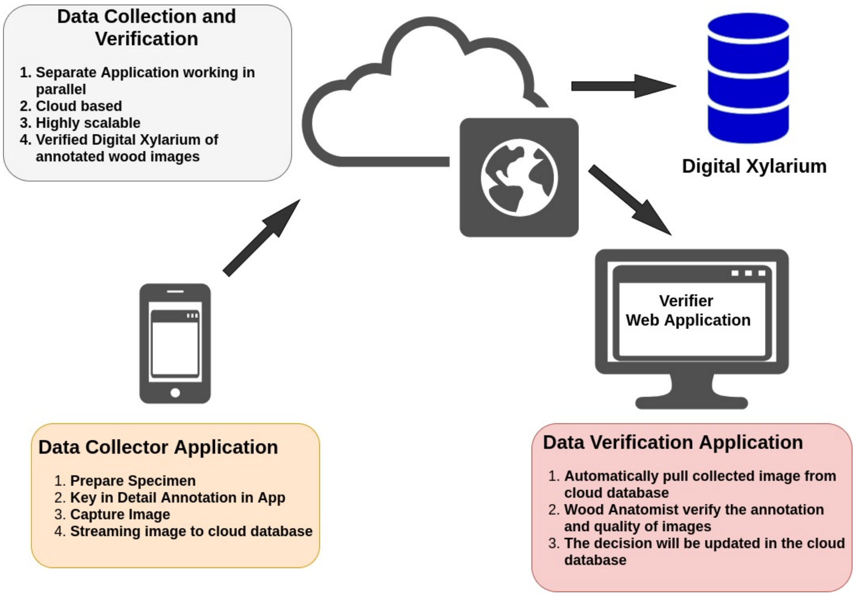

Data Collection and Verification
1. Separate Application working in parallel
2. Cloud based
3. Highly scalable
4. Verified Digital Xylarium of annotated wood images
Digital Xylarium
Verifier Web Application
Data Collector Application
1. Prepare Specimen
2. Key in Detail Annotation in App
3. Capture Image
4. Streaming image to cloud database
Data Verification Application
1. Automatically pull collected image from cloud database
2. Wood Anatomist verify the annotation and quality of images
3. The decision will be updated in the cloud database

FIGURE 4.4: The data collection and verification process using cloud based application.

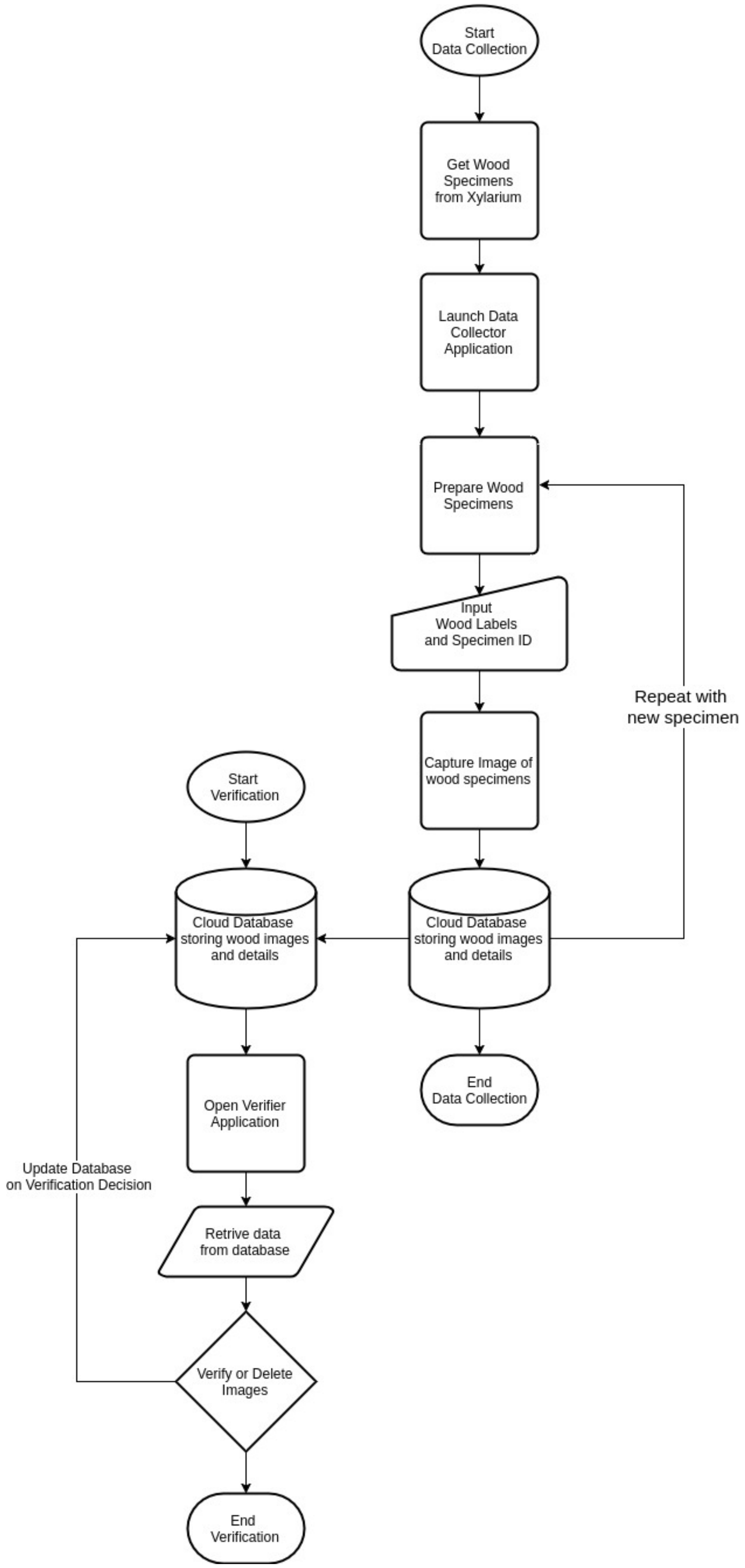

Start
Data Collection
Get Wood Specimens from Xylarium
Launch Data Collector Application
Prepare Wood Specimens
Input Wood Labels and Specimen ID
Capture Image of wood specimens
Cloud Database storing wood images and details
End Data Collection
Repeat with new specimen
Start Verification
Cloud Database storing wood images and details
Open Verifier Application
Retrive data from database
Verify or Delete Images
End Verification
Update Database on Verification Decision

FIGURE 4.5: The flow chart of data collection and verification

The overall architecture of data collection and verification is shown in Figure 4.4 and the process flow chart is shown in Figure 4.5.

To create an efficient and fluid data collection process, 2 cloud-based applications were developed namely Data Collector and Data Verifier. Data Collector, a mobile application was developed to run on image acquisition device, which is a smart-phone as shown in Figure 4.1. This application was a streamlined image annotation and image uploading application. On the other hand, the counterpart Data Verifier application was designed as a web-application to retrieve collected data from cloud database and let user, particularly wood anatomist to verify or delete images based on the quality and annotation accuracy.

The flow of data collection started from getting the uniquely labeled wood specimens from Wood Xylarium, a library that stored wood specimens in a controlled environment. Next, user had to prepare the image acquisition device by retrofitting macro lens and launched Data Collector application. Wood specimens were needed to be prepared as described in Figure 4.3 to expose clean and refreshed macroscopic wood features on the cross section of specimens. After that, user needed to key in the wood labels and specimen ID in the Data Collector input fields. With correct annotations, user now can start to capture multiple images on the prepared specimen with varying rotations, and lighting conditions. Captured and annotated images were then uploaded to cloud database for storage. The cycle continued when new specimens are prepared.

After kick-starting data collection process, wood anatomist on the other hand started verifying collected data. For data verification, user needed to use the Data Verifier web-application that is designed to run on desktop web browser. The process started by retrieving collected images and annotations from cloud database. Wood Anatomists had to decide between verifying and deleting images based on image quality and accuracy of annotations. Blurry images or incorrect annotations would result images being deleted. Decisions were sent to cloud database and data entries would be updated accordingly. The process ended when there are no unverified images. All in all, the flow chart shows that data collection and data verification run in parallel and in phases.

The data collection was carried out in phases over the span of 3 months. The data collection team was assigned to collect data on each wood types with all available specimens from the Xylarium within a given timeframe. On average, the team has to collect data for 33 wood types for every month. In the parallel timeline, data verification team formed by human anatomists were verifying the data collected by data collection team on the validity of specimens ID, class labels and image quality. Together, the data collection team and data verification team worked hand in hand in preparing dataset for training

The commercial value of wood types determined the order of data collection where highly commercialized wood types were collected first. As result, at the end of first month of data collection, a 20-class dataset was generated for training. Subsequently, 60-class and ultimately the 100-class were generated in the second and third months of data collection. Therefore, the research progressed in 3 phases where ConvNets were trained on dateset of 20-class, 60-class and 100-class followed by analysis.

4.2 Dataset Compilation

As per Figure 4.4, a digital Xylarium, digital wood library was created with the collected and fully annotated wood images at the end of data collection and verification. A total of 101,546 images from 1,919 specimens for 100 classes were collected as notated in Appendix A.

To ensure the generalization of a classifier, it is important to cross validate training and validation data. Hence, the dataset should be separated in such a way that the validation data is unseen by the classifier from training. For the dataset generation, the data was separated by the specimen ID. 5 random chosen specimen IDs from each wood type are used for validation dataset where the remaining specimen IDs are used for training. This cross-validation setup was to ensure the classifier would not over-fit to the training specimens and it was a good measure on the performance to unseen specimens in the field.

The next step was to compile collected images into datasets for training and validation. For cross-validation setup, images of wood types were separated based on their specimen ID. This was to ensure images from training specimens are unseen to validation specimens.

After cross-validation setup of wood images, these images were then augmented and balanced using methods as described in the following list. The details are discussed in the following subsections.

4.2.1 Cross validation

This subsection discusses the steps of cross validation in separating the data into training and validation datasets.

Cross validation procedure:

1. Retrieve verified data from database for specimen's IDs.

2. Random choose a maximum of 5 IDs for validation.

3. Remaining IDs are set for training

4. Update database on separated specimens' IDs for training and validation

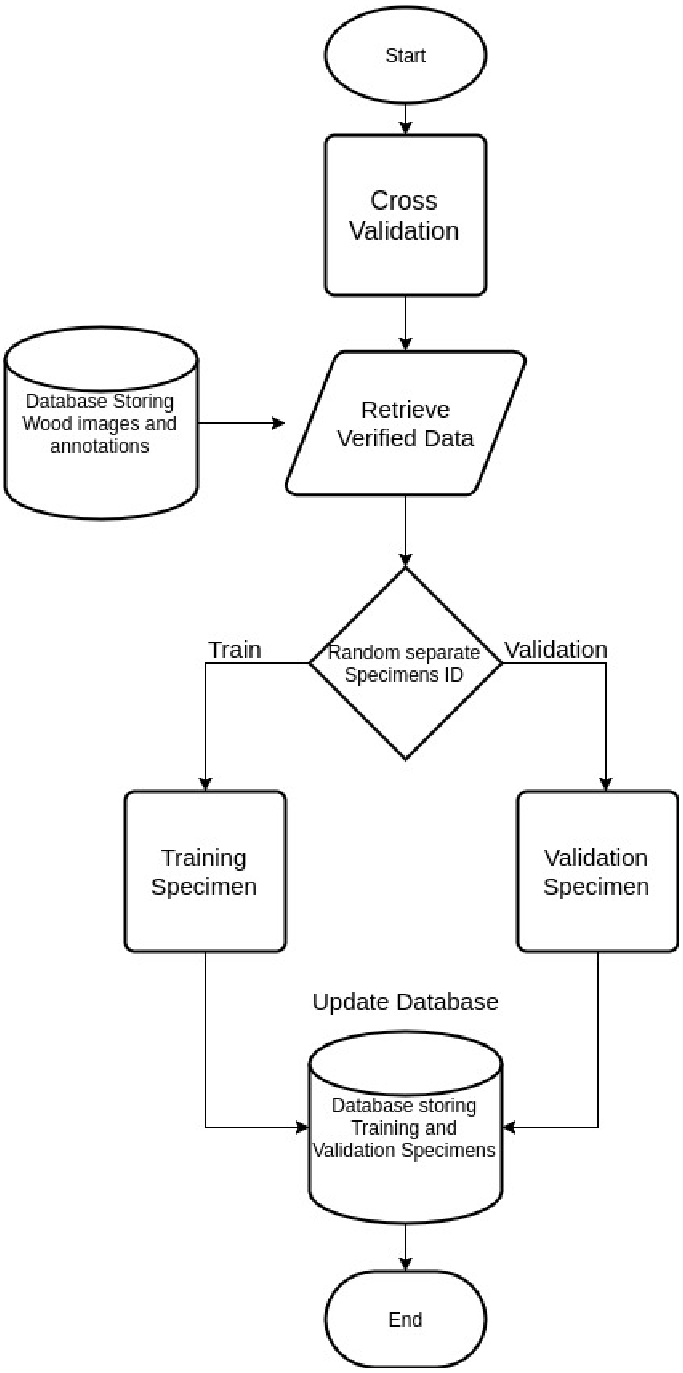

Start
Cross Validation
Database Storing Wood images and annotations
Retrieve Verified Data
Train
Random separate Specimens ID
Validation
Training Specimen
Validation Specimen
Update Database
Database storing Training and Validation Specimens
End

FIGURE 4.6: The flow chart for cross validation on dataset

Figure 4.6 shows the flow on separating dataset for cross validation purpose. This process was only applied to verified data by human anatomists after the completion of data verification process. A query was made to the database to get list of specimen's IDs for each wood types. After that, a maximum of 5 specimen's IDs were randomly chosen for validation while the remaining IDs were set for training. Next, the separated IDs were then updated to the database for future dataset generation.

Referring to Appendix A, averagely there were 681 images and 335 images allocated for training and validation for each class. On the specimen's side, there were about 15 and 5 specimens for training and validation.

4.2.2 Augmentation

The idea of augmentation is to increase size of dataset and balance number of images per class. Due to the capacity of ConvNet, performance is dependent to the size of dataset. Hence, a series of augmentation methods were used to increase the size of dataset.

The selection of augmentation method was based on the assumption to improve the robustness of classifier on varying conditions of use such as different lighting, limited focused image, and angle of orientation.

The augmentations were categorized into offline and online augmentations. For offline augmentation, images were augmented and store in database whereas online augmentation refered to augmentation applies to image during training of the classifier.

4.2.2.1 Offline Augmentation

Offline random augmentation methods:

1. Rotation augmentation
2. Brightness augmentation
3. Rim-blurring augmentation
4. Field of view resizing augmentation

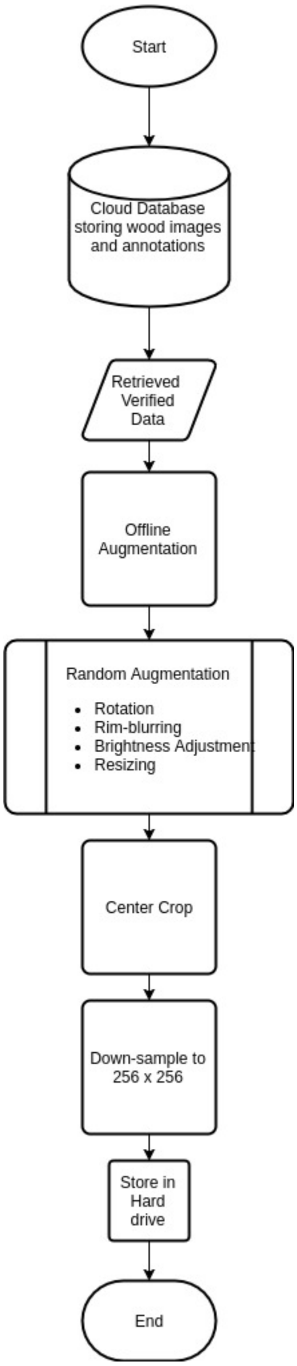
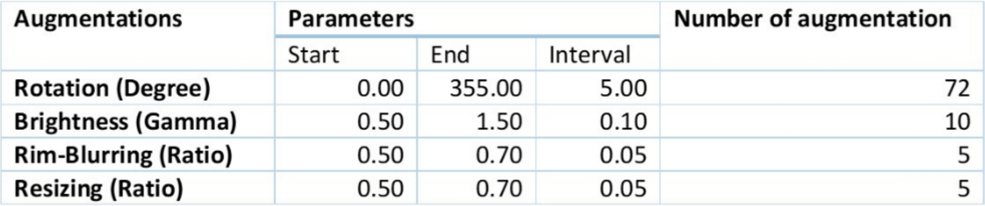


FIGURE 4.7: Flow chart for offline augmentation

Figure 4.7 shows the flow of offline augmentation. At the beginning, images were retrieved from cloud database as used in data collection and data verification. Note, augmentations only worked on verified images. Images were then randomly augmented by a combination of augmentation methods, namely rotation augmentation, rim-blurring augmentation, brightness adjustment augmentation and field of view resizing augmentation. After random augmentations, the center-square of image was then cropped followed by down-sampling to 256×256 pixels. Lastly, augmented images were stored in local hard drive and this signified the end of offline augmentation process.

The reason of down-sampling to 256×256 pixels for width and height was to follow ImageNet dataset standard by Deng *et al.* (2009) as most ConvNet architectures were designed to take 256×256 images.



Augmentations	Parameters			Number of augmentation
	Start	End	Interval	
Rotation (Degree)	0.00	355.00	5.00	72
Brightness (Gamma)	0.50	1.50	0.10	10
Rim-Blurring (Ratio)	0.50	0.70	0.05	5
Resizing (Ratio)	0.50	0.70	0.05	5

TABLE 4.2: Table of parameters range for offline augmentations

Table 4.2 shows the range of parameters for offline augmentations and these augmentations will be discussed in the following sunsections.

4.2.2.2 Rotation Augmentation

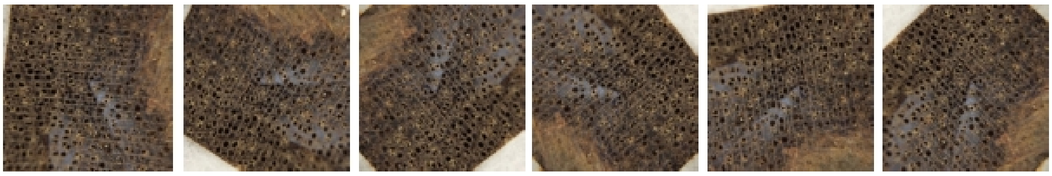


FIGURE 4.8: Examples of rotation augmentation, center-cropped and down-sampled to 256×256. From Left: Input Image, 65, 125, 225, 285 and 320 degrees rotations.

For rotation augmentations, the parameters were in the range of 0 – 355 degrees with the interval of 5 degrees. A randomly chosen degree rotated the image using geometric angle transformation method. Examples are shown in Figure 4.8.

4.2.2.3 Brightness Augmentation

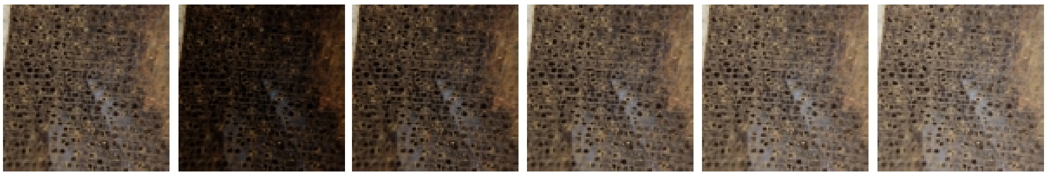


FIGURE 4.9: Examples of brightness augmentation, center-cropped and down-sampled to 256 × 256. From Left: Input Image, 0.5, 0.8, 1.1, 1.3, and 1.5 gammas brightness adjustment.

As for brightness adjustment, this augmentation method was to adjust the gamma value of image and the parameters used were in the range of 0.5 to 1.5 with an interval of 0.1. Hence, the image was randomly darkened or brightened based on the randomized chosen parameter. Examples are shown in Figure 4.9.

4.2.2.4 Rim-blurring Augmentation

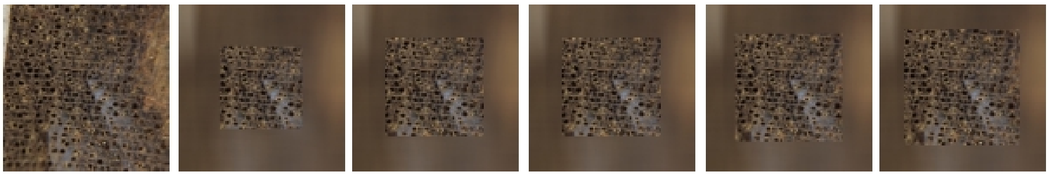


FIGURE 4.10: Examples of rim-blurring augmentation, center-cropped and down-sampled to 256 × 256. From Left: Input Image, 0.50, 0.55, 0.60, 0.65, and 0.70 ratios of rim-blurring.

As for rim-blurring augmentation method, this distortion method aimed to blur out the outer rim of the image which left the center part of the image in focused. The parameter for this method defined the percentage of portion of center in-focused relative to the blurred outer rim. For example, a ratio of 0.5 signifies that 50% of the center portion of the image was in-focused where the remaining 50% of the outer rim was blurred. A range of 0.5 – 0.7 with an interval of 0.05 were randomly chosen to rim-blur the images. The reason behind the design of this method was to apply a constraint to the classifier and tries to reduce the effect of optical barrel distortion of the macro lenses leaving the center portion in focused and undistorted. With this distortion method, it forced the classifier to focus its attention to the center portion of the image. Examples are shown in Figure 4.10.

4.2.2.5 Field of View Resizing Augmentation

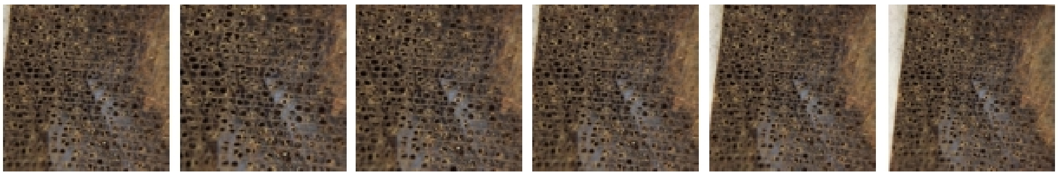


FIGURE 4.11: Examples of resize augmentation, center-cropped and down-sampled to 256×256. From Left: Input Image, 0.50, 0.55, 0.60, 0.65, and 0.70 resize ratios of resizing.

Image captured by image acquisition device was in $4 : 3$ aspect ratio. As per publicly open source image dataset, e.g. ImageNet Dataset, a square image was cropped from the raw image. This gave rise to the last augmentation method where user can define the field of view of the cropped image from the raw image. As a result, the difference of field of view could be chosen to create the effect of multi scaling of images. This method takes parameter within the range of 0.5 – 0.7 with an interval of 0.05 where 0.5 means that 50% of the center portion of the image was cropped from the raw image. Besides that, the center cropping method also mitigated the issue of optical barrel distortion on image introduced by the macro lens. Examples are shown in Figure 4.11.

4.2.2.6 Online Augmentation

Online augmentation referred to on-the-fly augmentation in training phase where images were randomly augmented before passing into the ConvNet for forward propagation. In comparison to offline augmentation, online augmentation did not store images on local storage but it required computational power to augment the data on-the-fly during training where offline augmentation requires one off computation. It was a compromise between storage space and computational power where user had to choose from in order to enjoy the benefits of limited local storage while having a hugely augmented dataset for training of ConvNet. If storage is abundant, it is advisable to use offline augmentation to reduce the computational workload and in turn reduces the training time of ConvNet. The augmentation methods are defined as follows:

Online random augmentation methods:

1. Colour jittering augmentation
2. Random cropping
3. Mirror flipping

Figure 4.12 shows the flow for online augmentation. There are two groups of augmentations where the first group is colour jittering augmentation follows by group of random cropping and mirror flipping.

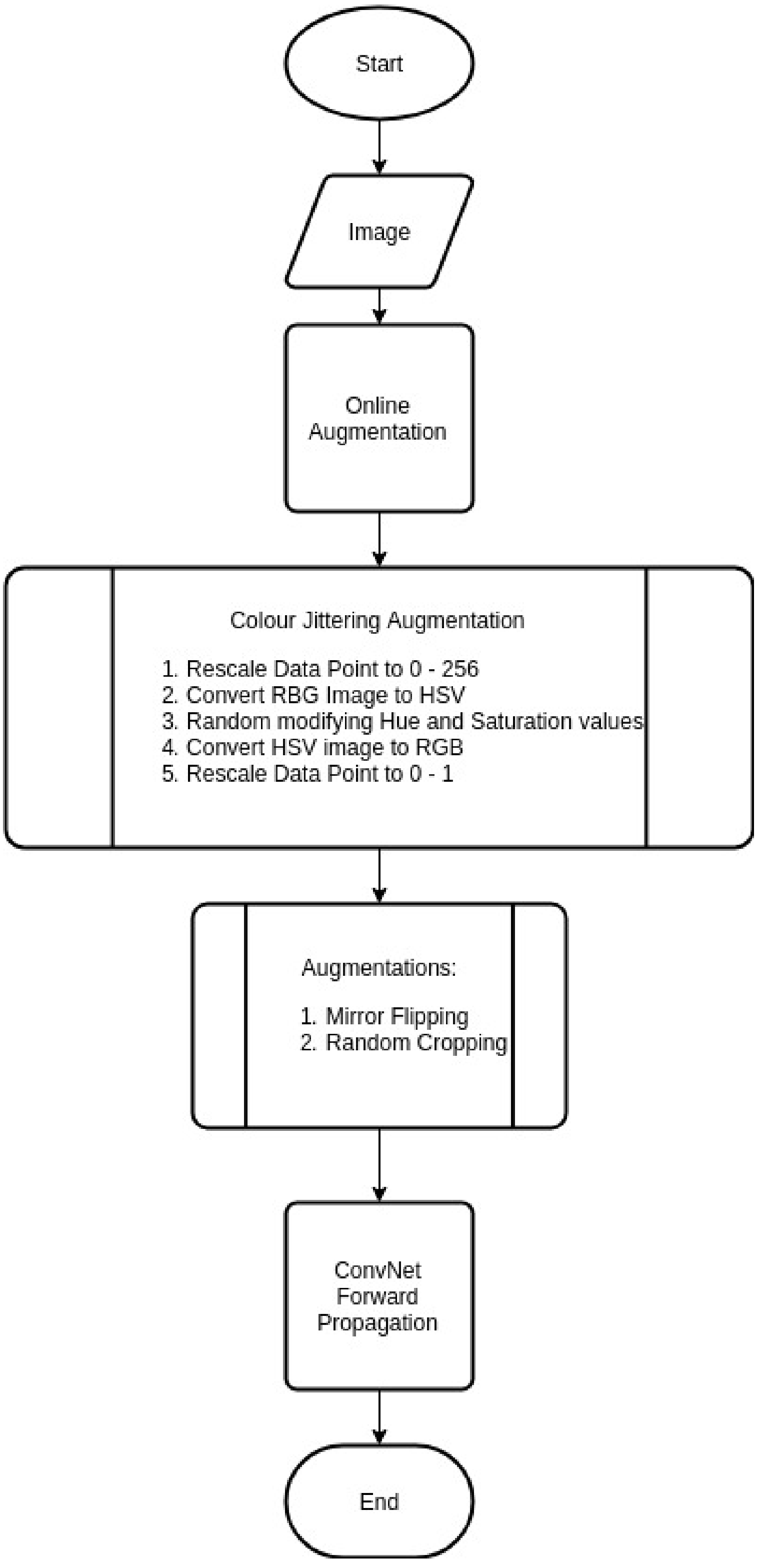
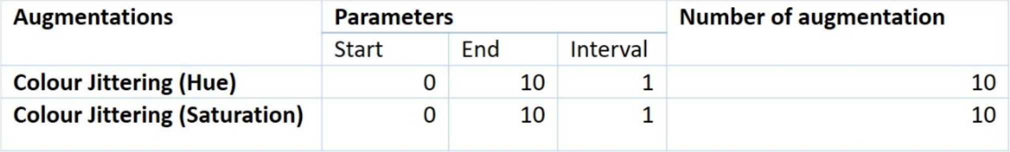

Start
Image
Online Augmentation
Colour Jittering Augmentation
1. Rescale Data Point to 0 - 256
2. Convert RBG Image to HSV
3. Random modifying Hue and Saturation values
4. Convert HSV image to RGB
5. Rescale Data Point to 0 - 1
Augmentations:
1. Mirror Flipping
2. Random Cropping
ConvNet Forward Propagation
End

FIGURE 4.12: Flow chart for online augmentation

4.2.2.7 Colour Jittering Augmentation



Augmentations	Parameters			Number of augmentation
	Start	End	Interval	
Colour Jittering (Hue)	0	10	1	10
Colour Jittering (Saturation)	0	10	1	10

TABLE 4.3: Colour jittering augmentation method and parameters detail

Table 4.3 shows the details of online augmentation method, namely colour jittering distortion method. This method randomly altered the hue and saturation of the image to simulate a warm or cool colour temperature in the image. The author foreseen that the white balance in the smart-phone camera may generate different colour temperature under the influence of varying ambient lighting temperature. Therefore, this method gave variation of colour temperature to the image for the ConvNet to learn. Examples are shown in Figure 4.13.

Image was first rescaled back to 0 – 255 for each data point. Next, image was then converted to HSV format. The hue and saturation values of image were randomly altered. Next, image was converted back to RGB format and lastly rescaled to 0 – 1 floating number for each data point.

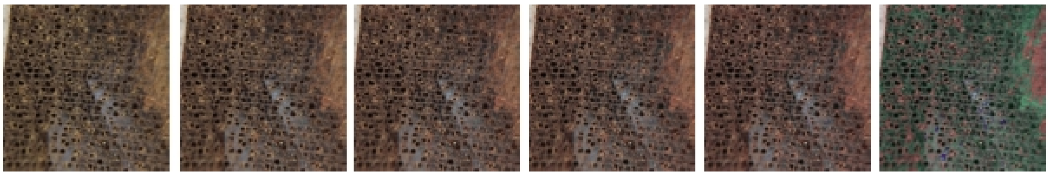


FIGURE 4.13: Examples of online colour jittering distortion augmentation. From Left: Input Image, 3, 5, 6, 8, and 15 hues and saturation values jittering.

4.2.2.8 Random Cropping Augmentation

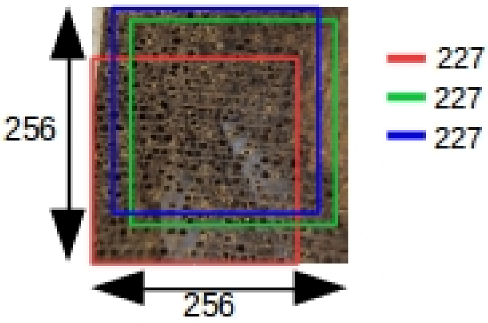


FIGURE 4.14: Random Cropping from input image of 256 × 256 to 227 × 227 output image.

Random cropping was used to augment data by randomly cropping a smaller squared image from input image to create translation to input image. This methods simulated translation in image. Examples are shown in Figure 4.14.

4.2.2.9 Mirror Flipping Augmentation

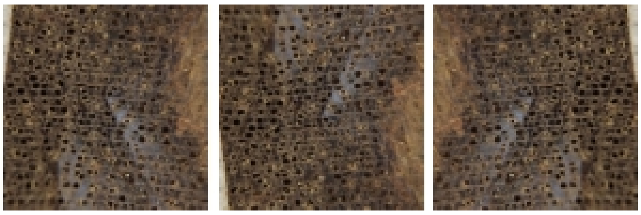


FIGURE 4.15: Examples of mirror flippings. From Left: Input image, vertical flipping and horizontal flipping.

Mirror flip method was used to randomly flip image using either horizontal or vertical axis. Examples are shown in Figure 4.15.

4.2.3 Preprocessing

For training of ConvNet, preprocessing was applied to data prior forward propagation. Preprocessing usually includes normalization of images to help the generalization of ConvNet.

For preprocessing, the author applied mean image subtraction, rescaling data to 0 – 1 floating number. A mean image was calculated from training dataset which served as a normalization to center the values of the image. ConvNet or neural networks often learns better from data between the range of 0.0 – 1.0. Therefore, the image was rescaled to between 0 – 1 for each pixel. As per major ConvNet designed for ImageNet dataset, input image was squarely cropped and down-sampled to 227×227 pixels. The procedures are defined in Figure 4.16.

Preprocessing Procedure:

1. Mean Image Subtraction
2. Rescale data point to $0 - 1$ floating number

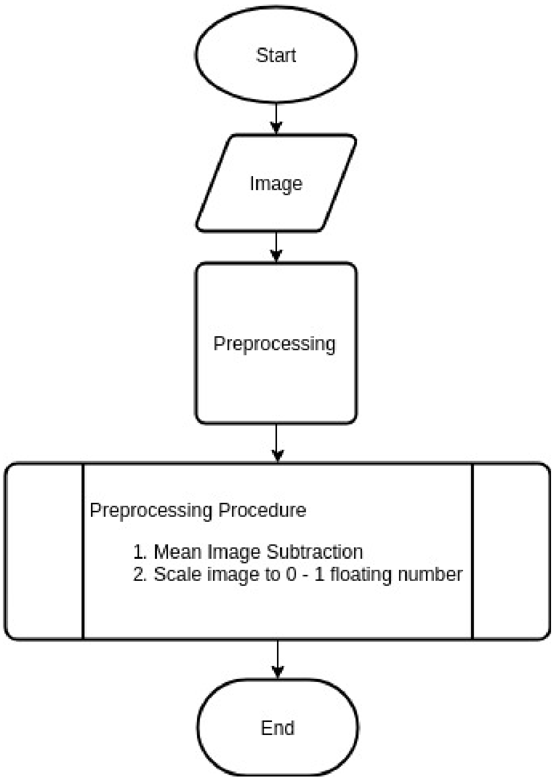

Start
Image
Preprocessing
Preprocessing Procedure
1. Mean Image Subtraction
2. Scale image to 0 - 1 floating number
End

FIGURE 4.16: Flow chart for preprocessing of image.

4.2.4 Dataset Generation

After offline augmentations, training and validation of different sizes were generated using the procedure as follows:

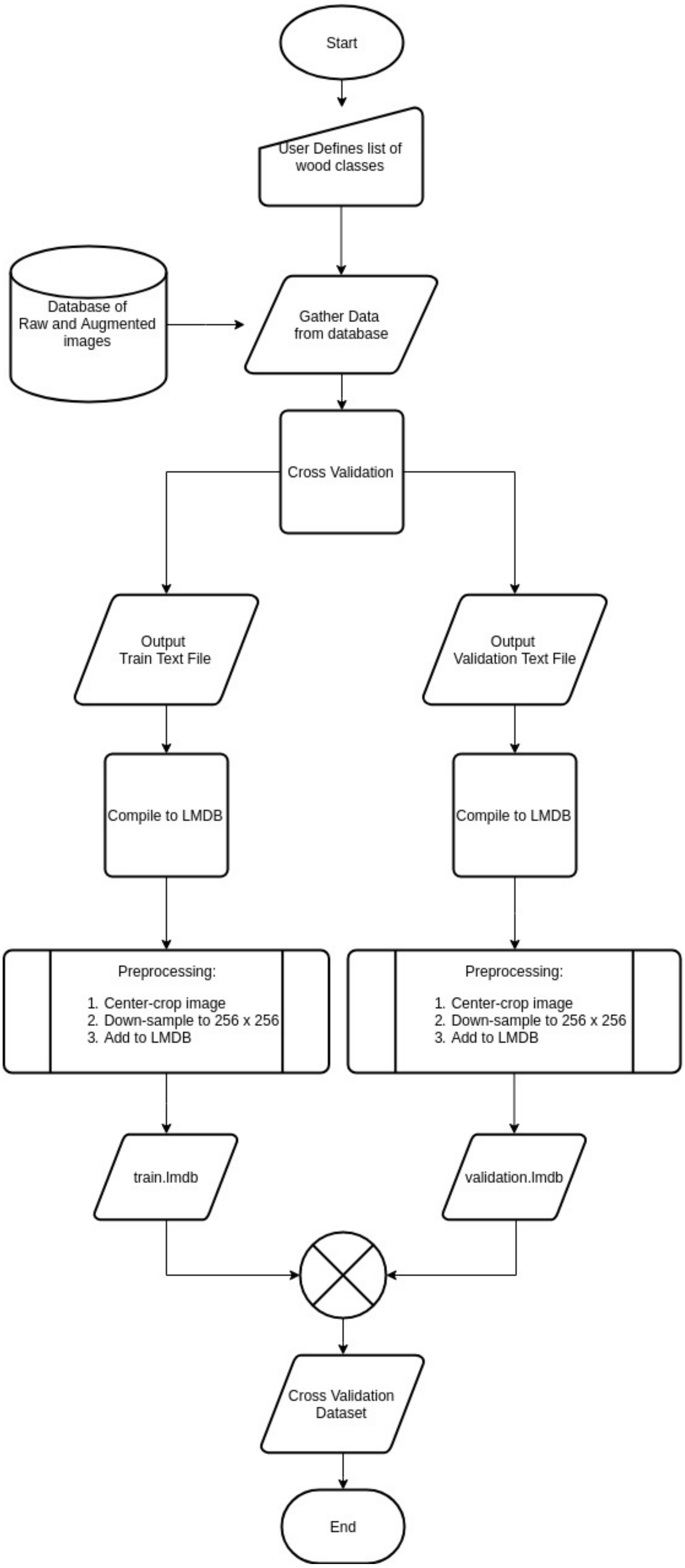


FIGURE 4.17: Flow chart for Dataset Generation.

Dataset generation procedure:

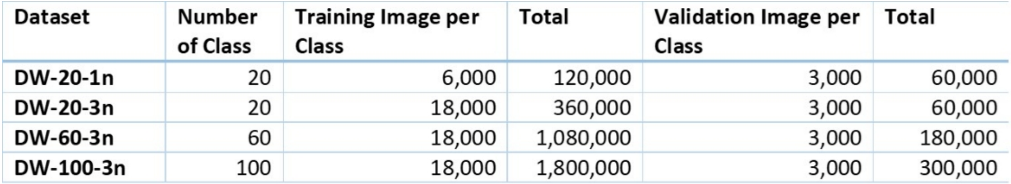
1. User defined the wood types to be included in the dataset
2. Query was made to the database to gather all the raw and augmented images for the given wood types
3. Wood types were separated into train and validation dataset based on specimen IDs for cross-validation purpose
4. Two text files with image file paths and labels were generated as train.txt and val.txt
5. The text files were then compiled into 2 LMDB datasets.
6. Preprocessing was applied to all the images as defined in text files.

   Center-cropped image into square image.

   Down-sampled the image to 256×256 pixels width and height.
7. A dataset was generated with train and validation LMDBs.

Figure 4.17 shows the flow on generating a dataset. User defined the list of wood types to be included in the dataset. The number of wood types determined the dataset sizes. Queries were made to the database that contains information of separated specimens' IDs for cross validation. Next, text files were created with image file paths and labels for training and validation. Given the text files, images were preprocessed and added to LMDB database. The preprocessing methods included center-cropping image into square image and down-sampling the cropped image to 256×256 pixels in width and height. Lastly, two of the training and validation LMDBs formed a dataset for training.

Table 4.4 shows the detail of dataset generated after augmentations. There were a total of 3 dataset of varying number of classes namely, 20-class, 60-class and 100-class with the exception of 20-class dataset where 2 sizes of are available. Each



Dataset	Number of Class	Training Image per Class	Total	Validation Image per Class	Total
DW-20-1n	20	6,000	120,000	3,000	60,000
DW-20-3n	20	18,000	360,000	3,000	60,000
DW-60-3n	60	18,000	1,080,000	3,000	180,000
DW-100-3n	100	18,000	1,800,000	3,000	300,000

TABLE 4.4: Detail of dataset

class was balanced with augmented and non-augmented images to 18,000 images for training. As for validation, each class was balanced up to 3,000 images. For the largest dataset of 100-class, there were 1,800,000 training images and 300,000 validation images. As mentioned in section 4.2.1, the training and validation images were separated based on specimens ID, where the validation specimens were unseen to the ConvNet during training.

4.3 Convolution Neural Network Design

To kickstart the process of ConvNet design, SqueezeNet (SqNet) by Iandola *et al.* (2016) was used for transfer learning and served as benchmark. Next, a new network design by the author, namely BlazeNet (BzNet) was used to improve the performance of classifier. Other than that, the author also introduced a variant of BlazeNet architecture with colour conversion layer (BzNet Colour). Besides, a grayscale classifier was also trained for benchmark purpose.

ConvNet Model Architectures:

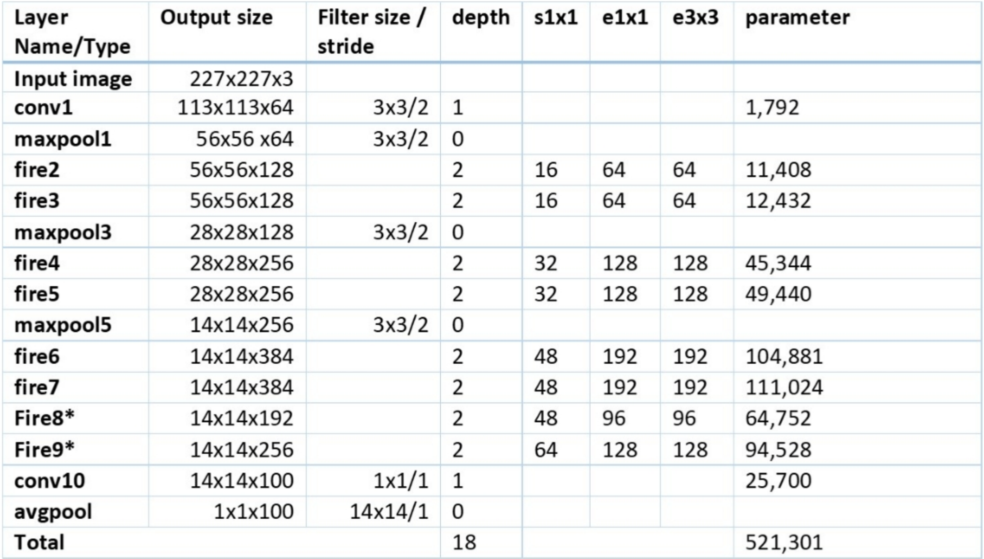
1. SqueezeNet (SqNet, Sq)
2. BlazeNet (BzNet, Bz)
3. BlazeNet Colour Conversion (BzNet Colour, BzC)
4. BlazeNet Grayscale (BzNetGray, BzG)

4.3.1 SqueezeNet

AlexNet by Krizhevsky *et al.* (2012) showed promising result on applying ConvNet on ImageNet-1000 class problem. Since then, researchers have been looking to minimize the size of ConvNet while achieving similar or better result than AlexNet. SqueezeNet (SqNet) was used as benchmark network in this research.

4.3.1.1 Transfer Learning

In practice, it is expensive in terms of computational power to train ConvNet from scratch, i.e. with weight initialization at the beginning of training. For ImageNet 1000-class training, it would take days to complete the full training from scratch. However, pretrained ConvNet weights can be used for transfer learning where a pretrained ConvNet on ImageNet used as starting weight for further training on new dataset. This is common practice to reduce computation power and mitigate the problem of small dataset.



Layer Name/Type	Output size	Filter size / stride	depth	s1x1	e1x1	e3x3	parameter
Input image	227x227x3						
conv1	113x113x64	3x3/2	1				1,792
maxpool1	56x56 x64	3x3/2	0				
fire2	56x56x128		2	16	64	64	11,408
fire3	56x56x128		2	16	64	64	12,432
maxpool3	28x28x128	3x3/2	0				
fire4	28x28x256		2	32	128	128	45,344
fire5	28x28x256		2	32	128	128	49,440
maxpool5	14x14x256	3x3/2	0				
fire6	14x14x384		2	48	192	192	104,881
fire7	14x14x384		2	48	192	192	111,024
Fire8*	14x14x192		2	48	96	96	64,752
Fire9*	14x14x256		2	64	128	128	94,528
conv10	14x14x100	1x1/1	1				25,700
avgpool	1x1x100	14x14/1	0				
Total			18				521,301

TABLE 4.5: Detail of minified SqueezeNet used for transfer learning with output class set to 100 classes. Note: * marked the layer that is different from original SqueezeNet.

As benchmark classifier, SqNet pretrained weights were used as transfer learning to train on DW-20 datasets. However, several modifications were made to the last three Fire Modules, namely Fire7, Fire8 and Fire9 to reduce the number of depth of filter as shown in Table 4.5. The overall network consists of 18 layers with 521,301 parameters.

4.3.2 BlazeNet

Using pretrained SqNet yielded impressive result in wood types identification. However, is there any better ConvNet architecture that can perform better than SqNet for this application? Since the introduction of SqNet, researchers were going deep into building smaller network with different modules to improve performance while reducing network size.

4.3.2.1 Blaze Module

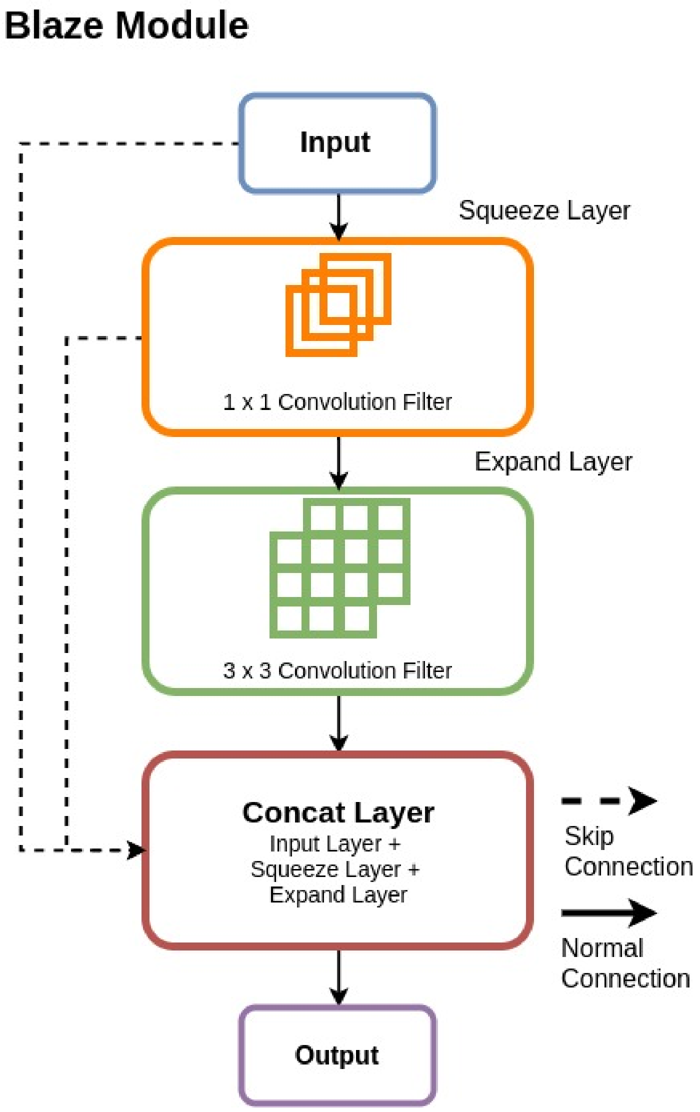


FIGURE 4.18: Blaze Module consists of Squeeze Layer with 1×1 convolution filter followed by Expand Layer with only 3×3 convolution filter.

Using the idea of DenseNet dense connections as discussed in Section 2.4, coupled with Fire Module from SqueezeNet, the author introduced a new module namely Blaze Module as shown in Figure 4.18. Blaze Module, with its root from Fire module consists of Squeeze Layer with 1×1 convolution filter followed be only 3×3 convolution filter in the subsequent Expand Layer. As opposed to Fire module, the major difference is the omission of 1×1 convolution filters in Expand Layer. This decision was made to reduce parameters and make way for skip connections. Besides that, skip connections inspired by DenseNet are added from the Input Layer and Squeeze Layer to the Expand Layer. This is implemented through concatenation layer which combines feature maps from Input Layer, Squeeze Layer and Expand Layer. Each Blaze Module is 2 layers in depth.

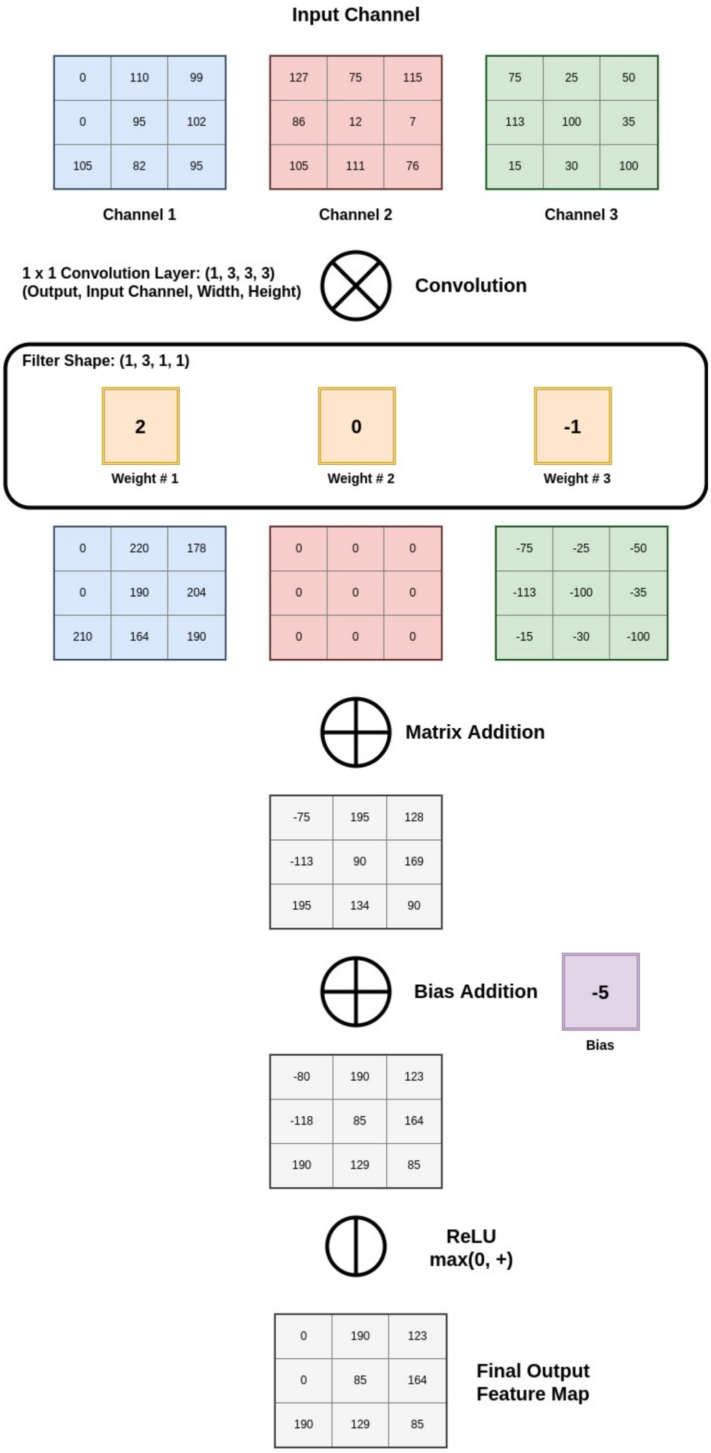
As per Squeeze layer in SqueezeNet, it reduces output feature maps using 1×1 convolution filters, hence the name ”squeeze”. For Expand layer, 3×3 convolution filters with padding set to 1 learn the abstractions. With skip connections, the output of Blaze module concatenates feature maps from Input Layer, Squeeze layer and Expand layer while maintaining the same feature maps dimensions. For example, input with 64 feature maps are squeezed into 16 feature maps in Squeeze Layer. Then, the Expand Layer applies 3×3 convolution onto the 16 feature maps. Lastly, the Blaze Module outputs a total of 96 feature maps where 64 from input layer, 16 from Squeeze Layer and 16 from Expand Layer. Note that all convolutions are followed by ReLU.

The intuition behind Blaze Module is that using 1×1 convolution layer to find the best feature maps from the input layer as discussed in Section 4.3.2.2. In other words, a Blaze module tries to look for the best input feature maps, applies abstraction to these feature maps and pass these feature maps to the next layer. This satisfied the idea of removing redundant feature maps as no recurring features are needed as the feature maps flow through the module via skip connections. Note that Input Layer feature maps are pass through the module to the output to strengthen the idea of free data flowing for better error back propagation.

To summarize, Blaze Module takes in input feature maps, finds the best feature maps, learns new abstraction and outputs these new feature maps along with existing input feature maps.

4.3.2.2 1 x 1 Convolution Filter

This section discusses the operation of convolution layer and how 1×1 convolution layer helps in achieving parameter reduction and finding the best feature maps.



Channel 1

0	110	99
0	95	102
105	82	95

Channel 2

127	75	115
86	12	7
105	111	76

Channel 3

75	25	50
113	100	35
15	30	100

0	220	178
0	190	204
210	164	190

0	0	0
0	0	0
0	0	0

-75	-25	-50
-113	-100	-35
-15	-30	-100

Matrix Addition

-75	195	128
-113	90	169
195	134	90

Bias Addition

-80	190	123
-118	85	164
190	129	85

Final Output Feature Map

0	190	123
0	85	164
190	129	85

FIGURE 4.19: Illustration of how a 1×1 convolution filter works on input channels of 3 with 1 output feature map.

Convolution Layer Operation Procedure:

1. Apply convolution on N-inputs with N-filters
2. New feature maps are produced with each input
3. Summation of feature maps with bias
4. A new feature map is produced at output

Figure 4.19 illustrates the procedure of convolution in a 1×1 Convolution Layer. This illustration consists of only one feature map as output. Hence, this convolution layer is in the shape of $(1, 3, 3, 3)$ based on the dimension of

(Number of Output Feature Map, Number of Input Channel, Width of Feature Map, Height of Feature Map)

To produce one feature map, the filter weight shape is set to $(1, 3, 3, 3)$ according to the dimension of

(Number of Output Feature Map, Number of Input Channel, Width of Filter Kernel, Height of Filter Kernel)

Based on the dimension, each input channel is assigned with one 1×1 filter weight. Moreover, the number of bias is dependent on the number of output feature map which in this case is 1. Convolution of 1×1 filter weight is simply a matrix multiplication of the input channel feature map with the weight as shown in Figure 4.19. For example, Channel 1 is multiplied with Weight 1 of value 2. After matrices multiplication of each channel and respective weight, the output feature maps are then sum together using matrix addition. The summation of feature maps is then added with bias of the filter. Lastly, a rectifier function ReLU is applied to the feature map to output the final output map.

The filter weights are used to demonstrate how 1×1 convolution filter select the best feature from input channel. The weights and bias are trainable in training phase. As shown in Figure 4.19, Weight 1 is set to 2, Weight 2 is 0 while Weight 3 is -1. The output feature map of Weight 2 is zeroed which means that the filter thinks that Channel 2 is not important. On contrary, Weight 1 and Weight 3 find that Channel 1 and Channel 3 consist of good features. With ReLU at the end of the process, negative values are zeroed and left only with positive values. Hence, positive weight can be regarded as asserting a good feature while negative weight tries to reduce the importance of feature.

In summary, 1×1 convolution learn to find the best input feature map using trainable 1×1 convolution filter kernel to apply weightage to each input feature map. The sign of the weights is an important tool to select the best features.

4.3.2.3 Blaze Block

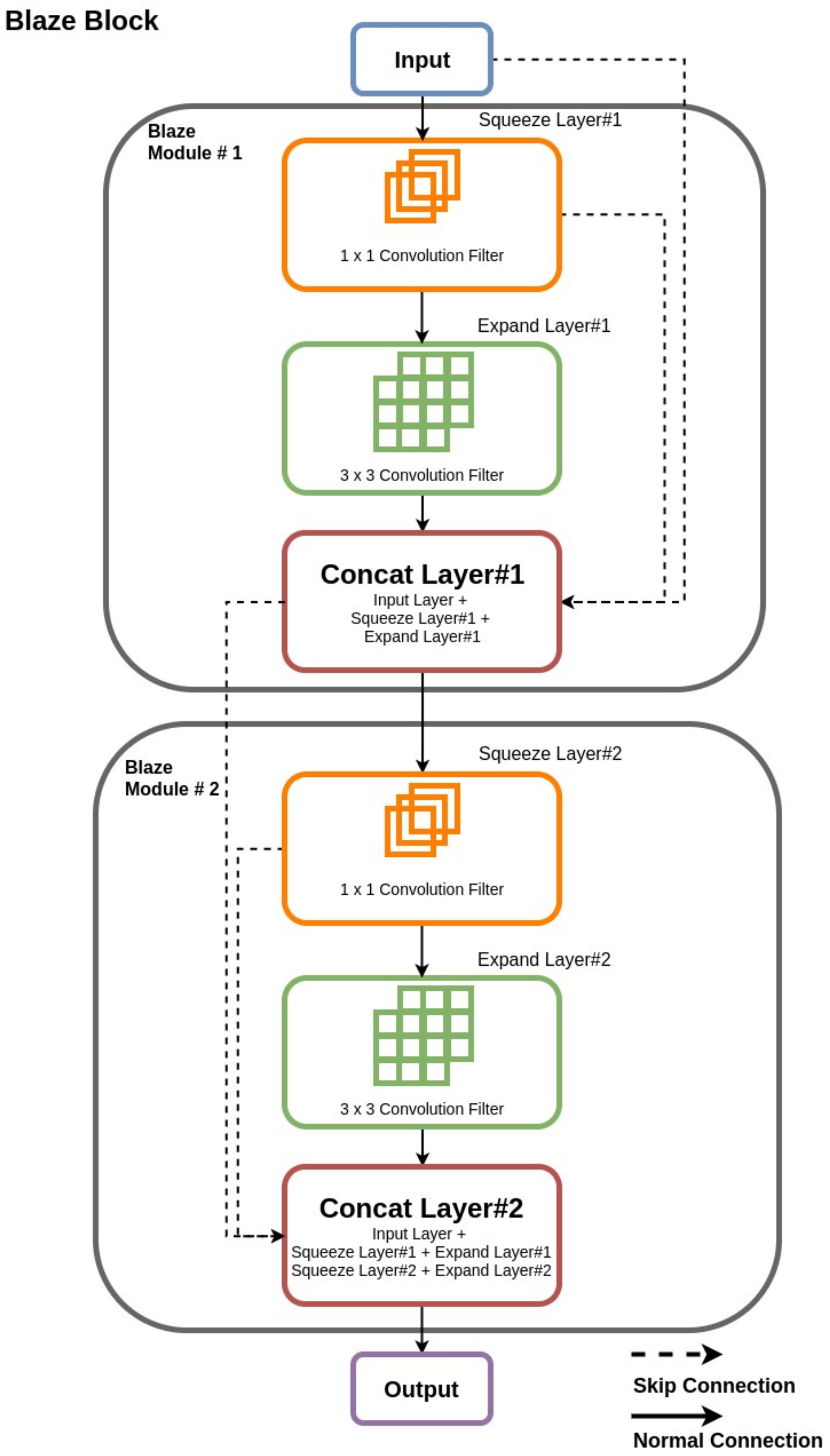


FIGURE 4.20: Architecture of a Blaze Block with 2 Blaze modules connected in subsequent fashion.

Figure 4.20 shows the basic connection of Blaze Block using 2 Blaze Modules. The input for Blaze Module 2 is Concatenate(Concat) Layer 1 which consists of feature maps from Squeeze Layer 1 and Expand Layer 2. The flow of feature maps continues through Squeeze Layer 2 and Expand Layer 2. At the end of output, Concat Layer 2 consists of feature maps from Squeeze Layer 1, Expand Layer 2, Squeeze Layer 1 and Expand Layer 2.

4.3.2.4 Architecture

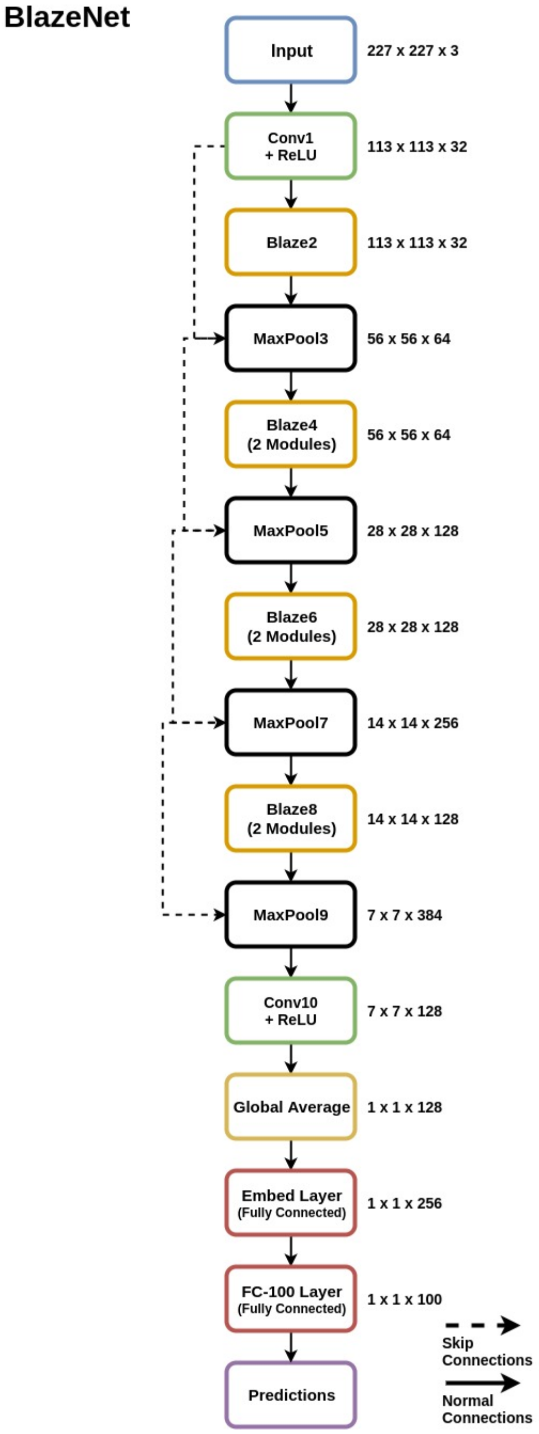
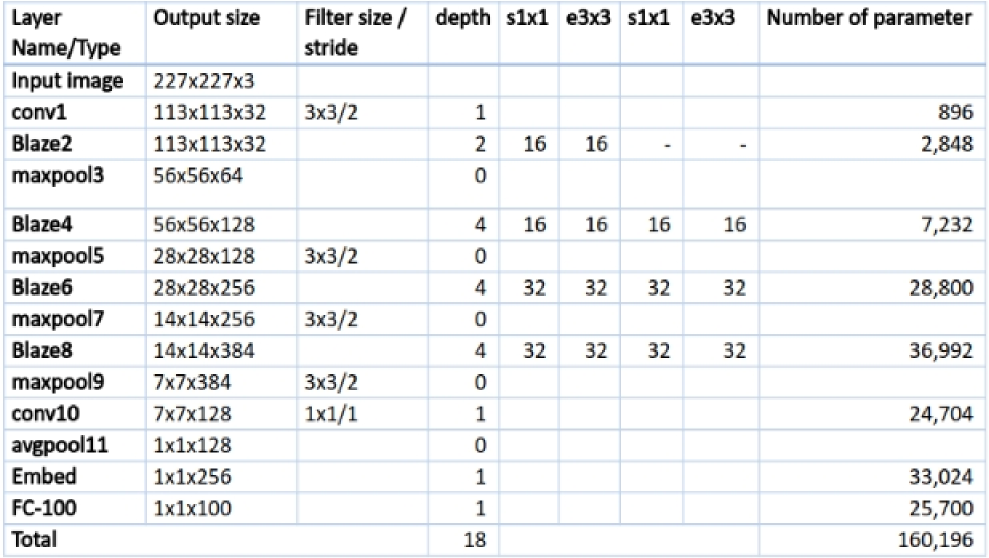


FIGURE 4.21: Overall Architecture of BlazeNet.

Figure 4.21 shows the overall architecture BlazeNet convolution neural network with detail of parameters shows in Table 4.6. As per SqueezeNet, BlazeNet took in RGB image in $227 \times 227 \times 3$ dimension. The first convolution layer, Conv1 was set to output 32 feature maps with ReLU. Next, Blaze2 Layer which contained only one Blaze Module set to output 32 feature maps with 16 from Squeeze Layer and 16 from Expand Layer. A skip connection was used to connect Conv1 with Blaze2. The concatenated 64 feature maps from Conv1 and Blaze2 were then max-pooled to reduce the dimension of feature maps by half.



Layer Name/Type	Output size	Filter size / stride	depth	s1x1	e3x3	s1x1	e3x3	Number of parameter
Input image	227x227x3							
conv1	113x113x32	3x3/2	1					896
Blaze2	113x113x32		2	16	16	-	-	2,848
maxpool3	56x56x64		0					
Blaze4	56x56x128		4	16	16	16	16	7,232
maxpool5	28x28x128	3x3/2	0					
Blaze6	28x28x256		4	32	32	32	32	28,800
maxpool7	14x14x256	3x3/2	0					
Blaze8	14x14x384		4	32	32	32	32	36,992
maxpool9	7x7x384	3x3/2	0					
conv10	7x7x128	1x1/1	1					24,704
avgpool11	1x1x128		0					
Embed	1x1x256		1					33,024
FC-100	1x1x100		1					25,700
Total			18					160,196

TABLE 4.6: Parameter detail for BlazeNet Architecture. Note, both Embed and FC-100 layers are Fully connected layers.

After that, the network was constructed with a series of Blaze Block and Max-pooling layer. Note that the Blaze Block hereafter uses 2 Blaze Modules with depth of 4 as shown in Figure 2. Max-pooling reduced the dimension of feature maps by half progressively as the network gets deeper. The idea was to let the Blaze Blocks to learn the features at different dimensions. At the output of Blaze Block, the input features were passed down and retained along with new features learnt in the block. These outputs are down-sampled and passed to another Blaze Block to learn on smaller dimension.

At Max-pool 9 Layer, which outputed 384 feature maps, consisted of original 32 features from Conv1 along with new features learned by Blaze Blocks. In other words, 352 additional new feature maps are added through Blaze Blocks.

After Maxpool9, the network resolves back to a more conventional ConvNet where a convolution layer is added. Conv10 reduced the input feature maps of 384 of Maxpool9 to 128 feature maps. Again, 1×1 convolution filters were used in Conv10 to learn 128 of the best feature maps from Maxpool9. These feature maps were then down-sampled using global average pooling to reduce the dimension to 1×1. Next, a fully connected layer was connected with output set to 256. This layer was designed as layer to embed the information of input image. Lastly, the last fully connected layer serves as the classification layer where the number of output was set to follow the number of classes which in this case was 100.

As a short summary, Blazenet can be regarded as a feature extractor that embed the input image for classification. In the feature extraction, Blaze Block was used to pass down information from input layers after each block while at the same time learning new features along the way that can best represent the input image. This network was designed to be compact with reduced parameters as it just 23% of the size of equivalent SqNet for this application.

4.3.3 BlazeNet Colour Conversion

A colour conversion module was introduced to improve the robustness against colour temperature while retaining the overall accuracy. Upon closer investigation on colour temperature, under different conditions, colour temperature affects the colour of the wood types in the image. Warmer or cooler colour tones may render the colour information associated with wood types useless. In this situation, one might propose that using grayscale to reduce the effect of colour information affecting the prediction of classifier.

To study and improve the performance of BzNet, grayscale dataset was generated and trained to compare with the Colour Conversion Module as introduced by the author.

4.3.3.1 Colour Conversion Module

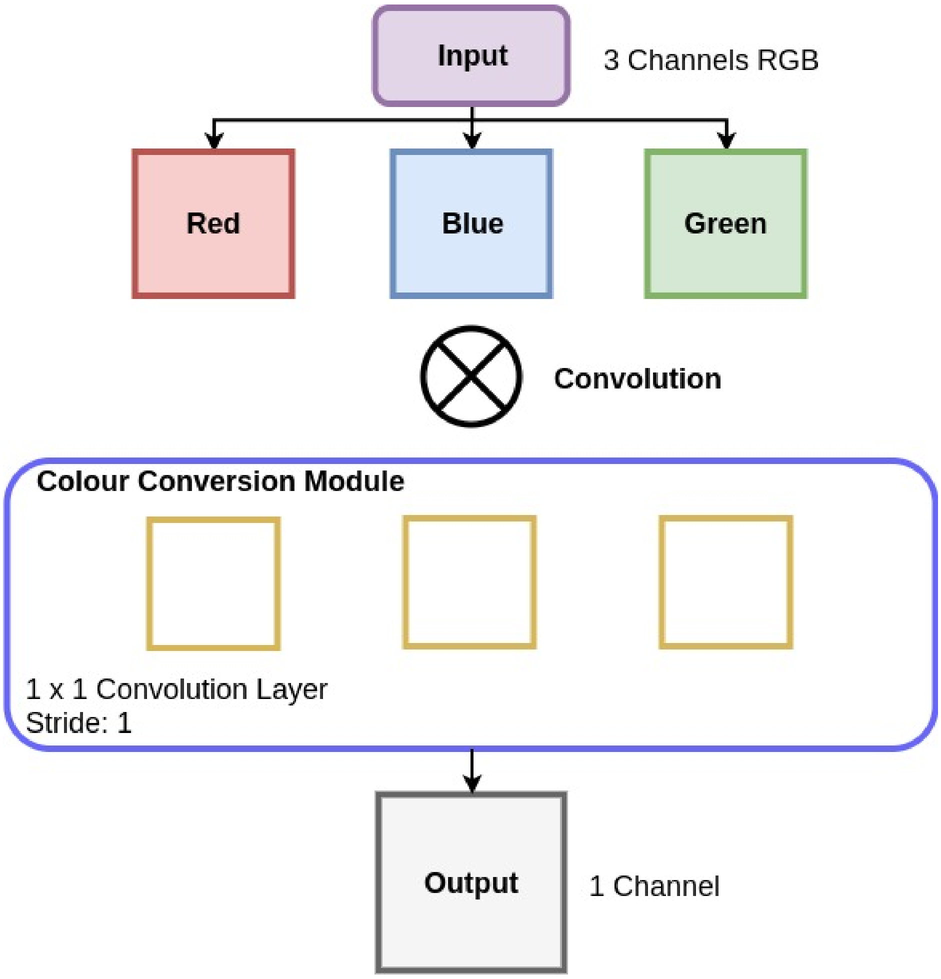
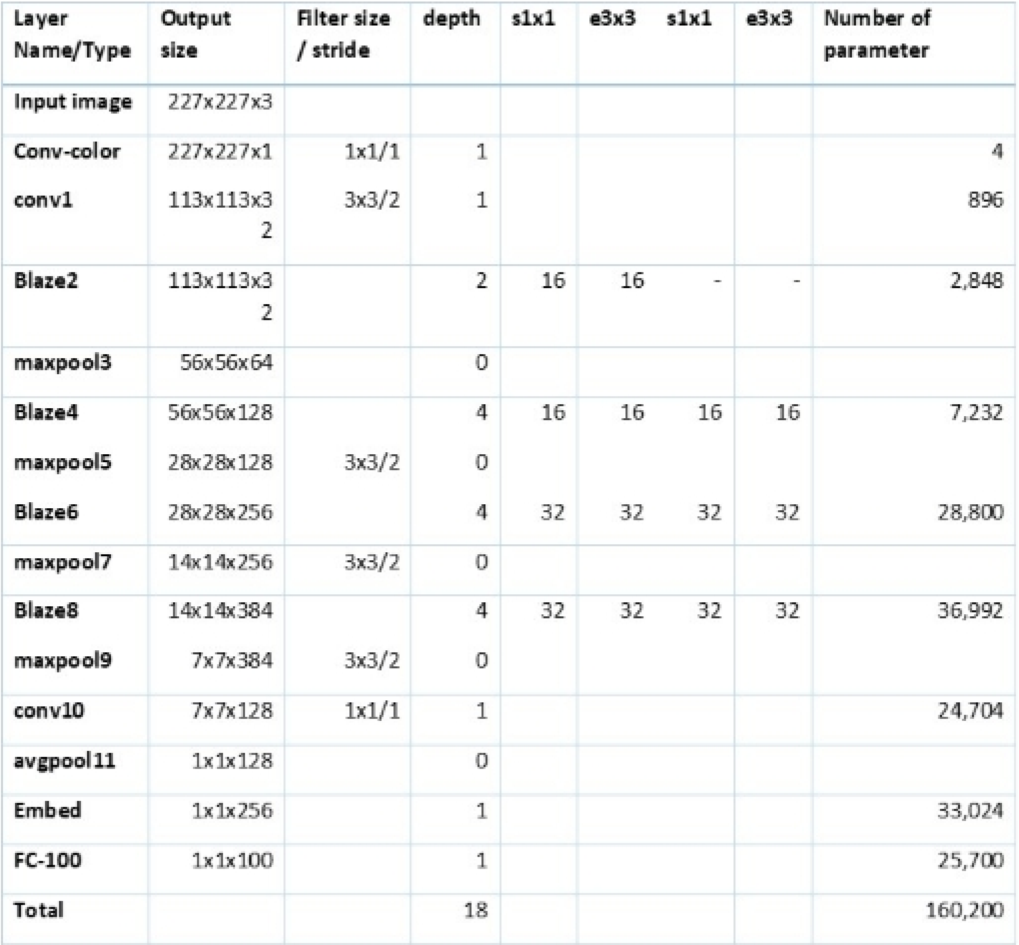


FIGURE 4.22: Colour Conversion Module for BlazeNet Colour Conversion architecture.

The intuition behind Colour Conversion Module was to let ConvNet to learn the best colour information that can represent a wood type. Grayscale is considered a straight forward conversion from a 3 channel RGB image to just single channel intensity.

Using a 1×1 convolution filter with single output, the Colour Conversion module learned to extract the best input feature maps which in this case the inputs are red, green and blue channels of input image. Setting stride parameter to 1 maintains the input channels width and height. As described in Section 4.3.2.2, Colour Conversion module was tasked to find the best colour channel or a combination of channels that can represent wood types. In contrast to grayscale conversion, Colour Conversion Module was a method that was learnable from data in converting 3 channels image to single channel image.

4.3.3.2 Architecture



Layer Name/Type	Output size	Filter size / stride	depth	s1x1	e3x3	s1x1	e3x3	Number of parameter
Input image	227x227x3							
Conv-color	227x227x1	1x1/1	1					4
conv1	113x113x32	3x3/2	1					896
Blaze2	113x113x32		2	16	16	-	-	2,848
maxpool3	56x56x64		0					
Blaze4	56x56x128		4	16	16	16	16	7,232
maxpool5	28x28x128	3x3/2	0					
Blaze6	28x28x256		4	32	32	32	32	28,800
maxpool7	14x14x256	3x3/2	0					
Blaze8	14x14x384		4	32	32	32	32	36,992
maxpool9	7x7x384	3x3/2	0					
conv10	7x7x128	1x1/1	1					24,704
avgpool11	1x1x128		0					
Embed	1x1x256		1					33,024
FC-100	1x1x100		1					25,700
Total			18					160,200

TABLE 4.7: Details of BlazeNet Colour Conversion network layer.

The Colour Conversion Module used a 1×1 convolution filter with 3 inputs and single output. It was connected in between the Input Data Layer and BzNet first layer. As planned, it converted RGB images to single channel image and pass into BzNet. The detail of layer parameters for BlazeNet Colour Conversion architecture is tabulated in Table 4.7.

4.3.4 BlazeNet Grayscale

BlazeNet Grayscale was served as a benchmark comparison to BzNet Colour. The architecture was similar to BzNet with the exception of input image which was a single channel grayscale image. Detailed architecture are described in section 4.3.4.1.

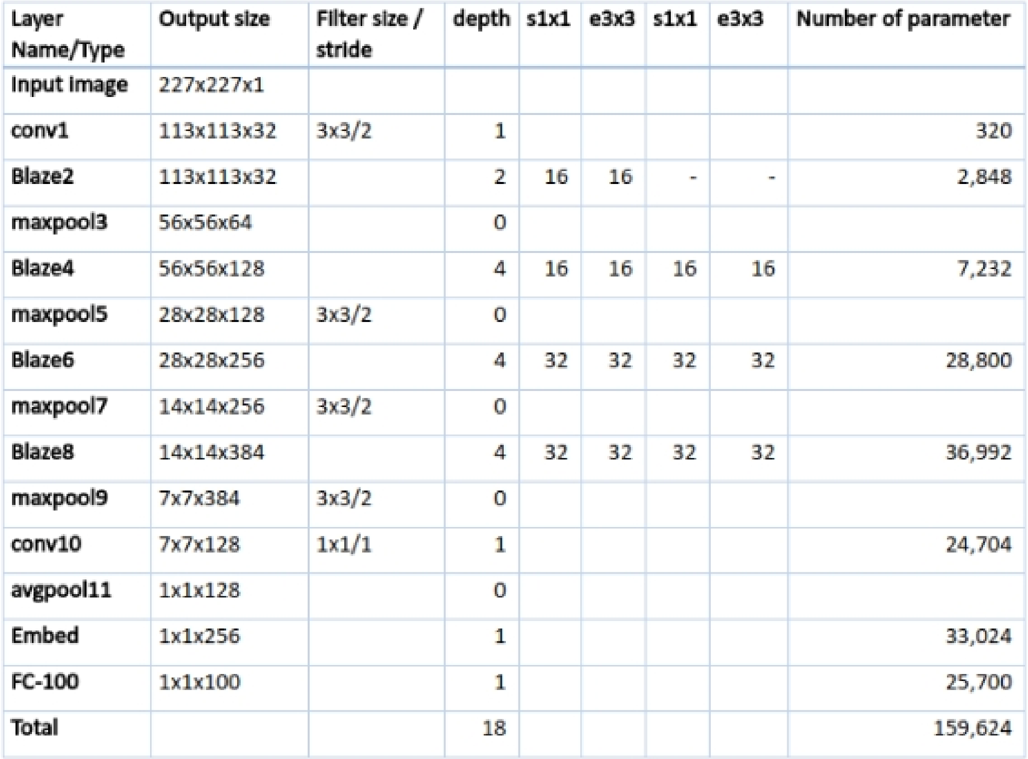
4.3.4.1 Architecture

Table 4.8 shows the details of BzNet Gray architecture in layers. The most significant difference is that the input image is a 1-channel grayscale image as opposed to 3-channel RGB image. Hence, the parameters of first conv1 layer was reduced due the reduction of input image channel.

4.3.5 Comparison of Architecture

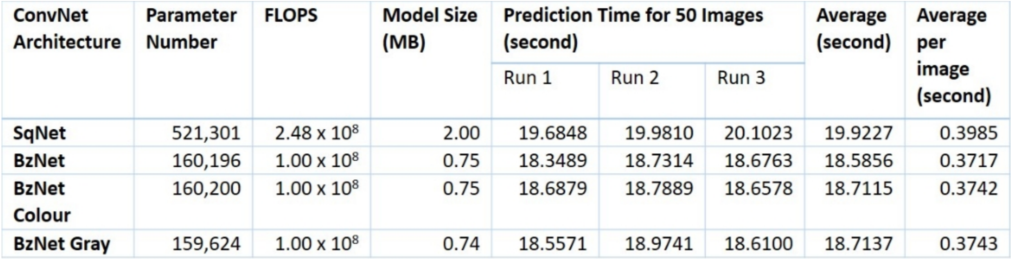
This section summarizes the ConvNet architectures mentioned in the previous sections and tabulated in Table 4.9.

The comparison criteria are Number of parameters for the architecture, Floating points operation per second (FLOPS), Model size in Megabytes, Prediction time for 50 images, Average of prediction time for 50 images and Average of prediction time for single image. As seen in Table 4.9, one observation is that the



Layer Name/Type	Output size	Filter size / stride	depth	s1x1	e3x3	s1x1	e3x3	Number of parameter
Input Image	227x227x1							
conv1	113x113x32	3x3/2	1					320
Blaze2	113x113x32		2	16	16	-	-	2,848
maxpool3	56x56x64		0					
Blaze4	56x56x128		4	16	16	16	16	7,232
maxpool5	28x28x128	3x3/2	0					
Blaze6	28x28x256		4	32	32	32	32	28,800
maxpool7	14x14x256	3x3/2	0					
Blaze8	14x14x384		4	32	32	32	32	36,992
maxpool9	7x7x384	3x3/2	0					
conv10	7x7x128	1x1/1	1					24,704
avgpool11	1x1x128		0					
Embed	1x1x256		1					33,024
FC-100	1x1x100		1					25,700
Total			18					159,624

TABLE 4.8: Details of BlazeNet Grayscale network layer.



ConvNet Architecture	Parameter Number	FLOPS	Model Size (MB)	Prediction Time for 50 Images (second)			Average (second)	Average per image (second)
				Run 1	Run 2	Run 3		
SqNet	521,301	2.48×10^8	2.00	19.6848	19.9810	20.1023	19.9227	0.3985
BzNet	160,196	1.00×10^8	0.75	18.3489	18.7314	18.6763	18.5856	0.3717
BzNet Colour	160,200	1.00×10^8	0.75	18.6879	18.7889	18.6578	18.7115	0.3742
BzNet Gray	159,624	1.00×10^8	0.74	18.5571	18.9741	18.6100	18.7137	0.3743

TABLE 4.9: Comparison of ConvNet Architectures. Note: The predictions were run on CPU only.

number of parameters is directly correlated to FLOPS and model size. The higher in number of parameters, the higher FLOPS and model size and vice versa.

As for number of parameter, SqNet records the highest at 521,301 followed by BzNet Colour at 160,200, BzNet at 160,196 and BzNet Gray at 159,624. The difference between BzNets are within the margin of 0.4%. Generally, the size of BzNets are 30% of SqNet or approximately 3 times smaller than SqNet as demonstrated in the model size. The reduction of parameters gives BzNets the gain in FLOPS which is approximately 40% of SqNet. In other words, SqNet FLOPS is

2.5 times of BzNets. Lower FLOPS is beneficial on computation as it reduces the load of computing.

On the Average Prediction Time, the gain of FLOPS reduction is not reflected in the prediction time of image. For single image prediction, the reduction of time is merely 6.72%. This is because of the preprocessing time of image is the same across all architecture.

In summary, BzNets are approximately 3 times smaller than benchmarked SqNet in number of parameter, FLOPS and model size.

4.4 Evaluation of Classifier Performance

This section discusses the evaluation criteria for classifier performance. The criteria are listed as follows:

Evaluation Criteria for Classifier:

1. Top-1 Accuracy
2. Top-2 Accuracy
3. Confusion Matrix
4. Recall
5. Precision
6. F1 score

The detail of each evaluation criterion is discussed in Section 4.4.1. These criteria are applied in the tests defined as follows:

Types of Test for Classifier:

1. Validation Test
2. Robustness Test

There are two types of tests defined by the author and are discussed in the following Section 4.4.2 and Section 4.4.3.

4.4.1 Evaluation Criteria

4.4.1.1 Top-N Accuracy

Assume P is the prediction of classifier that outputs an array of class labels based on number of class, k as follows:

$$P(s) = [s_1, s_2, ..., s_{k-1}] \tag{4.1}$$

where k = number of class

Top-1 is defined as follows:

$$Top1 \,\epsilon\, argmaxP(s) \tag{4.2}$$

where it is the highest probability of labels in single prediction of single image.

As for Top-2, it is defined as follows:

$$Top2 \,\epsilon\, argmaxP(s)_{0-1} \tag{4.3}$$

The correct label or True Positive is within the second highest of prediction for single image.

Using the following formula to calculate overall accuracy of classifier on all test images,

$$Accuracy = \frac{TP}{\sum(TP + FP + TN + FN)} \times 100\% \tag{4.4}$$

where TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative

At such, the Top-1 and Top-2 accuracy are defined as follows:

$$Top1Accuracy = \frac{\sum_j argmaxP(s)}{\sum_j(TP + FP + TN + FN)} \times 100\% \tag{4.5}$$

$$Top2Accuracy = \frac{\sum_j argmaxP(s)_{0-n}}{\sum_j(TP + FP + TN + FN)} \times 100\% \tag{4.6}$$

where n = 2, j = number of test images

4.4.1.2 Recall

Recall is formulated as follows:

$$Recall = \frac{TP}{\sum(TP + FN)} \times 100\% \tag{4.7}$$

4.4.1.3 Precision

Precision is formulated as follows:

$$Precision = \frac{TP}{\sum(TP+FP)} \times 100\% \tag{4.8}$$

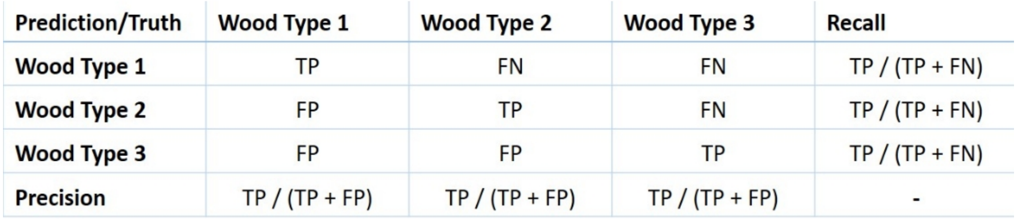
4.4.1.4 F1 Score

F1 score is formulated as follows:

$$F1score = 2\ \times \frac{Recall\ \times\ Precision}{Recall + Precision} \tag{4.9}$$

4.4.1.5 Confusion Matrix

Confusion matrix is a common tool to evaluate classifier on the given truth and predictions. It is often shown in a table form or diagram. The detail of table of confusion matrix is shown in Table 4.10.



Prediction/Truth	Wood Type 1	Wood Type 2	Wood Type 3	Recall
Wood Type 1	TP	FN	FN	TP / (TP + FN)
Wood Type 2	FP	TP	FN	TP / (TP + FN)
Wood Type 3	FP	FP	TP	TP / (TP + FN)
Precision	TP / (TP + FP)	TP / (TP + FP)	TP / (TP + FP)	-

TABLE 4.10: Sample of confusion matrix. NOTE: TP = True Positive, TN = True Negative, FP = False Positive.

Table 4.10 shows the sample confusion matrix table. Header row shows the prediction of each wood types where the first row shows the truth of each wood types. The last column and the last row show recall and precision respectively. The diagonal of the table shows the number of true positive, I.e. the predictions matched truths. The region to the right TP shows FN where predictions do not match the truths. The bottom region of TP shows FP where the predictions from the classifier is wrong.

At the end of each row, recall is calculated based on Equation 4.7 as the ratio of true positive and the sum of TP and FN. In other words, recall measures the per-class accuracy or how well the classifier in identifying each class.

At the end of each column, precision is calculated based on Equation 4.8 as the ratio of TP and the sum of TP and FP. Precision is the measure of how reliable is the prediction of classifier. For instance, when the classifier predicted class A, how likely it is in actual is class A? High precision means that classifier is very confident in its prediction. On the contrary, low precision means that the classifier is not very confident in the prediction where it is most likely going to be wrong.

4.4.2 Validation Test

As per Section 4.2, the validation test runs on the validation dataset generated. The dataset consists of unseen images from the training which are separated based on specimen ID.

The evaluation criteria for validation test is as follows:

Validation Test Evaluation Criteria:

1. Top 1 Accuracy
2. Top 2 Accuracy
3. Recall
4. Precision
5. F1 Score
6. Confusion Matrix

4.4.3 Robustness Test

4.4.3.1 Introduction

Robustness test is defined as how robust is the ConvNet on recognizing wood types under different conditions that may perturb the input image in the field. The author made a few assumptions to the perturbations that may occur in field when the classifier is deployed.

These perturbations are as defined in the list below includes variation in colour temperature, brightness, occlusion, rotations and field of view. Therefore, the author defined an evaluation on the performance of trained ConvNet on robustness against the aforementioned challenging conditions.

Robustness Test Conditions:

1. Colour Temperature
2. Brightness Level
3. Limited Focus
4. Orientation
5. Field of view

Since the image acquisition device consists of a smart-phone camera, the author assumed that colour temperature is critical in changing the colour of input image under different lighting. For example, under a tungsten light, the white balance of the camera may add coldness or warmness to input images. As a result, the colour information of the wood types might not be represented accurately as it is highly controlled by the camera automatic white balance control. Hence, colour temperature test is added to robustness test.

As for brightness, the author foreseen that the classifier has to work in different lighting situations such as a well-lit laboratory, a factory or in the forest with darkened shades. Digital images are represented with numerical integer. Brightness as well as colour information change the value directly to reproduce the brightness effect to the image. With this consideration, the author added brightness test to robustness test.

Due to nature of the retrofitted macro lens, it exhibits optical barrel distortion where center of the image is warped towards the camera. Together with limited depth of field, the outer rim of the image could be easily out-of-focus, resulting in a blurring image. Moreover, wood specimen is prepared by a simple cut on the cross-section where a flat cut is preferred as it can get the best focused image. However, it would take experience and skill to achieve such flat cut. Judging from the end user perspective, where not only confined to wood anatomist, it is difficult to get perfect image. Hence, the classifier has to be able to identify wood types with given limited focused image.

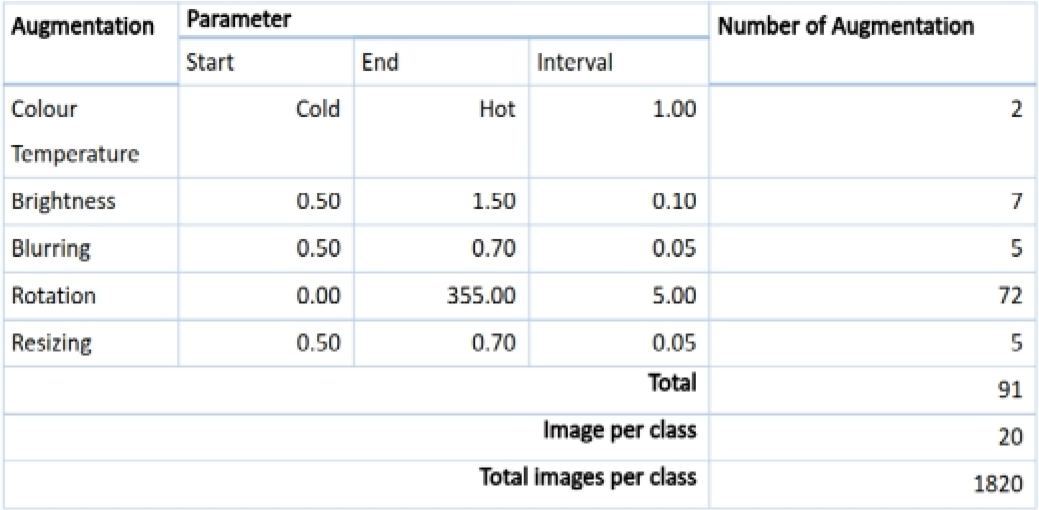
As for orientation, there are not specified reference orientation where the image has to align to certain orientation and it might cause confusion to classifier. Hence, it is added to the robustness test to find out how susceptible is the classifier to different orientation of the input macroscopic wood images.

Lastly, the magnification level of the macro lens directly determines the field of view in the input images. A smaller field of view is achieved with higher magnification level and vice versa. It is intriguing for the author to study the effect of different field of view on the classifier performance. The changing of field of view is added to robustness test.

In order to test on the robustness of ConvNet, a dataset is generated to serve as standardized test dataset for all trained ConvNet. The detail of generating robustness dataset is depicted in the following sections as well as the evaluations criteria.

4.4.3.2 Robustness Dataset

To test on trained ConvNet, the test data should be unseen from the training. As per Section 4.2.4, wood specimens were separated for training and validation. Hence, validation specimens were used to generate robustness dataset to ensure that the data is unseen. The augmentation methods as defined in Section 4.2.2.1 were used to generate the robustness dataset with every parameter in the range. In addition, a new colour temperature augmentation method is added to generate robustness dataset. 20 images from each wood type were randomly selected from the validation specimens and augmented. The details are tabulated in Table 4.11.



Augmentation	Parameter			Number of Augmentation
	Start	End	Interval	
Colour Temperature	Cold	Hot	1.00	2
Brightness	0.50	1.50	0.10	7
Blurring	0.50	0.70	0.05	5
Rotation	0.00	355.00	5.00	72
Resizing	0.50	0.70	0.05	5
			Total	91
			Image per class	20
			Total images per class	1820

TABLE 4.11: Detail of robustness dataset augmentations

A total of 91 augmentations were applied to each image which results in 1820 images for each wood types. The final dataset size was dependent on the number of classes.

4.4.3.2.1 Colour Temperature On testing the effect of colour temperature, images were augmented by changing the colour temperature. Warm and cool images were generated for each input images as shown in Figure 4.23. The augmented images had different colour of the wood specimen.

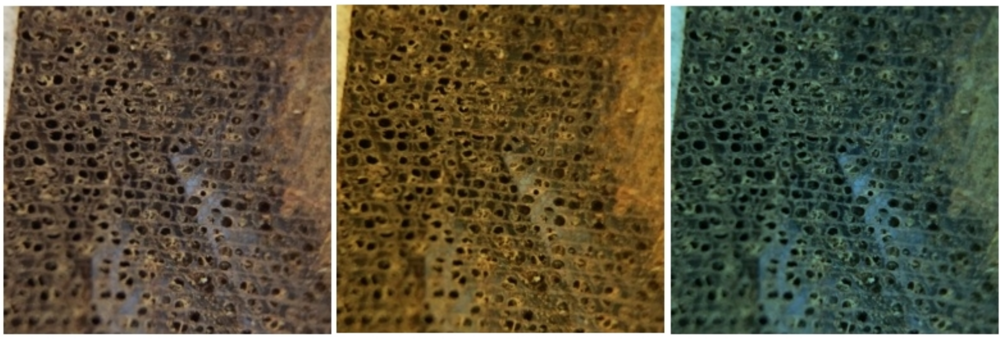


FIGURE 4.23: Example of colour temperature augmentation. From Left: Input image, warm colour image, cool colour image.

4.4.3.2.2 Brightness Level Brightness adjustment was used to augment image for different brightness. This was achieved by using look-up table with precomputed values from input gamma parameter to change the numerical values of image to either brighten or darken the image.

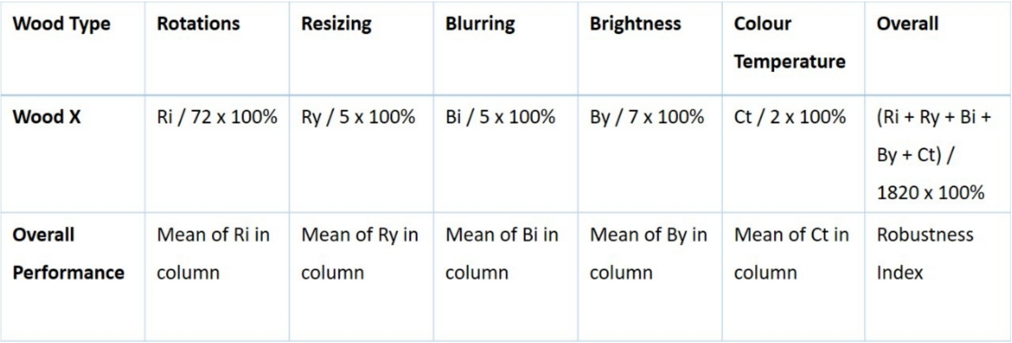
4.4.3.2.3 Limited Focus To test on limited focusing, rim-bluring method as defined in offline augmentation was used to blur the outer rim of the image to reduce the portion of in-focused object in the image. The parameter controls the ratio of in-focused object to out-of-focused outer rim part. For example, a ratio of 0.5 means that the center 50% of the image is in-focused where the remaining 50% of the outer rim is blurred.

4.4.3.2.4 Orientation Geometric transformation was used to rotate the input image based on given parameter on angles. Input images were rotated to create orientations spanning 0 – 355 degrees.

4.4.3.2.5 Field of View The idea of changing the field of view was to simulate different magnification level. The input images were in full size 8 to 12 Megapixels with $4:3$ aspect ratios. The middle portion of the image was then squarely cropped out and down-sampled to 256×256. To change the field of view, the middle portion of the input image could be altered before being down-sampled. For instance, input parameter of 0.5 signifies that 50% of the center portion is cropped. 0.7 means that 70% of the center portion is cropped. 50% crop results in a smaller field of view where 70% crop gives a bigger field of view.

4.4.3.3 Robustness Test Results

To evaluate the robustness, the trained ConvNet was then tested on generated robustness dataset. The performance was evaluated on how many correct predictions for each wood type and respective robustness augmentations of 20 images. Accuracies were calculated for rotations, resizing, blurring, brightness and colour temperatures for each wood type. This shows the robustness of classifying wood types. Wrong predictions are tabulated in confusion matrix along with top confusions for each wood types. Overall accuracies for all testing criteria are summed and the average was calculated to show the robustness of classifier.



Wood Type	**Rotations**	**Resizing**	**Blurring**	**Brightness**	**Colour Temperature**	**Overall**
Wood X	Ri / 72 x 100%	Ry / 5 x 100%	Bi / 5 x 100%	By / 7 x 100%	Ct / 2 x 100%	(Ri + Ry + Bi + By + Ct) / 1820 x 100%
Overall Performance	Mean of Ri in column	Mean of Ry in column	Mean of Bi in column	Mean of By in column	Mean of Ct in column	Robustness Index

TABLE 4.12: The sample generated report of robustness test.

Table 4.12 shows the sample of generated robustness test report. The table header shows wood type, tests and overall column. The wood type column shows the number of wood types included in the test. Every new row stores information for new wood type.

The following columns show individual tests result of correct predictions. For example, the next column shows the rotation accuracy of wood type. The calculation is the percentage of correct predictions of images generated for rotations.

The same calculations are applied to next tests, namely resizing, blurring, brightness and colour temperature.

The last column shows the overall summary of each row of wood types. This column summarizes the performance of wood types on robustness by calculating the percentage of correct predictions of all tests.

The last row of Table 4.12 shows the overall statistic for each column. This row sums up the overall test result of particular test by calculating the mean. For instance, in the rotation test column, the overall result is the average of rotation test of all wood types. This last row of results is essential to identify the best and the worst robustness of classifier. The last column and last row shows the overall Robustness Index where it averages all the test results in the last row. The last row of the table is the overall results that is crucial for comparison of classifiers.

4.5 Summary

This chapter discussed the overall methodology of this research that covered data collection and verification, dataset compilation, design of ConvNet and evaluation criteria for classifier.

The next chapter will discuss on the result with respective discussions.

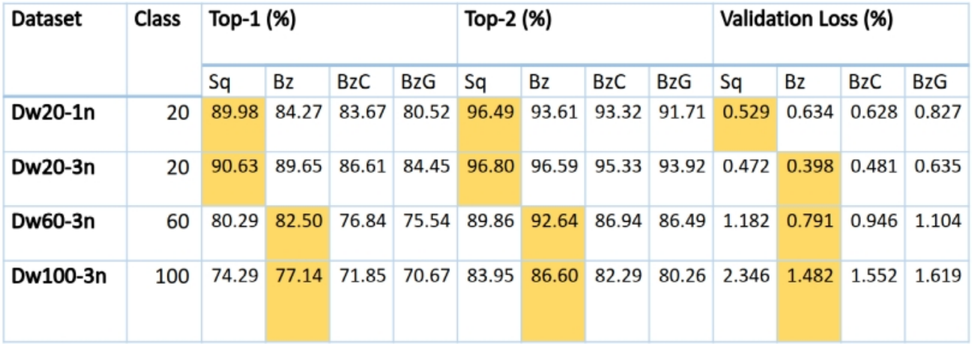
CHAPTER 5

RESULT AND DISCUSSION

This chapter discusses the results. First, the result for validation test as per Section 4.4.2 is shown and discussed followed by robustness test as per Section 4.4.3. Results on all datasets were shown, namely DW-20-1n, DW-20-3n, DW-60-3n and DW-100-3n as shown in Table 4.4. However, the largest dataset DW-100-3n will be emphasized in discussions because the corresponding classifier is deployed in service.

5.1 Validation Test Result

5.1.1 Accuracy



Dataset	Class	Top-1 (%)				Top-2 (%)				Validation Loss (%)			
		Sq	Bz	BzC	BzG	Sq	Bz	BzC	BzG	Sq	Bz	BzC	BzG
Dw20-1n	20	89.98	84.27	83.67	80.52	96.49	93.61	93.32	91.71	0.529	0.634	0.628	0.827
Dw20-3n	20	90.63	89.65	86.61	84.45	96.80	96.59	95.33	93.92	0.472	0.398	0.481	0.635
Dw60-3n	60	80.29	82.50	76.84	75.54	89.86	92.64	86.94	86.49	1.182	0.791	0.946	1.104
Dw100-3n	100	74.29	77.14	71.85	70.67	83.95	86.60	82.29	80.26	2.346	1.482	1.552	1.619

TABLE 5.1: Top-1 accuracy, Top-2 accuracy and validation loss for classifiers. NOTE: Highlighted cells are the highest among classifiers.

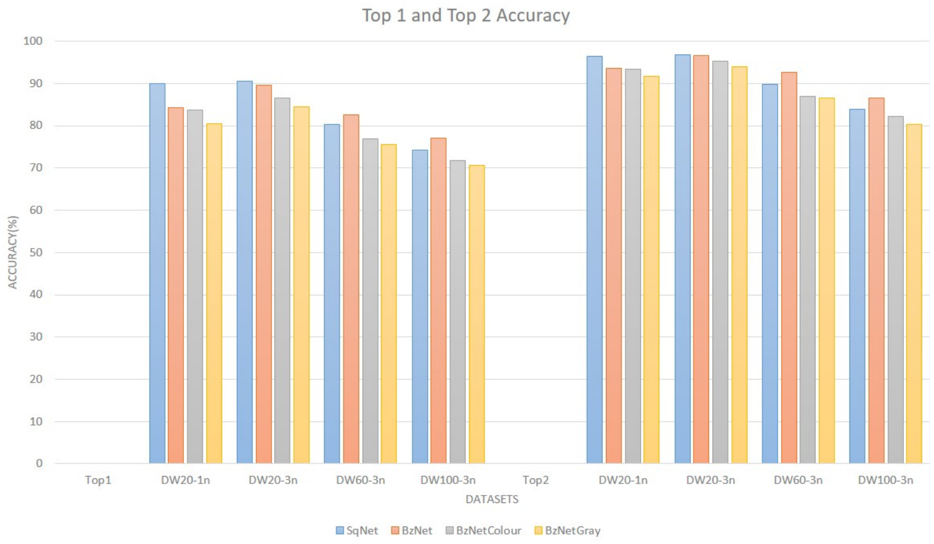


FIGURE 5.1: Overall Top-1 and Top-2 Accuracy for classifiers on various datasets. Top-1(Left) and Top-2(Right).

Table 5.1 tabulated overall Top-1, Top-2 accuracy of validation test on classifiers as well as validation loss. The results were plotted in Figure 5.1.

In 20-class datasets namely DW-20-1n and DW-20-3n, SqNet in overall recorded the best Top-1 and Top-2 accuracy. BzNet settled at second place followed by BzNet Colour and BzNet Gray. In DW-20-1n dataset, SqNet recorded 89.98% and 96.49% for Top-1 and Top-2 accuracies respectively. The difference between SqNet and second-placed BzNet is 5.71% on Top-1 accuracy. Besides that, SqNet recorded the lowest validation loss at 0.529%.

Moving on to DW-20-3n, which was three times larger than DW-20-1n dataset, SqNet still dominated with Top-1 and Top-2 accuracies of 90.63% and 96.80% followed by BzNet at 89.65% and 96.59%. The difference in Top-1 between SqNet and BzNet reduced to 0.98%, down from 5.71% in previous dataset. As for validation loss, BzNet was the lowest at 0.398.

One observation in the 20-class dataset was that the effect of pretrained weights of SqNet. All networks with exception of SqNet are trained from scratch, i.e. new weights were initialized at the beginning of training. Due to this advantage, SqNet generally performed better than other networks in terms of classification accuracy. However, as the dataset became larger, the advantage seemed diminishing as second-placed BzNet was closing in with 0.98% difference in accuracy.

On the 60-class dataset, DW-60-3n, BzNet recorded the best Top-1 and Top-2 accuracies at 82.50% and 89.86% respectively. The former recorded lower validation loss as well. SqNet, previously the best classifier in 20-class datasets settled at second place with 80.29% and 89.86% Top-1 and Top-2 accuracies. The gap between the former and latter classifier in Top-1 accuracy was 2.21%. As for validation loss, BzNet still recorded the lowest at 0.791.

As dataset grew larger in terms of number of class and number of training images, the effect of pretrained weights of SqNet lost its advantage. BzNet recorded the best accuracy while achieving the lowest validation loss.

Moving on to the largest dataset, DW-100-3n, the trend continued as BzNet recorded the highest Top-1 and Top-2 accuracies at 77.17% and 86.60%. SqNet was at second with Top-1 and Top-2 accuracies of 74.29% and 83.95%. As for validation loss, BzNet was the lowest at 1.482. The difference between BzNet and SqNet in Top-1 accuracy was 2.85% which translated to about 8,550 images in the validation dataset of 300,000 images.

Besides BzNet and SqNet, the remaining classifiers namely BzNet Colour and BzNet Gray fall into third and fourth place in all datasets in terms of Top-1 and Top-2 accuracies. As for validation loss, these two classifiers recorded lower loss than SqNet in DW-60-3n and DW-100-3n datasets. The lower loss in validation did not translate to higher accuracy in this case. This was partly explainable due to the calculation of validation loss which was the cumulative difference of distance of predicted label and ground truths label. It is possible that the predicted label was correct but the difference of distance was huge.

One common similarity between BzNet Colour and BzNet Gray was that the first convolution layer of these networks works on single channel image in grayscale. The difference was how the conversion from 3 channels RGB image to single channel image took place. For BzNet Colour, the conversion was done by learnable convolution layer whereby BzNet Gray applied simple grayscale conversion. To certain extent, the colour information of images was neglected by the networks. Wood type colour could be a crucial discriminative feature that separated certain wood types. Hence, these two networks fall behind BzNet and SqNet when they comes to validation accuracy.

In short summary in validation test, BzNet recorded the best Top-1 and Top-2 accuracies at 77.17% and 86.60% in DW-100-3n dataset with the lowest validation loss of 1.482. SqNet was second, BzNet Colour was third followed lastly by BzNet Gray in accuracy. Pretrained weights gave advantage to SqNet in smaller dataset.

5.1.2 Recall, Precision and F1 Score

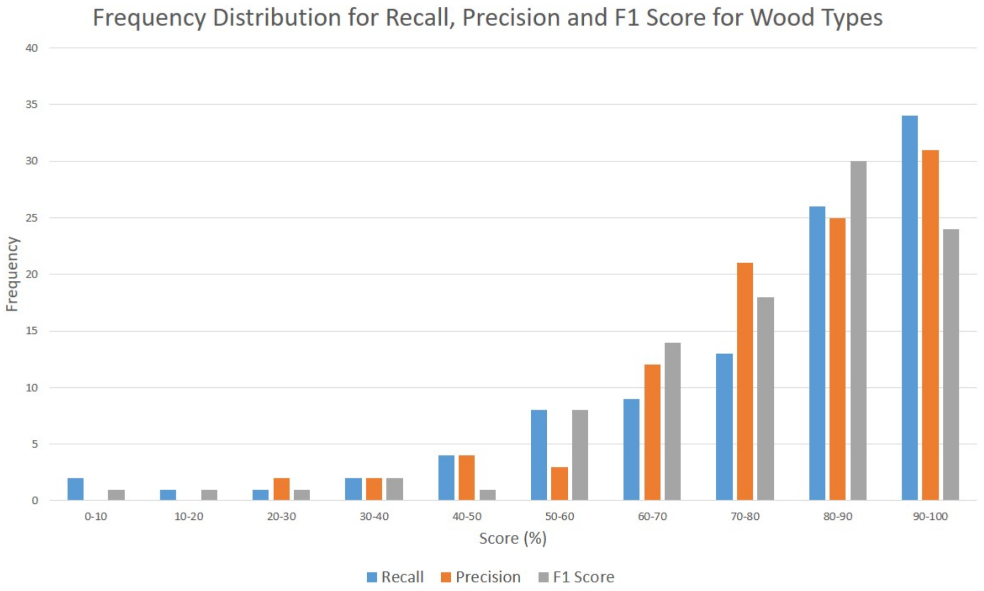


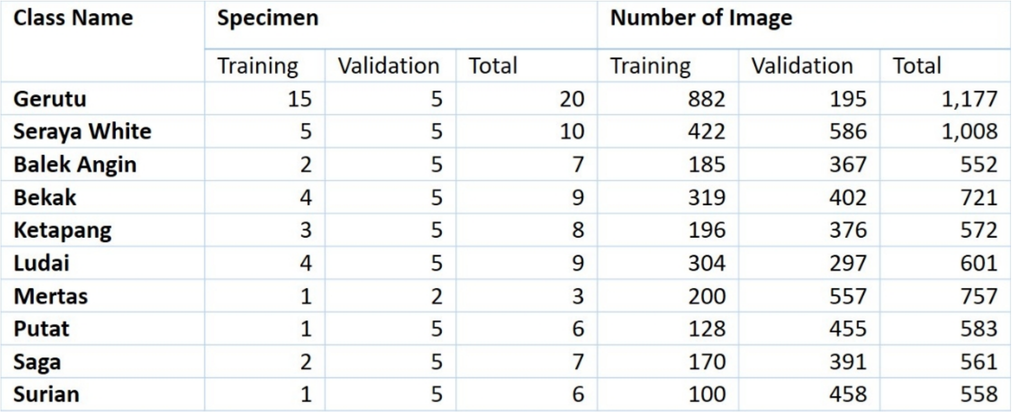
FIGURE 5.2: Frequency distribution for recall, precision and f1 score for validation test for BzNet in DW-100-3n dataset.

The previous section discussed the results on overall accuracy of classifiers. This section focuses on the detail on BzNet performance in DW-100-3n dataset. Figure 5.2 shows the frequency distribution of scores for recall, precision and f1 score for each wood types in validation test.

At first glance on Figure 5.2, one observation was that the majority of wood types fall on the right side of the figure which were the groups of 70% and above. Upon closer inspection, there were 73 out of 100 wood types scoring 70% and above in recall where 34 of these wood types scored 90% and above. This translated to 73% of wood types achieved 70% and above in recall.

The same trend applied to precision and f1 score. In the 70% and above groups, 77 wood types fall into these groups for precision. For f1 score, 72 wood types were 70% and above. In the 90% and above group, there are 31 wood types for precision and 24 wood types for f1 score. On the other side of low performers, there were 6 wood types scoring less than 50% in precision. Lastly, there are 10 wood types achieved 50% and lower for f1 score.

5.1.2.1 Low Performer in Recall



Class Name	Specimen			Number of Image		
	Training	Validation	Total	Training	Validation	Total
Gerutu	15	5	20	882	195	1,177
Seraya White	5	5	10	422	586	1,008
Balek Angin	2	5	7	185	367	552
Bekak	4	5	9	319	402	721
Ketapang	3	5	8	196	376	572
Ludai	4	5	9	304	297	601
Mertas	1	2	3	200	557	757
Putat	1	5	6	128	455	583
Saga	2	5	7	170	391	561
Surian	1	5	6	100	458	558

TABLE 5.2: Detail of low recall wood types in validation test on number of specimens and number of images.

Based on Figure 5.2, there were 10 wood types scores 50% and below in recall. These wood types were tabulated in Table 5.2 with details of number of specimens and number of images.

Based on Table 5.2, the number of specimens were below 10 averagely. Generally, these wood types were less commonly found wood types in Malaysia and there were subject to availability in FRIM Xylarium.

With cross-validation setup as per Section 4.2.1, 5 specimens were used for validation test dataset with the exception of **Mertas** as the total number of specimen was less than 5. After this setup, the training specimens were left with less than 5 specimens where in some classes the training specimens were less than the validation specimens. This partly explained the low performance of these wood types as there were low correlation between training and validation specimens. Moreover, there were variations in terms of ages for the specimens.

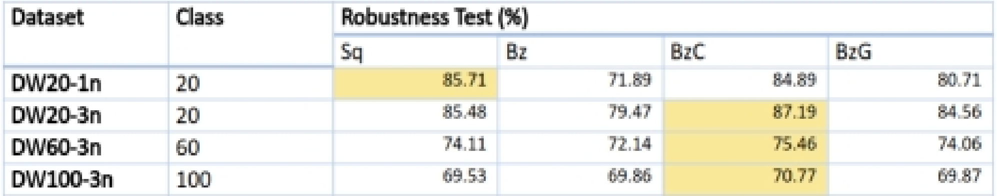
In order to improve recalls on these wood types, there are a few proposals. Firstly, it is impossible to train a classifier where the training samples are fewer than the validation samples while achieving good performance. Hence, the cross-validation setup should be rerun with emphasis on these wood types to distribute the specimens in such a way that the training specimens are higher than the validation specimens. This method can partly alleviate the problem of low correlation in training and validation specimens.

Besides that, one can increase the number of real specimens by collecting more specimens. However, this method is only applicable if the xylarium comes across new physical specimens.

The one outlier of these low performing wood types was **Gerutu**. The number of training specimens are larger than the validation specimens with a combined specimens of 20. However, upon analysis on this wood type, the confusions are mainly from the same family of wood types. Wood types from the same scientifically family shares similar macroscopic wood features. It can only be differentiated confidently when certain unique features are observed. This applies to wood anatomists as well.

5.2 Robustness Test Result

5.2.1 Overall Robustness Test



Dataset	Class	Robustness Test (%)			
		Sq	Bz	BzC	BzG
DW20-1n	20	85.71	71.89	84.89	80.71
DW20-3n	20	85.48	79.47	87.19	84.56
DW60-3n	60	74.11	72.14	75.46	74.06
DW100-3n	100	69.53	69.86	70.77	69.87

TABLE 5.3: Overall robustness test for classifiers on various datasets. NOTE: Highlighted cell represents the best score in each dataset.

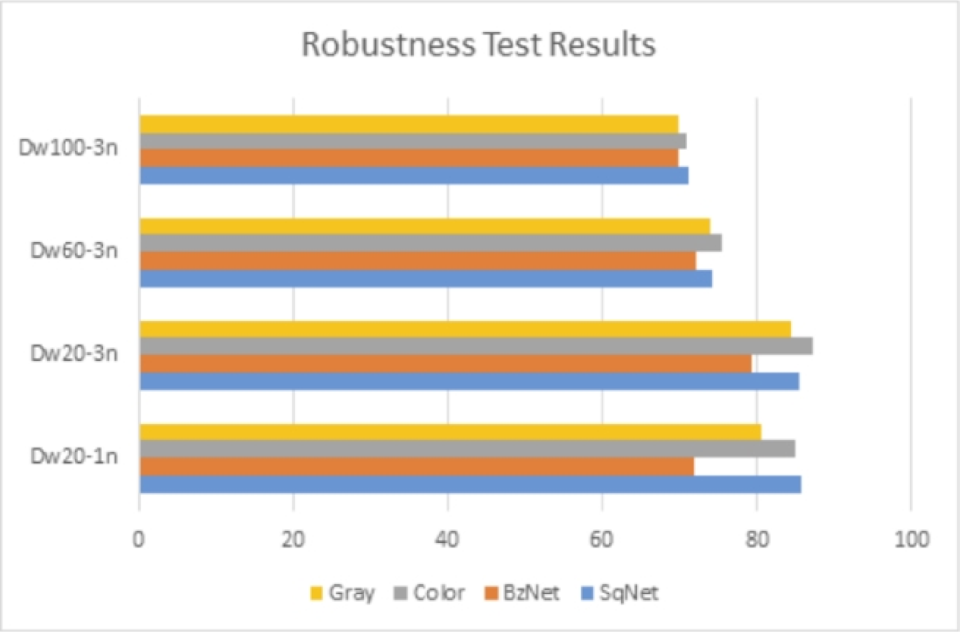


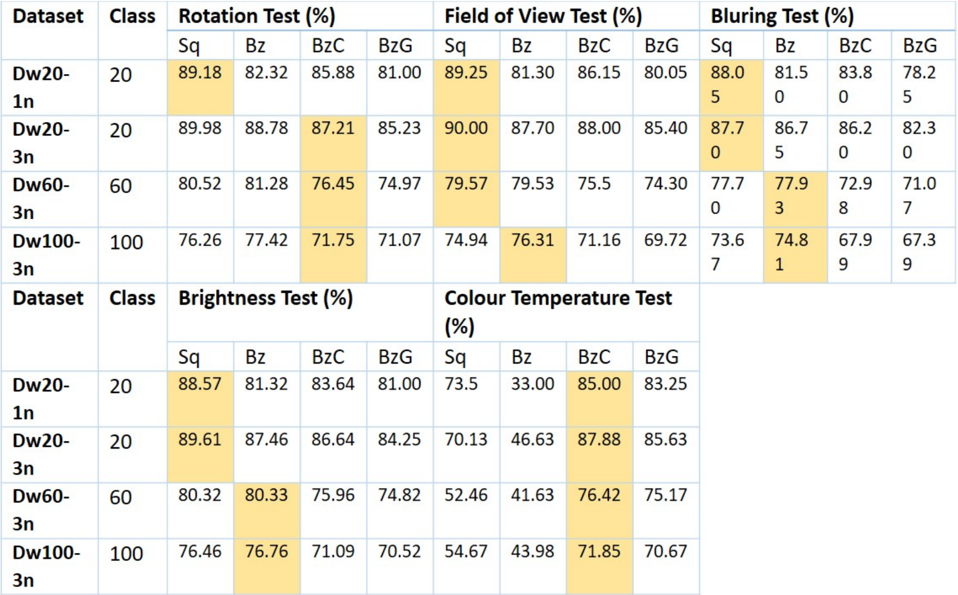
FIGURE 5.3: Overall Top-1 and Top-2 Accuracy for classifiers on various datasets. Top-1(Left) and Top-2(Right).

Table 5.3 tabulated the summary of overall robustness test for classifier on various datasets. The results were also plotted in Figure 5.3 for easy visualization.

For overall robustness test, on DW20-1n dataset, SqNet recorded the best performance at 85.71% followed by BzNet Colour. Pretrained weights gave advantage to SqNet once again. However, BzNet Colour recorded the best performance in robustness test for the remaining datasets.

BzNet, despite being a good performer in validation test in Section 5.1.1 scored poorly in 20-class datasets, grabbing the last place. On the 100-class dataset, all of the classifiers record scored within the margin of 1.27%. However, in the 100-class dataset, BzNet grabs third place where SqNet recorded the worst score. To further analyze robustness test, individual tests are discussed in the following Section 5.2.2.

5.2.2 Robustness Individual Test



Dataset	Class	Rotation Test (%)				Field of View Test (%)				Bluring Test (%)			
		Sq	Bz	BzC	BzG	Sq	Bz	BzC	BzG	Sq	Bz	BzC	BzG
Dw20-1n	20	89.18	82.32	85.88	81.00	89.25	81.30	86.15	80.05	88.05	81.50	83.80	78.25
Dw20-3n	20	89.98	88.78	87.21	85.23	90.00	87.70	88.00	85.40	87.70	86.75	86.20	82.30
Dw60-3n	60	80.52	81.28	76.45	74.97	79.57	79.53	75.5	74.30	77.70	77.93	72.98	71.07
Dw100-3n	100	76.26	77.42	71.75	71.07	74.94	76.31	71.16	69.72	73.67	74.81	67.99	67.39

Dataset	Class	Brightness Test (%)				Colour Temperature Test (%)			
		Sq	Bz	BzC	BzG	Sq	Bz	BzC	BzG
Dw20-1n	20	88.57	81.32	83.64	81.00	73.5	33.00	85.00	83.25
Dw20-3n	20	89.61	87.46	86.64	84.25	70.13	46.63	87.88	85.63
Dw60-3n	60	80.32	80.33	75.96	74.82	52.46	41.63	76.42	75.17
Dw100-3n	100	76.46	76.76	71.09	70.52	54.67	43.98	71.85	70.67

TABLE 5.4: Detail of individual robustness test results. NOTE: Highlighted cells represent the best score in each test.

Table 5.4 tabulated the details of individual robustness test, a total of 5 tests as defined in Section 4.4.3. The tests will be discussed in sequence as follows.

1. Rotation Test
2. Field of View Test
3. Blurring Test
4. Brightness Test
5. Colour Temperature Test

5.2.2.1 Robustness Rotation Test

Firstly, Rotation Test which was set to investigate the robustness of classifier against images with different rotations. In 20-class datasets, SqNet recorded the best score of 89.18% and 89.98%. In DW-20-1n dataset, BzNet scored last with a margin of 6.86% behind SqNet. BzNet Colour was in second place followed by BzNet Gray in third. However, BzNet closed the gap between leading SqNet to 1.2%, clinching second place in DW-20-3n dataset. BzNet Colour took third place followed by BzNet Gray. The higher the scores mean that the classifier is robust against rotations in images.

Moving to DW-60-3n dataset, BzNet grabbed the first place at 81.28% followed by SqNet at 80.52%, BzNet Colour at 76.45% and BzNet Gray at 74.97%. This trend was observed in the validation test. The same trend continued in the DW-100-3n dataset.

5.2.2.2 Robustness Field of View Test

In the next test, Field of View Test was to study the effect of changing field of view on classifier, i.e. zooming in and out. In this test, SqNet recorded best scores with a margin of 2-3% for 20-class datasets. In the DW-60-3n dataset, SqNet remained the best with a marginal lead of 0.04% over second-placed BzNet. However, BzNet scored the best result in 100-class DW-100-3n dataset. As per rotation test, BzNet Colour and BzNet Gray sat in third and fourth place respectively in all datasets with exception on DW-20-1n where BzNet came last.

5.2.2.3 Robustness Blurring Test

Blurring Test was to simulate limited in-focus of the object within the image. As per tests above, same trend was observed where SqNet led in 20-class datasets whereas BzNet recorded best results in 60-class and 100-class datasets. As opposed to previous tests above, BzNet did not come last in any dataset, the worst place was third in DW-20-1n dataset. In similar fashion, third and fourth place went to BzNet Colour and BzNet Gray.

5.2.2.4 Robustness Brightness Test

Brightness test studied the effect of varying brightness level displayed in images on classifier. As usual, SqNet performed better in 20-class and BzNet performed better in remaining datasets. As observed in Blurring Test, BzNet did not come last in any dataset where the worst being third place in DW-20-1n dataset. Upon closer observation in DW-60-3n and DW-100-3n dataset, second-placed SqNet was marginally behind BzNet.

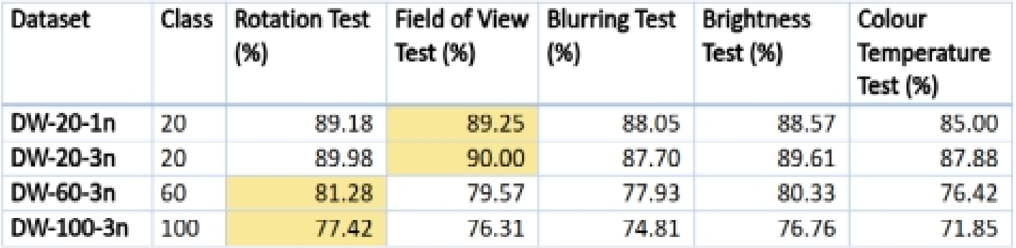
5.2.2.5 Robustness Colour Temperature Test

Lastly, the final test is Colour Temperature Test. This test put the classifier to test on warm and cold temperature images. The preceding trends as observed in previous four tests did not apply in this test. BzNet Colour recorded the best scores for all datasets followed by BzNet Gray. BzNet grabbed last spot in all datasets.

Without colour information, the classifier was less likely to be confused by colour temperature of image as demonstrated by BzNet Colour and BzNet Gray. BzNet Colour with its learnable Colour Conversion Module as defined in Section 4.3.3.1, showed its advantage over BzNet Gray where it learned the best colour conversion.

On the other hand, colour information plagued SqNet and BzNet in this test. The same useful information used to differentiate some wood types was confusing the classifier when the colour temperature shifted heavily from warm to cold.

5.2.2.6 Robustness Test Summary



Dataset	Class	Rotation Test (%)	Field of View Test (%)	Blurring Test (%)	Brightness Test (%)	Colour Temperature Test (%)
DW-20-1n	20	89.18	89.25	88.05	88.57	85.00
DW-20-3n	20	89.98	90.00	87.70	89.61	87.88
DW-60-3n	60	81.28	79.57	77.93	80.33	76.42
DW-100-3n	100	77.42	76.31	74.81	76.76	71.85

TABLE 5.5: Top scores of individual robustness test. NOTE: Highlighted cells represent the best score in each dataset.

Table 5.5 tabulated the highest score of every individual robustness test in all datasets. This helps to understand classifiers regardless of architecture perform in robustness test as a whole.

In general, classifiers were weak in Colour Temperature Test as it recorded the least accuracy in all datasets. Next, Blurring Test was one place ahead of Colour Temperature Test. This was expectable as deliberately blurred images reduce image quality as well as losing important information graphically that could be useful discriminative feature for classifier. It was glad to find out that classifiers fared relatively good in Brightness, Field of View and Rotation Tests. Some degrees of varying brightness did not severely affect the classification performance.

BzNet recorded the best Top-1 and Top-2 accuracies in validation test however it was not the top performer in robustness test especially in the Colour Temperature Test. However, on 60-class and 100-class datasets in robustness test, it was relatively well performing in other individual tests.

As for SqNet, the pretrained weights on 1000-class ImageNet dataset gave it the robustness on robustness tests. It was performing relatively good on individual tests scoring averagely on second place.

On the other hand, BzNet Colour topped Colour Temperature Test which made it robust against shifting of colour temperature in images. However, BzNet Colour was third best in validation test followed by BzNet Gray in fourth position.

Colour information played an important role in the discrepancy BzNet Colour and BzNet Gray in validation and robustness test. Colour information gave the edge to BzNet and SqNet in discriminating some wood types hence both these networks top validation test. It was worth noting that images in validation test are randomly augmentated with combination of augmentation methods but they were less likely to be in extreme condition. In the robustness test dataset, images were augmented to every parameters defined in Table 4.11 which includes extreme conditions. At the same time, this information introduced confusion when the colour temperature was heavily shifted. This was the opportunity for BzNet Colour and BzNet Gray to shine.

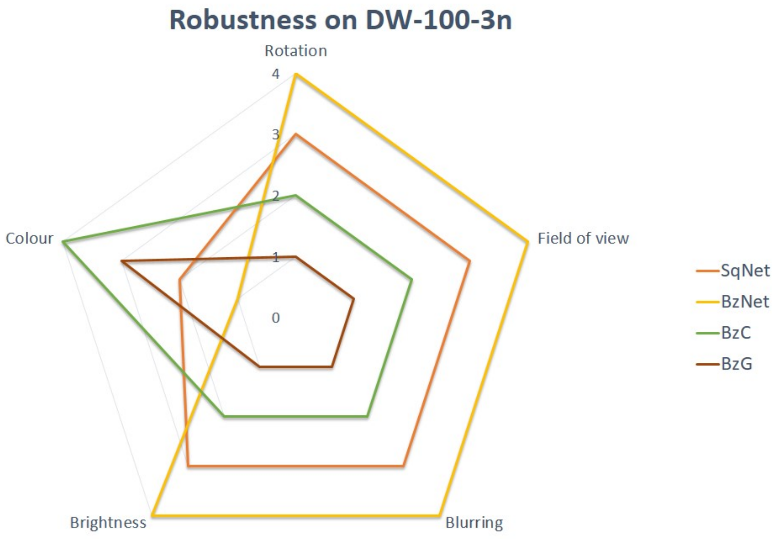


FIGURE 5.4: Strength comparison of classifier on robustness test on 100-class dataset, DW-100-3n.

Figure 5.4 shows the strength of classifiers on the DW-100-3n dataset. The top performer in each test was given a rank of 4, followed by 3 for second place and down to 2 and 1. This figure shows how classifiers fare against each other in every robustness test.

One observation was that BzNet clinched top spots for every test except Colour Temperature Test which was its weak spot. The second best classifier went to SqNet followed by BzNet Colour in third place. As per previous discussion, BzNet Colour scored highest accuracy in Colour Temperature Test.

Judging from the combination result of validation test and robustness test in DW-100-3n dataset, BzNet was chosen to be the best classifier. Colour Temperature Test simulates the worst case scenario which is less probably to encounter in real world field deployment.

In a summary, BzNet scored the best in validation test and most robustness test with SqNet being the second best candidate.

5.3 Error Analysis

This section discusses the errors made by BzNet in particularly DW-100-3n dataset and their corresponding solution. There were two major errors namely natural error and robustness error. The main discussion lied in the errors that are either image content-bound or classifier-bound which are discussed in the following sections.

5.3.1 Natural Error

Natural error or same family error was defined as error that human wood anatomist faces in macroscopic wood identification. In other words, it is sensible error in macroscopic wood identification. This error is not preventable even with deep learning algorithm as it relates to the content of the captured image. Hence, this error is regarded as image content-bound.

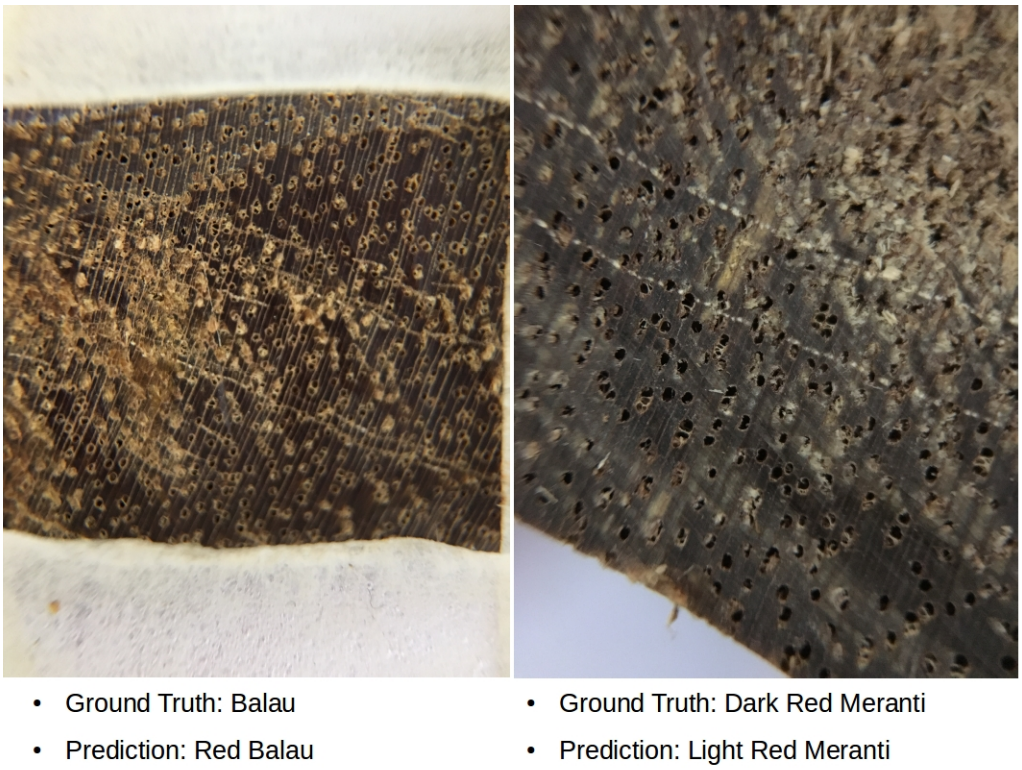
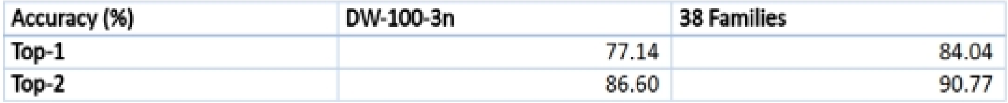

• Ground Truth: Balau
• Prediction: Red Balau
• Ground Truth: Dark Red Meranti
• Prediction: Light Red Meranti

FIGURE 5.5: Example of natural error. Left image is Balau but predicted as Red Balau. Right image is Dark Red Meranti, but predicted as Light Red Meranti.

Appendix B shows wood types grouped together within their respective scientific family group of 38 families. Genus within the same family shares some similar macroscopic features with some minor differences. In a analogy to human family, siblings often shares lookalike faces. Can they be differentiated confidently? The answer is yes when distinctive features are present in the context. However, the confidence level is greatly depends on the present of unique discriminative features in the observation. The same goes to wood types. In a way, natural error can be considered as the limitation of macroscopic wood identification as the macroscopic level cannot differentiate wood types confidently without resolving to microscopic level. Figure 5.5 shows examples of natural error within the family of *Dipterocarpaceae*. Balau, Red Balau, Dark Red Meranti and Light Red Meranti are within the family of *Dipterocarpaceae* which shares similar macroscopic features.

According to inputs from wood anatomists in FRIM, some wood types can be identified physically using quantities such as weight and odour. However, these physical measures were absent in the context of image. Besides that, when in doubt, wood anatomist would get a few different cut on the wood specimens to expose more macroscopic feature with the hope of finding unique wood type defining macroscopic feature. However, this was often not the case for the training and validation data collected. Not every single images contain uniquely discriminative features.



Accuracy (%)	DW-100-3n	38 Families
Top-1	77.14	84.04
Top-2	86.60	90.77

TABLE 5.6: Top-1 and Top-2 accuracy on DW-100-3n and 38 families for BzNet

Table 5.6 shows Top-1 and Top-2 accuracies on DW-100-3n and 38 families for BzNet. For Top-1 accuracy, the gap between wood types and families is 6.90% which was the same family confusion error. In other words, 6.90% of wood types predicted did not match the ground truths but they were within the same family. On the families classification, these predictions were correct.

On 38-family classification, the difference between Top-1 and Top-2 accuracy was 6.73%. This difference can be regarded as different family error. This happened when the wood types predicted was not from the same family as the ground truths and it was not natural from the perspective of wood anatomists. Robustness error may contribute to this error which is discussed in Section 5.3.2.

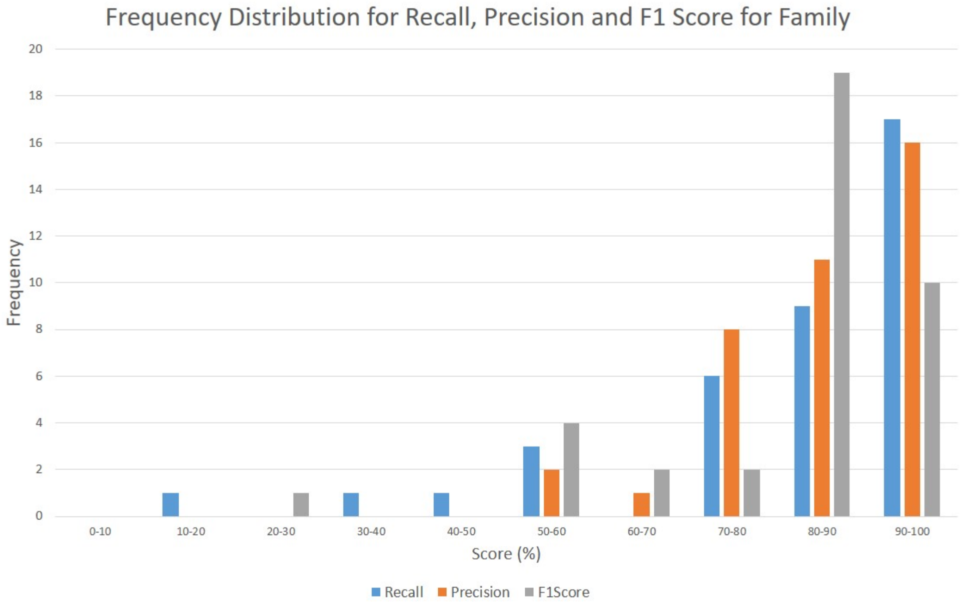


FIGURE 5.6: Frequency distribution for recall, precision and f1 score for families classification on DW-100-3n for BzNet.

Figure 5.6 shows the frequency distribution for recall, precision and f1 score for 38-families classification for BzNet. At first glance, the major families fall to the right portion of the figure which are the 70% and above groups. Closer inspection shows that 32 out of 38 families score 70% and above in recall where 17 of these families score 90% and above. On the other hand, 6 out of 38 families score 70% and below which were tabulated in Table 5.7.

From Table 5.7, there were only 3 out of 37 families scored 70% and below for precision. As for f1 score, there are 6 out 38 families scored 70% and below. 6 out these families were found in recall with the additional of *Ebenaceae* family.

TABLE 5.7: Families that achieved 70% and lower in recall, precision and f1 score

Recall	Precision	F1 Score
Tiliaceae	Tiliaceae	Tiliaceae
Combretaceae	-	Combretaceae
Ctenolophonaceae	-	Ctenolophonaceae
Lecythidadceae	-	Lecythidadceae
-	Ebenaceae	Ebenaceae
Lauraceae	Lauraceae	Lauraceae
Meliaceae	-	Meliaceae

Upon cross referencing with Appendix B and low recall performer of wood types in Section 5.1.2.1, those wood types with low performance in recall belonged to the 6 low performing families. As discussed in Section 5.1.2.1, the same problems plagued the recall of these families.

5.3.2 Robustness Error

Robustness error refers to classifier errors in varying conditions as defined in Section 4.4.3. The inability of classifier to be invariant to conditions happen in the images is defined as robustness error. This error is strictly classifier-bound and it is irrelevant to the contents captured in the image.

As shown in Section 5.2, robustness error is found due to the fact that no 100% accuracy is achieved in every individual Robustness Test. The chosen classifier, BzNet suffers from Colour Temperature Test where BzNet Colour shines. This is the main challenge for this research as the author proposed the usage of smart-phone as imaging device to acquire macroscopic wood images as opposed to standardized method of image acquisition practiced in the laboratory under controlled conditions.

5.3.3 Solution and Recommendation

In this section, solutions and recommendations are discussed to improve the classifier performance against errors discussed in previous sections, namely natural error and robustness error.

There are no direct solution to solve natural error as it is the limitation of macroscopic wood identification. However, in practical usage, users are recommended to get more cut on the specimens to expose more macroscopic wood features that may confidently define a wood type.

As for robustness error, there is one solution proposed to tackle this error. This first solution is the application of ensemble networks. Multiple classifiers with different weights train to compliment each other by fusing the predictions in a hope to minimize the effect of varying conditions on image.

Besides, the author proposed that merging the best characteristics of BzNet and BzNet Colour in future ConvNet designs to help tackle the colour temperature problem.

5.4 Summary

In this chapter, the results for validation test and robustness test were presented and discussed in detailed. Besides that, error analysis on the classifier was discussed along with recommendations on improvement. The best classifier in validation test was BzNet in DW-100-3n dataset. As for robustness test, BzNet achieved the best overall performance however, it underperformed in Colour Temperature Test where BzNet Colour topped the test.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter concludes research and states future work for this research.

6.1 Conclusion

The objectives as defined in Section 1.4 are met at the end of this research. In this research, the goal is to develop a practical macroscopic wood identification system using deep learning method. At such, smart-phone with macro-lens was used as image acquisition device. With this setup, a total of 101,546 macroscopic cross section images of 100 types of wood were collected and verified by wood anatomists. These data is used as dataset to train the deep learning model which is ConvNet for this research. Besides that, a huge digitized Wood Xylarium is created from the collected data.

The focus of this research lies in the design of ConvNet to improve classification of wood types. New ConvNet architectures, BlazeNet and its variant BlazeNet Colour Conversion were designed and benchmarked against SqueezeNet in two tests namely validation test and robustness test. In the 100-class dataset, DW-100-3n, BzNet achieved the best Top-1 accuracy of 77.14% tested on the validation dataset that consists of 300,000 images. The benchmark SqNet achieved 74.29% on the same dataset, 2.85% behind the leader BzNet. On the robustness test, BzNet grabbed the third place behind BzNet Colour and BzNet Gray with a close margin of 0.91% and 0.01%. The major downfall of BzNet was the Colour Temperature Test where its variant BzNet Colour triumphed. In comparison to

SqNet, BzNet was better in both validation and robustness test. Moreover, BzNet is 3x smaller in terms of number of parameters.

The main errors faced by BzNet in this research was the same family error and robustness error. Same family error refers to the predicted wood types are in the same scientific family of the ground truth wood types. This is a common error in the macroscopic wood identification as per wood anatomist. Besides that, robustness test showed that the classifier suffered from robustness error.

In conclusion, this outcome of this research is a deployed and running cloud service of 100 Malaysian Timber Identification system. The system is accessible via an iOS application, namely Mywood-ID published in Apple Appstore. BzNet is deployed as the classifier in the cloud server.

6.2 Future Work

The research presented in this thesis solved the problem defined in Section 1.3. However, it created some works to be done in the future to improve current system and future research direction.

Firstly, the performance of BzNet suffered from effect of varying colour temperature on image despite being the best classifier in validation test. BzNet Colour was designed to mitigate this problem but it partly solved the problem. The first recommendation is to combine the BzNet and BzNet Colour to form an ensemble classifier by fusing outcomes from two classifiers to get the best of both world to achieve great accuracy in validation test and robustness test.

As discussed in Section 5.1.2.1, some of the wood types that suffered from bad recall were having less number of training specimens compared to validation specimens. To mitigate this problem, the cross validation setup of dataset has to be rerun with emphasis on these wood types to transfer some of the validation specimens to become training specimens. The rule of thumb is that the number of validation specimens should be lower than the training specimens.

Besides the improvements above, this research created some new research directions in the future. Current cloud based system is good in accessibility due to the wide coverage of the Internet. However, there is real use case that the system is unreachable in area with limited Internet coverage. Hence, the future direction is to embed the classifier into the smart-phone to make offline service available. However, this is subject to the processing power of current and future smart-phone.

Lastly, dataset is crucial to this research. To improve the system, new data is needed to increase the size and variation of dataset. To solve this problem, Deep Convolutional Generative Adversarial Network (DCGAN) is proposed to learn from current available dataset to generate new data. DCGAN consists of two parts, namely discriminator and generator. Discriminator is analogous to classifier that learns to discriminate the input into their respective classes. Generator, on the other hand, learns to generate data that are closely resemble the real input data. A robust generator can generate data that in turn used to training a new classifier.

LIST OF REFERENCES

Anonymous. 2009. *Wood Identification Using Macroscopic Features*. Forest Research Institute Malaysia. Available at https://info.frim.gov.my/woodid/e-woodID-introduction7Dec09.pdf.

Anonymous. 2016. *Best Practice Guide for Forensic Timber Identification United Nations Office on Drugs and Crime*.

Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. 2007. Greedy layer-wise training of deep networks. In: *Advances in neural information processing systems*. pp. 153–160.

Cavalin, P. R., Kapp, M. N., Martins, J. and Oliveira, L. E. 2013. A multiple feature vector framework for forest species recognition. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, pp. 16–20.

de Paula Filho, P. L., Oliveira, L. S. and Britto Jr, A. S. 2009. A database for forest species recognition. In: *Proceedings of the XXII Brazilian Symposium on Computer Graphics and Image Processing*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 248–255.

Duda, R. O., Hart, P. E., Stork, D. G. *et al.* 1973. *Pattern classification*, vol. 2. Wiley New York.

Gasson, P. 2011. How precise can wood identification be? wood anatomys role in support of the legal timber trade, especially cites. *IAWA Journal* 32(2), pp. 137–154.

Hafemann, L. G., Oliveira, L. S. and Cavalin, P. 2014. Forest species recognition using deep convolutional neural networks. In: *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, pp. 1103–1107.

He, K., Zhang, X., Ren, S. and Sun, J. 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778.

Henderson, F. 1964. Identification of hardwoods: a lens key. *Forest Products Research* 25.

Hermanson, J., Wiedenhoeft, A. and Gardner, S. 2013. *A Machine Vision System For Automated Field-level Wood Identification*. Global Timber Tracking Network. Presentation at Regional Workshop for Asia, Pacific and Oceania on Identification of Timber Species and Origins, Beijing China.

Hinton, G. E., Osindero, S. and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7), pp. 1527–1554.

Huang, G., Liu, Z., Weinberger, K. Q. and van der Maaten, L. 2017. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. vol. 1, p. 3.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. and Keutzer, K. 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360* .

Ilic, J. *et al.* 1991. *CSIRO atlas of hardwoods.* Springer-Verlag.

Khalid, M., Lee, E. L. Y., Yusof, R. and Nadaraj, M. 2008. Design of an intelligent wood species recognition system. *International Journal of Simulation System, Science and Technology* 9(3), pp. 9–19.

Koch, G., Haag, V., Heinz, I., Richter, H.-G. and Schmitt, U. 2015. Control of internationally traded timber-the role of macroscopic and microscopic wood identification against illegal logging. *Journal of Forensic Research* 6, p. 317.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp. 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1(4), pp. 541–551.

Lee, J. Y. 2016. *Mobile Wood Identification System Using Deep Learning Algorithm.* University Tunku Abdul Rahman.

Nasirzadeh, M., Khazael, A. A. and bin Khalid, M. 2010. Woods recognition system based on local binary pattern. In: *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*. IEEE, pp. 308–313.

on International Trade in Endangered Species of Wild Fauna, C. and Flora. 2002. *CITES identification guide-tropical woods: guide to the identification of tropical woods controlled under the Convention on International Trade in Endangered Species of Wild Fauna and Flora.* Environment Canada= Environnement Canada.

Ravindran, P., Costa, A., Soares, R. and Wiedenhoeft, A. C. 2018. Classification of cites-listed and other neotropical meliaceae wood images using convolutional neural networks. *Plant methods* 14(1), p. 25.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature* 323(6088), p. 533.

Siew, K. F., Tang, X. J. and Tay, Y. H. 2017. Improved convolutional networks in forest species identification task. In: *Second International Workshop on Pattern Recognition*. International Society for Optics and Photonics, vol. 10443, p. 104430C.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1), pp. 1929–1958.

Srivastava, R. K., Greff, K. and Schmidhuber, J. 2015. Highway networks. *arXiv preprint arXiv:1505.00387* .

Tou, J. Y., Lau, P. Y. and Tay, Y. H. 2007. Computer vision-based wood recognition system. In: *Proceedings of International workshop on advanced image technology*. Citeseer.

CHAPTER A

TIMBER TYPES DETAILS

TABLE A.1: Table of timber types collected

Class Name	Specimen			Number of Image		
	Training	Validation	Total	Training	Validation	Total
Ramin	14	5	19	986	381	1,367
Karas	6	5	11	416	653	1,069
Merbau	10	5	15	878	678	1,556
Kempas	13	5	18	1,265	393	1,658
Rubberwood	3	5	8	675	410	1,085
Balau	25	5	30	1,543	255	1,798
Chengal	22	5	27	842	226	1,068
Keruing	70	5	75	1,737	116	1,853
Dark Red Meranti	35	5	40	1,600	200	1,800
Light Red Meranti	58	5	63	1,73	102	1,375
Jelutong	3	2	5	741	322	1,063
Red Balau	25	5	30	1,221	100	1,321
Teak	25	5	30	1,640	478	2,118
Tualang	3	5	8	537	465	1,002
Nyatoh	45	5	50	1,020	57	1,077
Keranji	9	5	14	733	319	1,052
Bintangor	31	5	36	986	129	1,115
Mengkulang	11	5	16	711	336	1,047
White Meranti	45	5	50	1,133	69	1,202
Yellow Meranti	47	5	52	1,001	109	1,110
Giam	27	5	32	1,107	215	1,322
Kapur	25	5	30	984	196	1,180
Kedondong	53	5	58	1,578	123	1,701
Kekatong	11	5	16	752	250	1,002
Kelat	44	5	49	1,494	195	1,689
Merawan	79	5	84	1,235	77	1,312
Mersawa	33	5	38	1,069	147	1,216
Resak	43	5	48	986	121	1,107
Sepetir	15	5	20	827	328	1,155
Sesendok	6	5	11	438	613	1,051

TABLE A.2: Table of timber types collected (cont. 1)

Class Name	Specimen			Number of Image		
	Training	Validation	Total	Training	Validation	Total
Durian	26	5	31	867	166	1,033
Geronggang	7	5	12	475	558	1,033
Gerutu	15	5	20	882	295	1,177
Merpauh	10	5	15	905	296	1,201
Kasai	7	5	12	612	509	1,121
Kembang Semangkok	5	5	10	604	424	1,028
Machang	20	5	25	1,349	275	1,624
Medang	44	5	49	1,722	238	1,960
Melunak	12	5	17	771	236	1,007
Penarahan	28	5	33	1,035	180	1,215
Pulai	13	5	18	866	167	1,033
Bitis	17	5	22	720	360	1,080
Rengas	24	5	29	1,234	339	1,573
Mempisang	41	5	46	932	76	1,008
Belian	1	5	6	53	1,039	1,092
Tembusu	7	5	12	520	489	1,009
Terentang	6	5	11	511	503	1,014
Acacia Mangium	3	2	5	571	551	1,122
Keledang	34	5	39	940	226	1,166
Terap	3	5	8	325	834	1,159
Kulim	12	5	17	610	432	1,042
Kungkur	2	2	4	624	388	1,012
Mata Ulat	11	5	16	898	262	1,160
Melantai	14	5	19	755	312	1,067
Meranti Bakau	8	5	13	548	517	1,065
Perupok	21	5	26	744	296	1,040
Punah	10	5	15	668	342	1,010
Simpoh	18	5	23	935	204	1,139
Sena	3	2	5	755	269	1,024
Seraya White	5	5	10	422	586	1,008
Ara	6	5	11	634	446	1,080
Bakau	8	5	13	641	480	1,121
Balek Angin	2	5	7	185	367	552
Batai	2	5	7	150	416	566
Bekak	4	5	9	319	402	721
Berangan	5	5	10	480	590	1,070
Tampoi	3	2	5	576	285	861

TABLE A.3: Table of timber types collected (cont. 2)

Class Name	Specimen			Number of Image		
	Training	Validation	Total	Training	Validation	Total
Chempaka	11	5	16	451	227	678
Delek	3	5	8	248	461	709
Derum	6	5	11	306	318	624
Dungun	16	5	21	493	147	640
Gelam	3	2	5	341	216	557
Kandis	9	5	14	408	202	610
Kasah	0	1	1	572	572	1,144
Kayu Malam	18	5	23	874	210	1,084
Kekabu	2	2	4	153	408	561
Kelampayan	3	2	5	326	227	553
Ketapang	3	5	8	196	376	572
Leban	5	5	10	308	301	609
Mahang	5	5	10	264	315	579
Ludai	4	5	9	304	297	601
Mempening	9	5	14	908	207	1,115
Meransi	4	5	9	279	324	603
Merbatu	2	2	4	585	582	1,167
Mertas	1	2	3	200	557	757
Minyak Berok	22	5	27	535	126	661
Nyireh	2	5	7	148	411	559
Pauh Kijang	5	5	10	364	298	662
Pelawan	5	5	10	299	259	558
Penaga	1	5	6	101	503	604
Perah	5	5	10	305	303	608
Putat	1	5	6	128	455	583
Rambutan	3	5	8	191	390	581
Saga	2	5	7	170	391	561
Sawa Luka	2	5	7	301	274	575
Sentang	3	2	5	407	150	557
Surian	1	5	6	100	458	558
Surian Batu	3	2	5	441	120	561
Tempinis	1	5	6	100	476	576
Tulang Daing	6	5	11	577	510	1,087

CHAPTER B

TIMBER TYPES FAMILY DETAILS

TABLE B.1: Table of timber types family

Family Name	Trade Name	Family Name	Trade Name
Dipterocarpaceae	Balau	Moraceae	Keledang
	Chengal		Terap
	Keruing		Ara
	Dark Red Meranti		Tempinis
	Light Red Meranti	Anacardiaceae	Merpauh
	Red Balau		Machang
	White Meranti		Rengas
	Yellow Meranti		Terentang
	Giam	Myrtaceae	Kelat
	Kapur		Gelam
	Merawan		Pelawan
	Mersawa	Thymelaeaceae	Ramin
	Resak		Karas
	Gerutu	Fagaceae	Berangan
	Melantai		Mempening
	Meranti Bakau	Verbenaceae	Teak
	Seraya White		Leban

TABLE B.2: Table of timber types family

Family Name	Trade Name	Family Name	Trade Name
Leguminosae	Merbau	Lauraceae	Medang
	Kempas		Belian
	Tualang	Celastraceae	Mata Ulat
	Keranji		Perupok
	Kekatong	Sapotaceae	Nyatoh
	Sepetir		Bitis
	Kungkur	Bombacaceae	Durian
	Sena		Kekabu
	Batai	Sapindaceae	Bakau
	Tulang Daing		Meransi
	Saga		Kasai
Euphorbiaceae	Rubberwood		Rambutan
	Sesendok	Apocynaceae	Jelutong
	Balek Angin		Pulai
	Tampoi	Dilleniaceae	Simpoh
	Mahang	Olacaceae	Kulim
	Ludai	Tiliaceae	Melunak
	Perah	Annonaceae	Mempisang
Guttiferae	Bintangor	Loganiaceae	Tembusu
	Geronggang	Fabaceae	Acacia Mangium
	Derum	Proteaceae	Sawa Luka
	Kandis	Tetrameristaceae	Punah
	Penaga	Myristicaceae	Penarahan
Meliaceaes	Bekak	Burseraceae	Kedondong
	Nyireh	Anisophylleaceae	Delek
	Sentang	Chrysobalanaceae	Merbatu
	Surian	Combretaceae	Ketapang
	Surian Batu		Mertas
Sterculiaceae	Mengkulang	Ebenaceae	Kayu Malam
	Kembang	Semangkok	Putat
	Dungun	Magnoliaceae	Chempaka
	Kasah	Simaroubaceae	Pauh Kijang
Polygalaceae	Minyak Berok	Rubiaceae	Kelampayan