IMAGE CLASSIFICAION OF ALL MTU CHAPEL BOOKS USING ARTIFICIAL NEURAL NETWORK

BY

OBIKE EMMANUEL

MATRIC NUMBER: 16010301001

**BEING A PROJECT SUBMITTED IN THE DEPARTMENT OF COMPUTER SCIENCE
AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF DEGREE OF BACHELOR OF SCIENCE
MOUNTAIN TOP UNIVERSITY, IBAFO,
OGUN STATE, NIGERIA**

November, 2020

CERIFICATION

This Project titled, **IMAGE CLASSIFICAION OF ALL MTU CHAPEL BOOKS USING ARTIFICIAL NEURAL NETWORK,** prepared and submitted by **OBIKE EMMANUEL** in partial fulfilment of the requirements of the degree of **BACHELOR OF SCIENCE** (Computer Science), is hereby accepted

___________________________ (Signature and Date)

DR. I. O. AKINYEMI

Supervisor

___________________________ (Signature and Date)

DR. I. O. AKINYEMI

Head of Department

Accepted as partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE (Computer Science)

___________________________ (Signature and Date)

Prof. A. P. OLALUSI

Dean, College of Basic and Applied Sciences

DEDICATION

I dedicate this project to God Almighty.

ACKNOWLEDGEMENT

The success and final outcome of this project goes to the Almighty God for wisdom and understanding.

I specially appreciate my Supervisor Dr. I. O. AKINYEMI who took keen interest in my project work and guided me all along, and never relented to attend to me anytime I came to him for assistance.

I would like to acknowledge the Head of Department Computer Science and Mathematics Dr. I. O. Akinyemi, and owe him my deepest gratitude for the efforts, constant encouragement, guidance and support of all the academic and non-academic staff of the Department of Computer Science and Mathematics. For the teachings that have brought out positive values in me and making my stay a worthwhile one. I extend my gratitude to Mountain Top University for setting greater heights for me. I say God bless you richly

My heart-felt gratitude goes to the Dean, College of Basic and Applied Sciences – Prof. Akinwande, A. I., and all other staff members of the department of Computer Science: Dr. Oladejo, Dr. Kasali, F. A and other members of staff.

I heartily would like to thank my parents, Mr & Mrs Sinclair Obike and siblings, thank you all for your moral and financial support. I am grateful for all the investments into my education and future. I would not forget to remember all the students in the Department of Computer Science and Mathematics, for making my stay a worthwhile one, I say God bless you all richly.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This project is based on Image Classification of all Mountain Top University chapel books using artificial neural network which will be used in Mountain Top University chapel. The staff in chapel checks for student that don’t come with their chapel books, even when the students are so many, in the process time wastage comes in, with the help of artificial neural network which will be used to check student without their chapel book then, that can make students comes to the chapel with their chapel manuals.

The aim of this project is to create a system that can notify students that enters Mountain Top University chapel without their chapel book which has been made compulsory to all students to get it. To achieve this, the pictures of all the Mountain Top University chapel books will be taken for the system to recognize.

This project was built using the Jupyter Notebook environment. In the Jupyter Notebook environment, data was uploaded in the environment where the data will be trained so, the system will be able to recognize it anytime it is called.

Artificial neural network (ANN) also known as neural networks is the piece of a computing device to model the human brain's way of interpreting and processing knowledge. It is the basis of artificial intelligence (AI) and solves problems that would prove impossible or complicated by human or statistical standards.

CHAPTER ONE

INTRODUCTION

1.1 Background to the study

The deep neural network is also regarded as the feature that mimics the functions in the processing of data and of the human brain producing patterns for decision-making. A subset of artificial intelligence is deep learning in machine learning which has networks able to learn from unstructured or unlabeled knowledge. They are efficient recognizers and classifiers of patterns. They act as black-box, model-free, and adaptive instruments to capture and learn critical data structures. In the fields of prediction and estimation, pattern recognition and optimization, their computational skills have been proven. They are especially suitable for problems that are too complex for classical mathematics and conventional procedures to model and solve. (Mehrotra et al., 1997).

One of the reasons for popularity of the neural network is the development of the simple error backpropagation (BP) training algorithm which is based on a gradient-descent optimization technique Training a neural network with a supervised leaning algorithm such as BP implies using a series of training examples to find the weights of the links connecting the nodes. An error function in the form of the number of the squares belonging to the errors between the true ones which the output from the training set and the computed outputs is minimized iteratively. The law of learning or training determines how the weights are changed in each iteration.

In computer vision, classification of images relates to a phase that can identify an image by its visual content. An image classification algorithm, for example, it could be designed to say whether a picture includes a human figure or not. While for humans, there is the identification of an object that is trivial, in computer vision applications, a robust picture classification is still a big challenge. The most common neural network class used for image classification is Convolutional Neural Networks (CNNs). CNN consists of layers of input and output with multiple hidden layers in-between. It derives its name from the kind of hidden layers of which it consists. Usually, the hidden layers of a CNN consist of convolution layers, pooling layers, completely related layers and layers of normalization. CNN uses a kernel and slides it through the image to create the convolutional layer. (Amir Shahroudy et al., 2017)

1.2 Statement of the problem

the proposed system is set to eradicate the issue associated with the manual way, the chaplaincy unit check for students present with their Mountain Top University chapel books but by the introduction of Image Classification Model which will make the chaplaincy officials to stop one-by-one checking of chapel books or manual, which will take a lot of time especially in a situation where Mountain Top University grows to the extent of having over 10, 000 students

1.3 Aim and Objectives

The aim of this study is to use methods of deep learning to develop a model that can accurately classify images. The objectives are to;

i. Collect the data to be used to train and test the model.

ii. Create several models and construct the layers of the models or use transfer learning.

iii. Train different models using the data collected

iv. Test the accuracy of the models to find the most accurate model.

1.4 Scope of the study

This research is limited to Mountain Top University as the model is made to classify the various books used by students in the student chapel.

1.5 Significance of the study

This research provides an application that serves as the framework for platform upon which applications can be built and operationalized for the classification of images worldwide. This model can be used to verify if students are with the valid chapel material depending on the service at hand.

1.6 Definition of Terms

Convolutional Neural network: Convolutional neural networks are a subset with profound neural networks that are most widely used for visual imagery analysis.

Deep Learning: Deep learning is part of a wider family of techniques for machine learning focused on artificial neural networks with teaching representation.

Artificial Neural network: An artificial neural network (ANN) is the piece of a computing system to simulate the human brain's way of analyzing and processing information.

Web Scraping: Web scraping is the scraping of data used for website data extraction.

Machine Learning: Machine learning is the study of algorithms in computers that are improved by experience automatically.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter provides a review of literature on classification of images. The presentation of this chapter begins with the conceptual review, theoretical review, review of related works.

In the fields of remote sensing, image processing and pattern recognition, image classification plays a significant role. The method to sort all the pixels of an image into a finite number of individual categories is digital image classification. (Rashmi & Mandar, 2011). Different advanced techniques such as Artificial Neural Networks in image classification (ANN). For image classification, Vector Machines (SVM), Fuzzy steps, Genetic Algorithm (GA), Fuzzy support Vector Machines (FSVM) and Genetics Algorithm with Neural Networks are being created. According to (Seetha & Muralikrishna, 2008) non-convex networks can accommodate artificial neural networks. decisions.

A minimal number of samples and computing to deal with units, conventional machine learning techniques (such as multilayer perceptron machines, vector support machines, etc.) mostly use shallow structures. When the target items have rich meanings, the effectiveness and generalization potential. Obviously, there are difficult classification issues. inadequate. In recent years, the field of image processing, the Convolutional Neural Network (CNN) has been commonly used because it is excellent at handling image classification and recognition problems. and has significantly enhanced the accuracy of many assignments for machine learning. It has become a solid and universal system model of deep learning. (Mingyuan & Yong, 2019).

2.2 Conceptual Review

2.2.1 Convolutional Neural Network

A multilayer neural network is the Convolutional Neural Network (CNN), and it is also the most traditional and general paradigm for deep learning. CNN is really interested in learning machines, and has outstanding hyperspectral image classification results. The representation based on CNN presents the spatial spectral contextual sensitivity that is critical for correct pixel classification by integrating a collection of different discriminant appearance variables. Newman et al (2019). The experimental results of the commonly used hyperspectral image datasets show that any other conventional deep-learning cl-based approach will outperform the proposed method. Neural Network Context-based Convolution algorithms are the most advanced methods of depth learning and classical non-neural network algorithms that are recognized by deep structure and multilayer perceptron (MLP) pixel-based with shallow structure. (Zhang et al., 2018). In a succinct and successful way, the two algorithms are paired with very distinct habits and a rule-based fusion of decisions. approach is used to classify very fine spatial resolution. (VSFR) remote sensing images.

Convolutional Neural Networks are a category of Neural Networks in areas such as identification of images and, which they have proven very successful. Besides in powering robotic vision and robots, CNNs have been effective in recognizing objects, faces and traffic signs. self-driving cars. (jcgonzalez, 2017).

A neural network is said to be convolutional if it basically has a kernel sliding over images as shown below.

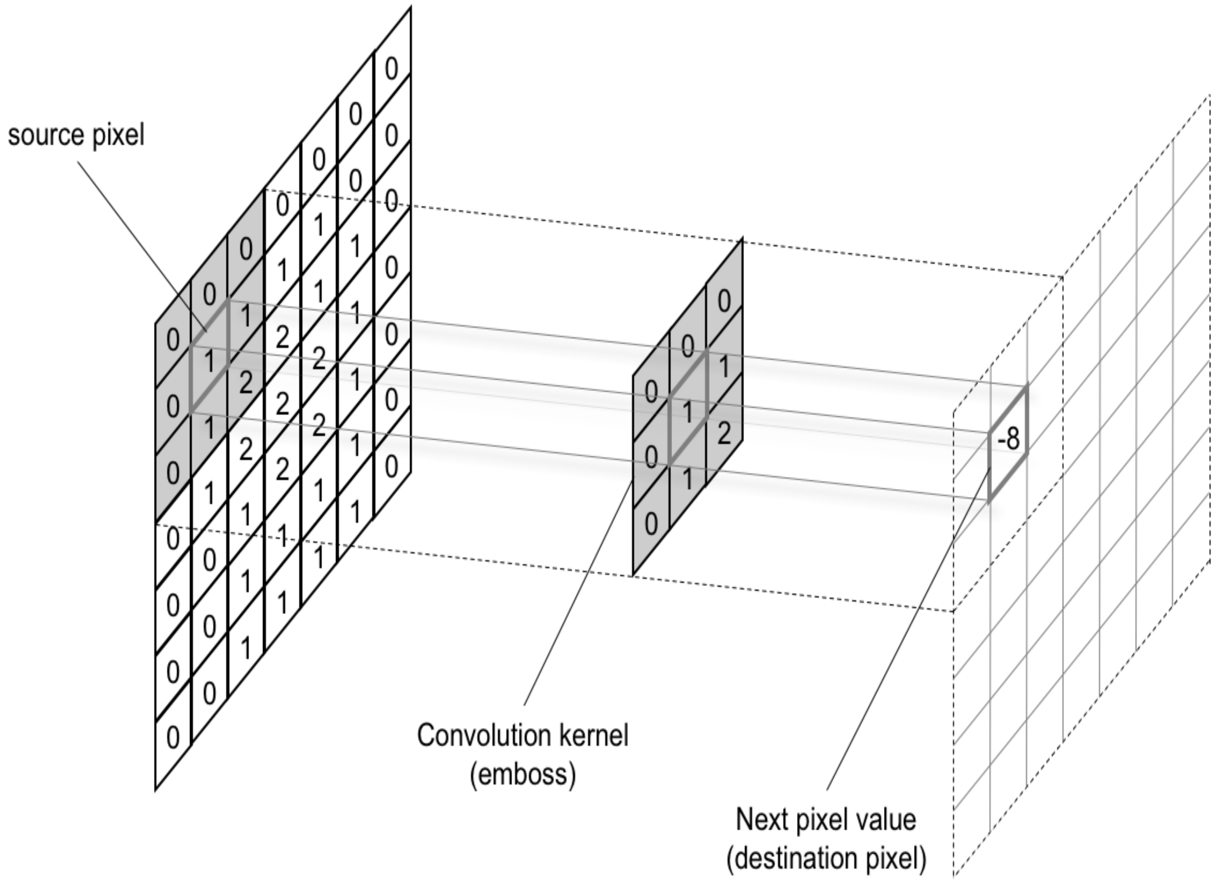


Figure 2.1: Convolutional Neural Network

A convolutional neural network can have different layers and different stages, but ideally, after a convolutional layer comes a reduction layer normally done using max pooling function. This function takes for example a 2*2 matrix and reduces it to just one number which is the maximum number in the matrix. After this, several layers can be added, some can be repeated. The output of CNNs is equal to the number of labels or classifiable objects. For example, if we are classifying objects of a cat and a dog, the output layer will have just two nodes. One for dogs, one for cats.

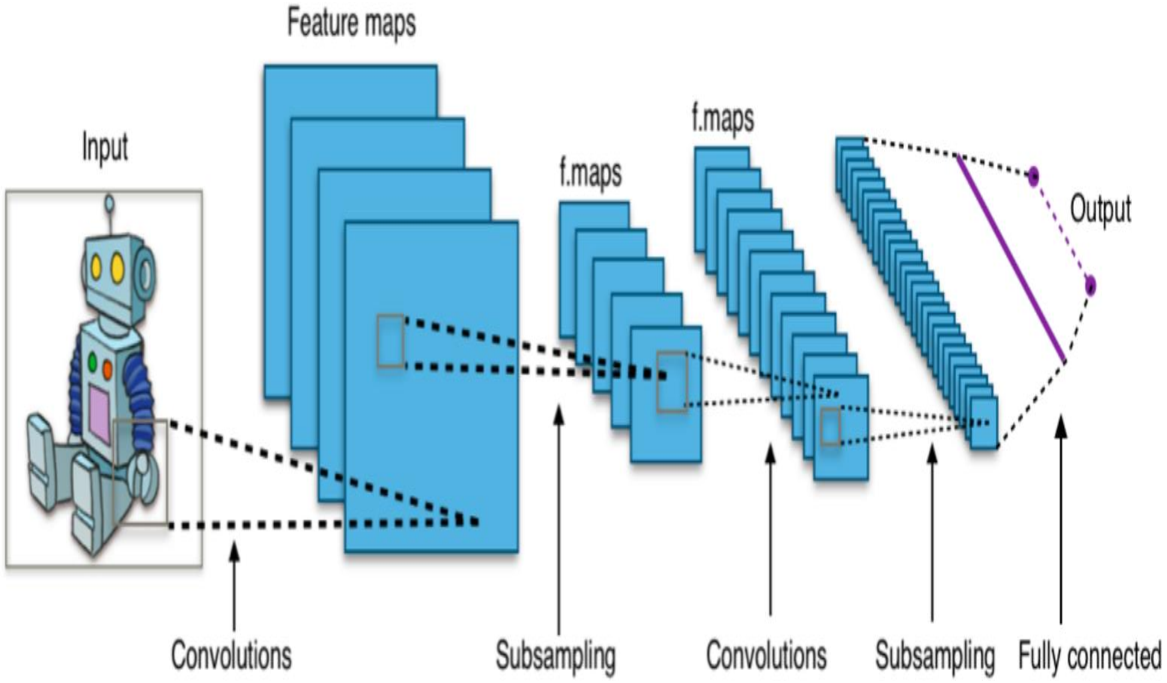


Figure 2.2: Convolutional Neural Network (different layers)

2.2.2 Machine Learning

Machine Learning (ML)is the data-based field in AI which provides systems the potential for learning automatically, and improve from practice without being programmed specifically. The computer is fed with lots of data which it learns from. It detects trends in the data and the usage can be made for prediction, clustering or classification. The process of learning first begins with observation of data. The primary objective of machine learning is to make it possible for the computer to automatically learn without the intervention or assistance of humans.

Some ML methods;

Supervised learning: This is the type of learning when the data used to train the model is labelled. Beginning from the analysis of a known training dataset, the inferred function is generated by the learning algorithm to make predictions about the output values. The system can provide goals for any fresh feedback after adequate training. The algorithm used can also equate its output with the right results to change the model accordingly and find errors.

Unsupervised Learning: In this case, the data used is not labelled. It studies how systems can deduce a function from which to define a hidden structure unlabelled data. It identifies patterns in the data and the usage can be made for cluster the attributes in the data into groups.

Reinforcement Learning: In this case, the agent meets with its environment through action development and discovers mistakes or rewards. For the agent to learn which behaviour is better, incentive feedback is needed; this is known as the reinforcement signal.

2.2.3 Web Scraping and Cleaning

Web scraping: Web scraping is a tool used to retrieve vast quantities of data from websites by extracting the data and saving it to a local disk in the computer or made it simpler for data scientists to have access to relevant data. This data collection technique is mostly used when text analysis is in mind to a database in table format. A lot of information is on the web and most data in the world are unclean. Working with an unclean data can be disastrous as it leads to wrong prediction and poor accuracy. When a data set has missing values, there are several steps one can take:

a. Remove the entire row
b. Remove the entire column
c. Replace the missing value(s)

Replacing the missing value is the most common practice in this kind of situation. The missing value is replaced with the most prevalent value in the column or with the average of all other values in the column.

2.2.4 Transfer Learning

This is a machine learning research issue that concentrates on storing the weights and prejudices, in other words, information acquired while solving one issue and the application of it to a separate but related problem. For example, awareness acquired when learning to recognize dollars, could apply when trying to recognize naira. This is a very effective way to build precise models in a time-saving way (Rawat & Wang 2017). With transfer learning, you start with trends that have been learned, rather than beginning the mechanism of learning from scratch. while resolving a separate problem. Previous learning is therefore leveraged and we stop starting from scratch.

In computer vision. Transfer learning is typically conveyed by the use of templates that are pre-trained. A pre-trained model is a prototype trained on a model broad benchmark dataset to solve a problem similar to the one we want to solve. (Marcelino, 2018).

2.3 Theoretical Review

The difficulty of classifying images from a large dataset has been the focus of recent research. The classifier of the support vector machine (SVM) proves to be quite effective in the categorization of images. (Wu et al., 2012).

Developed Artificial Neural Network which manages the noisy data efficiently, and this approach can represent AND, OR, and NOT. (Wang, 2014).

(Bianchini & Scarselli, 2014) Proposed a method on how the depth of feedforward neural networks that affects their ability to implement functions of high complexity.

In this research, a diagnosis for bone malignant growth using Kmeans segmentation and KNN classifier for bone disease recognition is suggested by (Subbaraya, 2019).

Kader et al. (2015) focused on the use of input image matrix to conduct training and testing on sample images acquiring the recognition accuracy establishing the network by adjusting the weights in a supervised manner.

Nagare (2015) proposed the number plate recognition using back propagation and learning vector quantization neural network and compared the results of both types. Extracted Characters are converted to features and are trained to check the performance.

Mostafa et al., (2009) proposed Vector Machines for Directed Acyclic Graph Support (DAG SVM). It suggests a multi-class weighted classification technique that divides the space of input into many subspaces. A DAG SVM is trained and its density function (pdf) is guessed for each subspace in the training process of the technique. The test step is determined when pdf of the subspace is used to match the value of each input pattern as the weight of each DAG SVM to each subspace. Finally, to determine the class mark of the given input pattern, a fusion operation is specified and applied to the DAG SVM outputs. The results of the evaluation indicate the prominence of our multi-class classification system compared with DAG SVM.

2.4 Reviews of Related work

A deep learning architecture based on the convolutional neural network (CNN) and the data fusion scheme of Naive Bayes (called NB-CNN) was proposed by Huang et al., (2019), which the usage can be made for the study of a single crack detection video frame. At the same time, to combine the information collected from each video frame to increase the overall efficiency and robustness of the system, a modern scheme for the fusion of data is suggested.

A survey on techniques and methods of image classification was carried out by (Lu & Wend, 2007). Image classification is a dynamic process that can be influenced by several variables. They discuss current image classification practices, challenges, and prospects. The focus is on summarizing the main approaches to advanced classification and the methods used to improve the accuracy of classification.

(Jipsa & Vkarunakaran, 2012) A survey on the method of image classification was conducted and Image classification was found to be one of the most significant complex fields of image processing. If it is blurred and noisy material is used, it is nuanced and hard to define. There are many methods for classifying, they include effective classification of images and results, but when the image includes fuzzy and noisy information, they fail to have adequate rankings results. Supervised and unsupervised classification are the two primary methods for image classification. Both classifications have their own benefits and drawbacks. The noisy and distorted image makes it impossible to produce a better result than the regular image.

Saurabh et al., (2013) increase the classification using support vector machine. One of the reasons for the elevated dimensionality of the space of a feature is conventional to classification approaches that deal poorly with content-based image classification tasks. Color image classification in this paper is performed on features derived from histograms of component colors. Better efficiency and insensitivity to minor shifts in camera perspectives i.e. translation and rotation, are the advantages of using color picture histograms. For the classification of the remotely sensed hyper spectral images, a restricted linear discriminate analysis (CLDA) method. Its basic concept is to plan and to design an efficient linear transformation operator that can optimize the inter-class to intra-class distance ratio while meeting the constraint of aligning the numerous centers of the class in different directions after transformation. Its key benefit over the conventional linear discrimination study of Fisher is that it is feasible to accomplish the classification concurrent with the transition. The CLDA, i.e. the class spectral signatures, is a supervised approach. (Qian, 2007).

Zhang et al., (2012) Automatic Image Classification Using the Ant-Colony Classification Algorithm The classification based on ant-colony is described in this paper in order to improve the usability, robustness, and convergence rate of automatic image classification. The standard Ant-Colony algorithm is implemented and improved by this model according to the functionality of the picture classification. It defines two kinds of ants with various search policies and refreshing processes. New categories are defined by stochastic ants; category tables are constructed and the clustering center for each segment is calculated. The analysis shows that the ant-colony algorithm increases performance and precision.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This section describes the approach model used to establish the model of image classification in a comprehensive manner. The methodology comprises of a series of methods/techniques that started with the identification and collection of data needed to develop the model.

3.2 Data Collection

Data collection will be done manually as there might not be possible way of scraping the images of Mountain Top University's chapel books from the internet. Several pictures of these books will be taken with 'noise'. Noise in the sense that the photographs are taken in multiple contexts. Pictures of the books will be taken with other objects to help increase the reliability of the template so that it can identify these books even if a part of it is showing or it is really showing. Similar pictures would be taken in different lightning backgrounds so that the model would be able to identify these objects even when there is poor lightning.

Images will be manually split into two directories, one for training and one for validation with the validation data being 20% of the whole data set. The validation images will be selected randomly.

3.3 Data Preprocessing

Some of the pictures that will be taken for the Image classification might be faulty like they might be dark, blurry, e.tc so it will have to be sorted out and deleted. In Data preprocessing the images will be converted to number because neural network basically works with digits. So, it will be in digits for neural network to understand.

Data preprocessing is a technique for data mining that requires converting the raw data into a comprehensible format. Real-world knowledge is always incomplete, unreliable, and/or missing, and is likely to contain several errors in some habits or patterns. Preprocessing of data is an established method of solving such problems. (Sharma, 2020)

3.3.1 Data Cleaning

Data Cleaning Data cleaning is the method of preparing data for review by deleting or altering inaccurate, incomplete, obsolete, duplicated or incorrectly formatted data. The data will be cleaned to avoid any form of error, cleaning the data can be in form of incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

3.3.2 Data augmentation

The amount of data will be increased by adding copies of already slightly changed ones existing data or newly created synthetic data from existing data. Even when I have lots of data, it will support and boost the sum of relevant data in my dataset. This is linked to the way neural networks communicate with one another to learn. Data augmentation will be used to take an image and change the shape so it looks like a new image but still the same image.

3.4 Creating the model for image classification

Two models will be created for this research purpose.

The first model will have its layers constructed by me, while the other would be a pre-trained model. ImageDataGenerator will be used to implement data augmentation for my training data and will shuffle generator. A model will be created in which the beginning layers have little nodes as compared to the middle or ending layers. In other words, there will be an increase in the amount of node numbers as the layers increase. Dropouts will be used in several areas to avoid overfitting. The activation of my convolution layers and dense layers would be ‘relu’ aside the final layer which would be softmax since I am classifying more than two objects.

A Stochastic Gradient Descent optimizer will be used with a really low learning rate, and the loss metric would be categorical crossentropy.

The model will be trained for 10 epochs and then the training accuracy will be plotted alongside the validation accuracy, and also plot the training loss alongside the validation loss.

A pre-trained model will be loaded, freeze its layers, add a final layer which would be used for classification, then train it with the data. The loss and optimizers will be same as that used for my custom-built model.

After this is done, the training accuracy will be plotted alongside the validation accuracy, and plot the training loss alongside the validation loss.

3.4.1 Convolutional neural network

With convolutional neural networks, under the limitations of a particular problems of predictive modeling, such as picture recognition, large numbers of filters specific to a dataset of training in parallel will be needed. The effect is extremely unique characteristics that can be observed on input images anywhere. A subset of deep neural networks, most widely used for visual imagery processing, is the Convolutional neural network. A Convolutional Neural Network (ConvNet/CNN) is a deep learning algorithm that can take an input image, apply significance (learnable weights and biases) to different aspects/objects in the picture and the ability to differentiate between them. Compared to other classification algorithms, the pre-processing needed within a ConvNet is much smaller. Although filters are hand-engineered in primitive processes, with sufficient preparation, ConvNets have the ability to learn these filters/traits (Sumit, 2018).

With their special mechanism, Convolutional Neural Networks perform better than other Deep Neural Network architectures. CNNs group multiple pixels together, one pixel at a time instead of looking at the background. (an example of 3 to 3 pixels as in the image above so that they can recognize a temporal pattern.

In another way, CNNs can see a group of pixels that form a line or curve. Due to the deep Neural Networks architecture, not a pixel party, but groups of lines and curves that form certain shapes can be seen in the next step. And so on until they create a complete picture.

3.4.2 Learning rate

With learning rate, the model will be changed in response to the estimated error each time the model weights are updated. It can be difficult to select a learning rate since a value that is too small can lead to a long training period that can get stuck, whereas a value that is too large can lead to too quickly learning a sub-optimal set of weights or an inconsistent training system where the learning rate is a tuning parameter in machine learning and statistics in an optimization algorithm that specifies the step size at each iteration when progressing towards a minimum of loss function.

Learning rate is a hyper-parameter that regulates the weight of our neural network with regard to the gradient of loss. It explains how easily the neural network updates the ideas it has learned. A suitable level of learning is low enough for the network to converge into something useful, but big enough for it to be able to be educated within a reasonable time. An empirical approach cannot measure the weights of a neural network. Instead, via an empirical optimization method called stochastic gradient descent, the weights must be discovered.

Stochastic gradient descent is discussed by difficult and the solution space for neural networks, the problem of optimization (weight sets) can consist of several nice and easy to find solutions (called global optima) but low in ability solutions (called local optima). During each stage of this search process, or the phase size, the sum of the model modification is called the "learning rate" and delivers probably the most significant hyperparameter tuning into your neural network to achieve good results on your problem. (Jason, 2019)

3.5 Training the Model

In training the model, firstly the model will be compiled. The goal of training a model is to find a range of low-loss weights, on average. The ML training model method includes providing the ML algorithm (i.e. the learning algorithm) with training data from which to learn. The training data will contain the correct response known as an attribute of the target or target. In the training data, the learning algorithm will then find patterns that map the attributes of the input data to the target (the response you want to predict), and it generates an ML model that captures these patterns.

3.5.1 Transfer Learning

Transfer learning is a form of machine learning where a model designed for a task, a blueprint on a second task is reused as the starting point.

It is a common deep learning method in which it uses pre-trained models as the starting point for the processing of computer vision and natural language tasks, given the vast computational and time resources needed to build neural network models on these issues and the enormous skill leaps they provide on related issues. (Jason, 2017).

With transfer learning the trained time will be decreased for a neural network model and make its result in lower generalization error.

3.6 Test the Model

The performance of the test model is about checking the models with the test data/new data sets and comparing the model's efficiency with parameters such as accuracy/recall. etc., with the model already built and moved into production, to that of pre-determined precision. This is the most trivial of different techniques which could be used for blackbox testing (Kumar, 2018) The test data will be used to check the model's accuracy and display the result.

CHAPTER FOUR

IMPLEMENTATION AND RESULT

4.1 Introduction

This chapter shows the information of implementing image classification system. For the purpose of implementation data was uploaded in the environment, the data was trained so the system can see it and so you can call the data anytime in the environment. The system was developed using python, deep learning, tensorflow, numpy, matplotlib e.t.c.

The Anaconda Navigator was used to access the Jupyter Notebook where the codes were written.

4.2 Software and Hardware Requirements

The recommended requirements for the Designed system are shown below:

Processor:	Intel® Celeron (Minimum)
Processor speed:	2.5GHz (Minimum)
RAM:	2GB (Minimum)
Hard disk:	319GB (Minimum)
Monitor Display:	LED
Mouse:	Touchpad with multi-touch gesture support, USB or PS/2

4.3 Installation Processes (Needed Packages)

Step 1: Go to Anaconda Navigator

Step 2: Go to the terminal located in Anaconda Navigator

Step 3: Change directory (if necessary)

Step 4: pip install tensorflow

Same applies to all the packages you will need (pip install) then, the package name next to install.

Detailed

After all the requirements has been met, launch the Anaconda Navigator, open the terminal (click on environment by the top left, then click on base(root)) then you see the terminal and run the following (pip install tensorflow) command in order to install all the software requirements.

4.4 Design Architecture of Image Classification

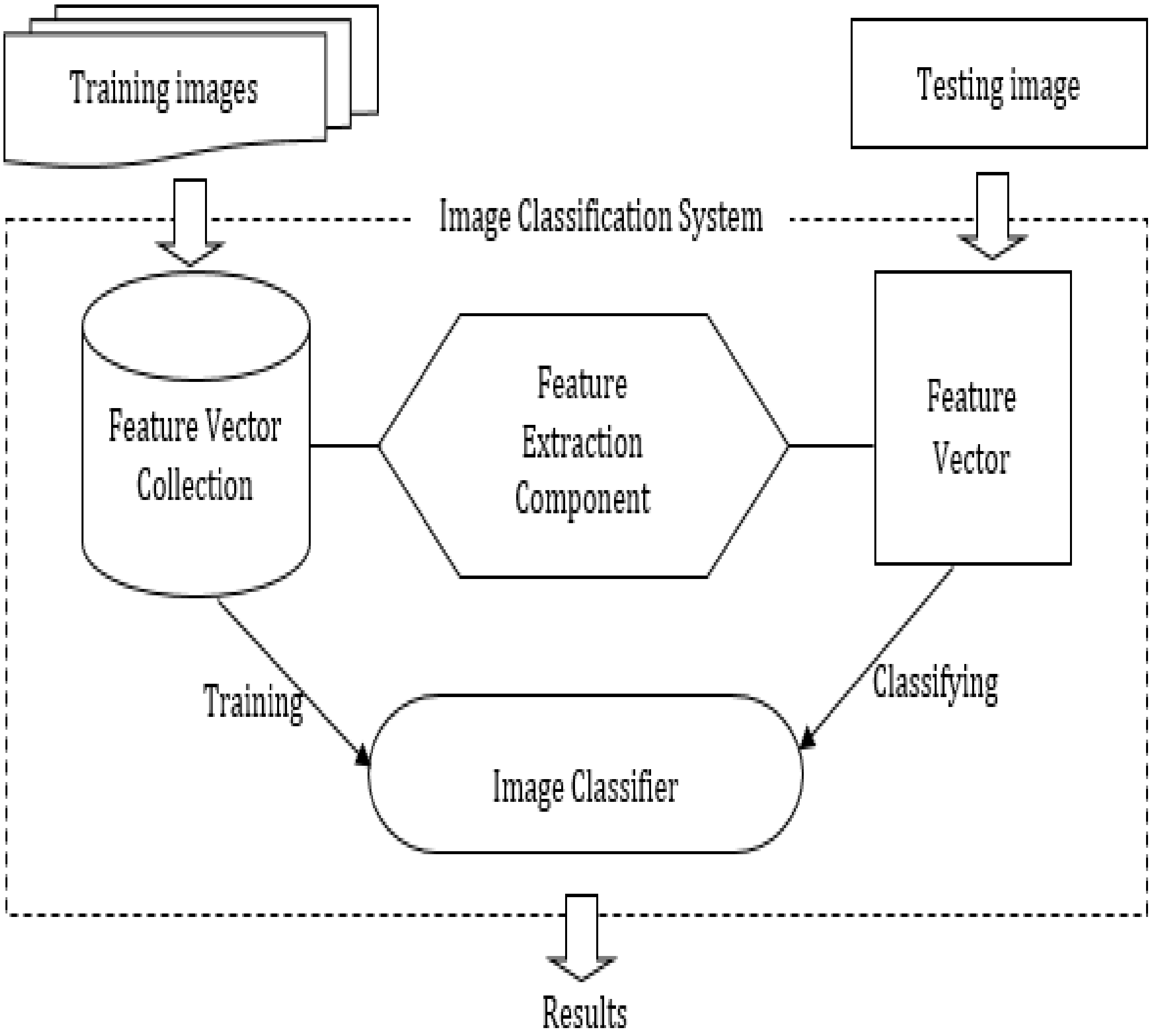
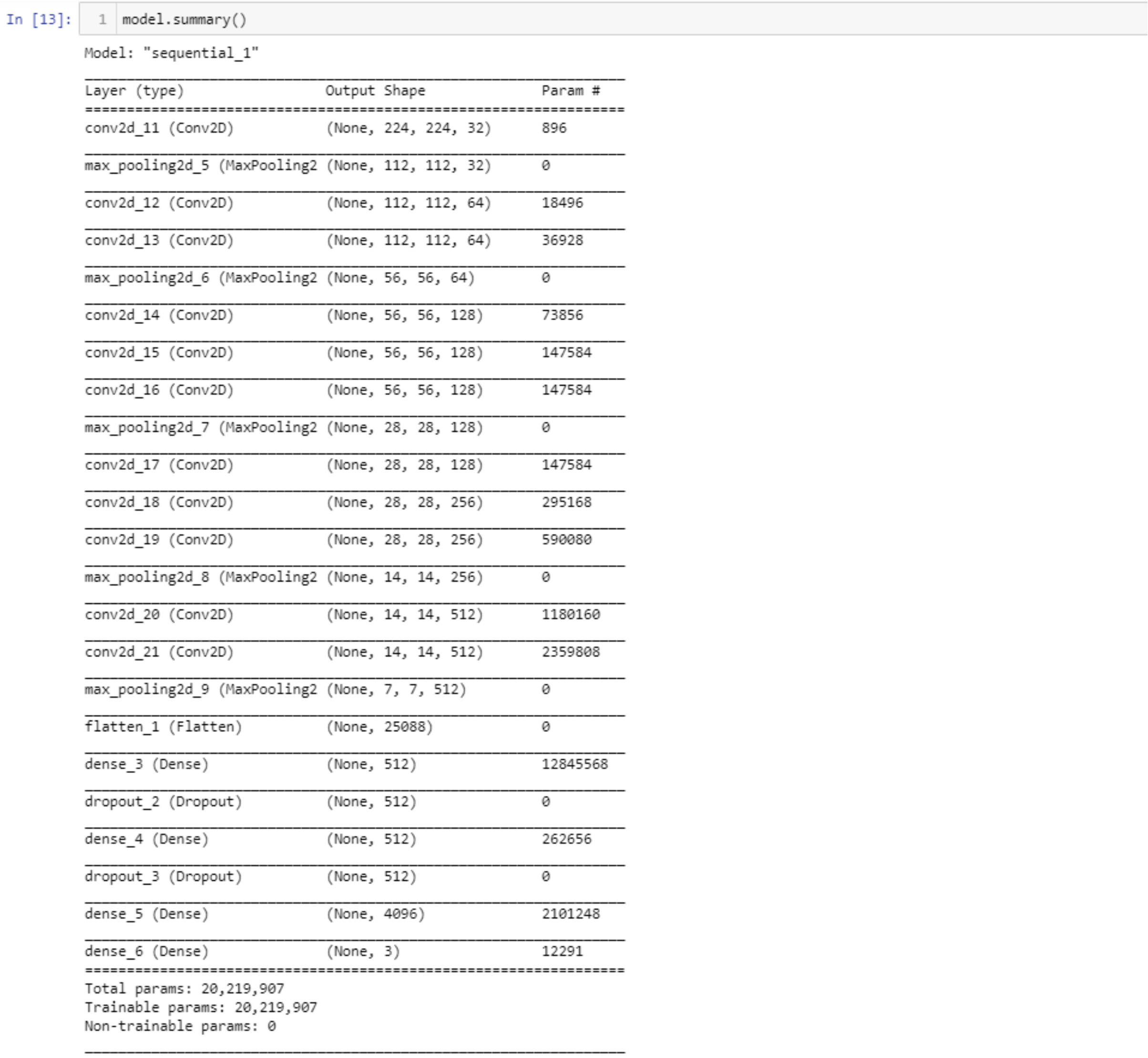


Figure 4.1: Design Architecture of Image Classification

4.5 Results of Custom Model

The custom model had 11 convolutional layers and 23 layers in total including the classification layer. Below is a diagram of the model's summary



```
In [13]:  model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_11 (Conv2D)           (None, 224, 224, 32)      896       
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 112, 112, 32)      0         
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 112, 112, 64)      18496     
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 112, 112, 64)      36928     
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 56, 56, 64)        0         
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 56, 56, 128)       73856     
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 56, 56, 128)       147584    
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 56, 56, 128)       147584    
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 28, 28, 128)       0         
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 28, 28, 128)       147584    
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 28, 28, 256)       295168    
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 28, 28, 256)       590080    
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 14, 14, 256)       0         
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 14, 14, 512)       1180160   
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 14, 14, 512)       2359808   
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 7, 7, 512)         0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0         
_________________________________________________________________
dense_3 (Dense)              (None, 512)               12845568  
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0         
_________________________________________________________________
dense_4 (Dense)              (None, 512)               262656    
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0         
_________________________________________________________________
dense_5 (Dense)              (None, 4096)              2101248   
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 12291     
=================================================================
Total params: 20,219,907
Trainable params: 20,219,907
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.2: Custom Model with 11 convolutional layers and 23 layers

The model was trained for 10 epochs and it had a training accuracy of 0.65 and a validation accuracy of 0.66. Below is the plot of the training accuracy alongside the validation accuracy and the training loss alongside the validation loss.

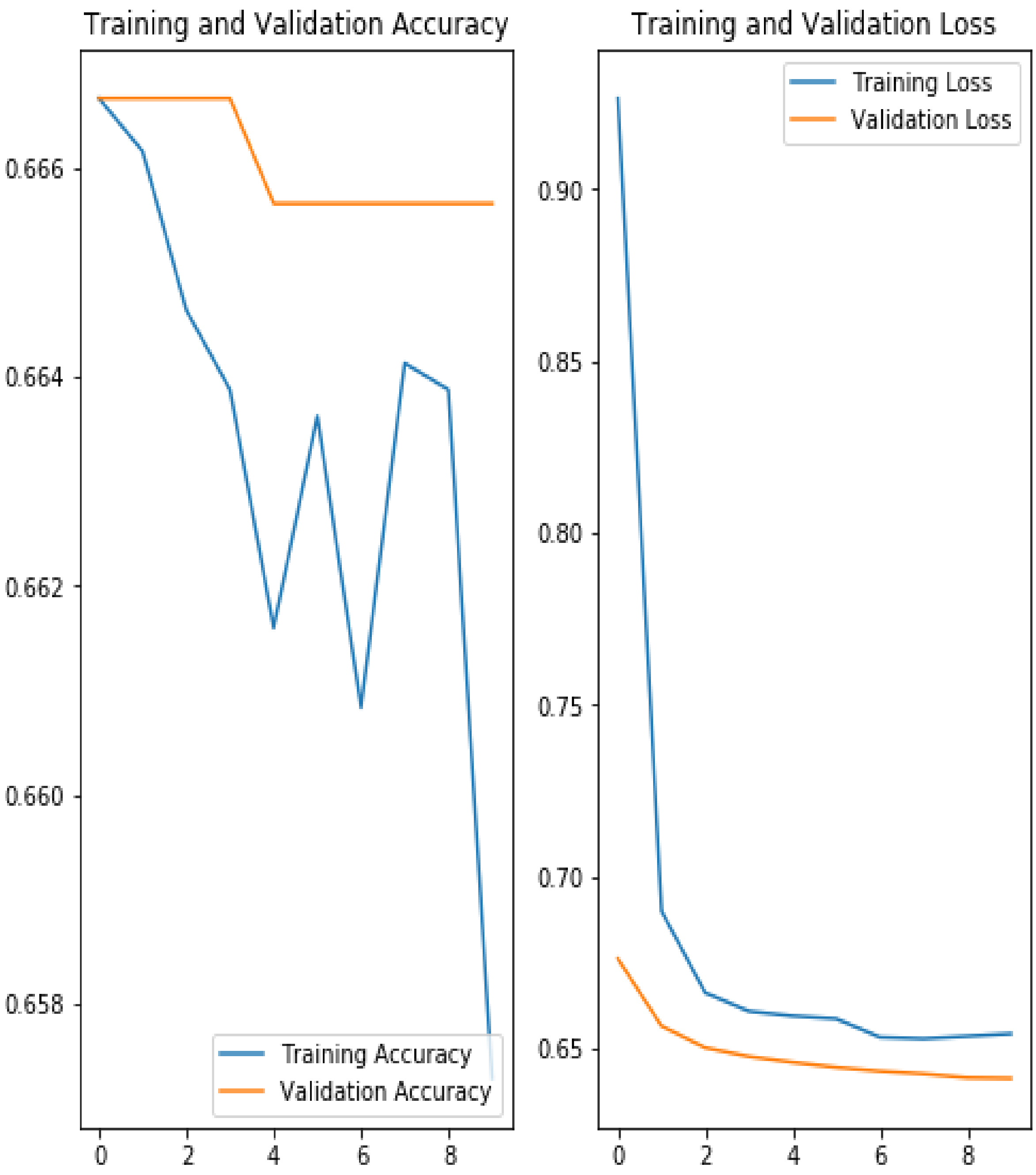


Figure 4.3: Training and Validation accuracy & loss

4.6 Mobile Net V2

MobileNetV2 is a pre-trained model which is part of TensorFlow-Slim image classification library. I loaded the model and trained it for 100 epochs. It got a training accuracy of 79% and a validation accuracy of 76% as shown below. The compilation parameters were the same as used for the custom model built.

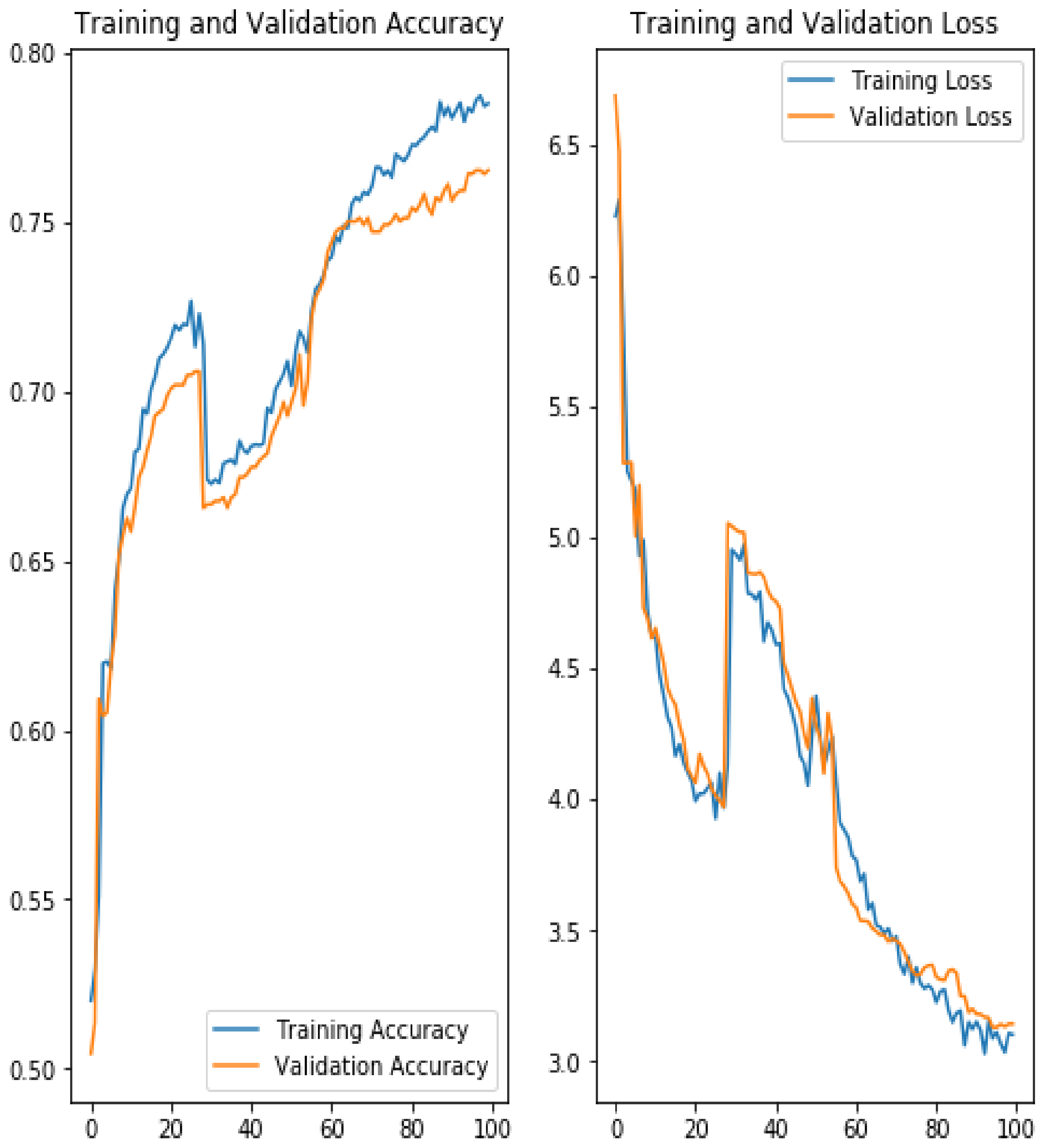


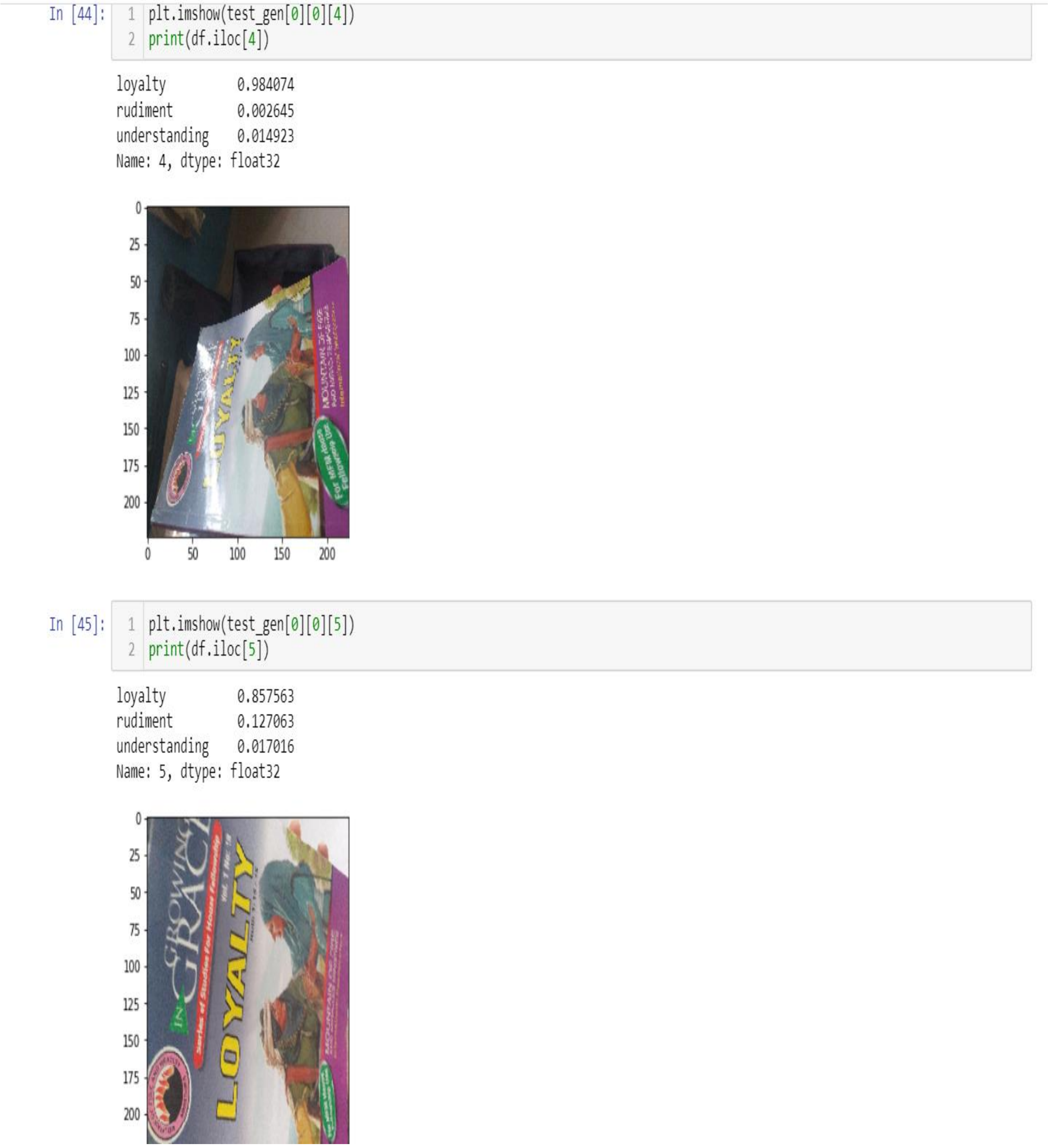
Figure 4.4: Training and Validation accuracy & loss

4.7 Testing of The Pre-Trained Model

I took random pictures of the three books and fed it to the model to predict them. The accuracy was pretty high and a screenshot of some of its predictions are shown below.

For loyalty book;

As shown above, for the first image the model was 98% sure it is the loyalty book and 86% sure for the second image.



In [44]:
```
plt.imshow(test_gen[0][0][4])
print(df.iloc[4])
```

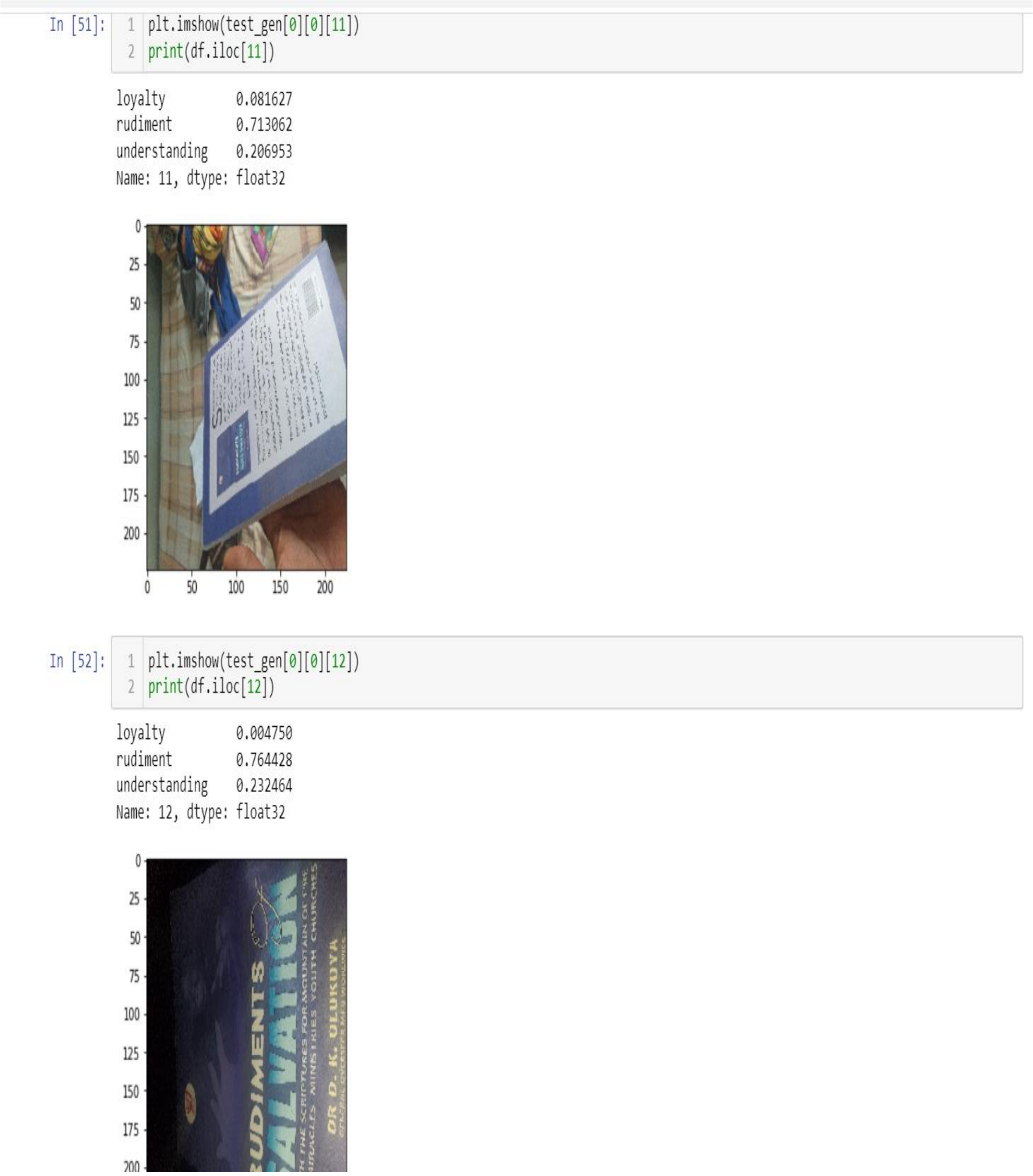
```
loyalty          0.984074
rudiment         0.002645
understanding    0.014923
Name: 4, dtype: float32
```

In [45]:
```
plt.imshow(test_gen[0][0][5])
print(df.iloc[5])
```

```
loyalty          0.857563
rudiment         0.127063
understanding    0.017016
Name: 5, dtype: float32
```

Figure 4.5: Picture of the first chapel book

For Rudiments of salvation.



```
In [51]: plt.imshow(test_gen[0][0][11])
         print(df.iloc[11])
```

```
loyalty          0.081627
rudiment         0.713062
understanding    0.206953
Name: 11, dtype: float32
```

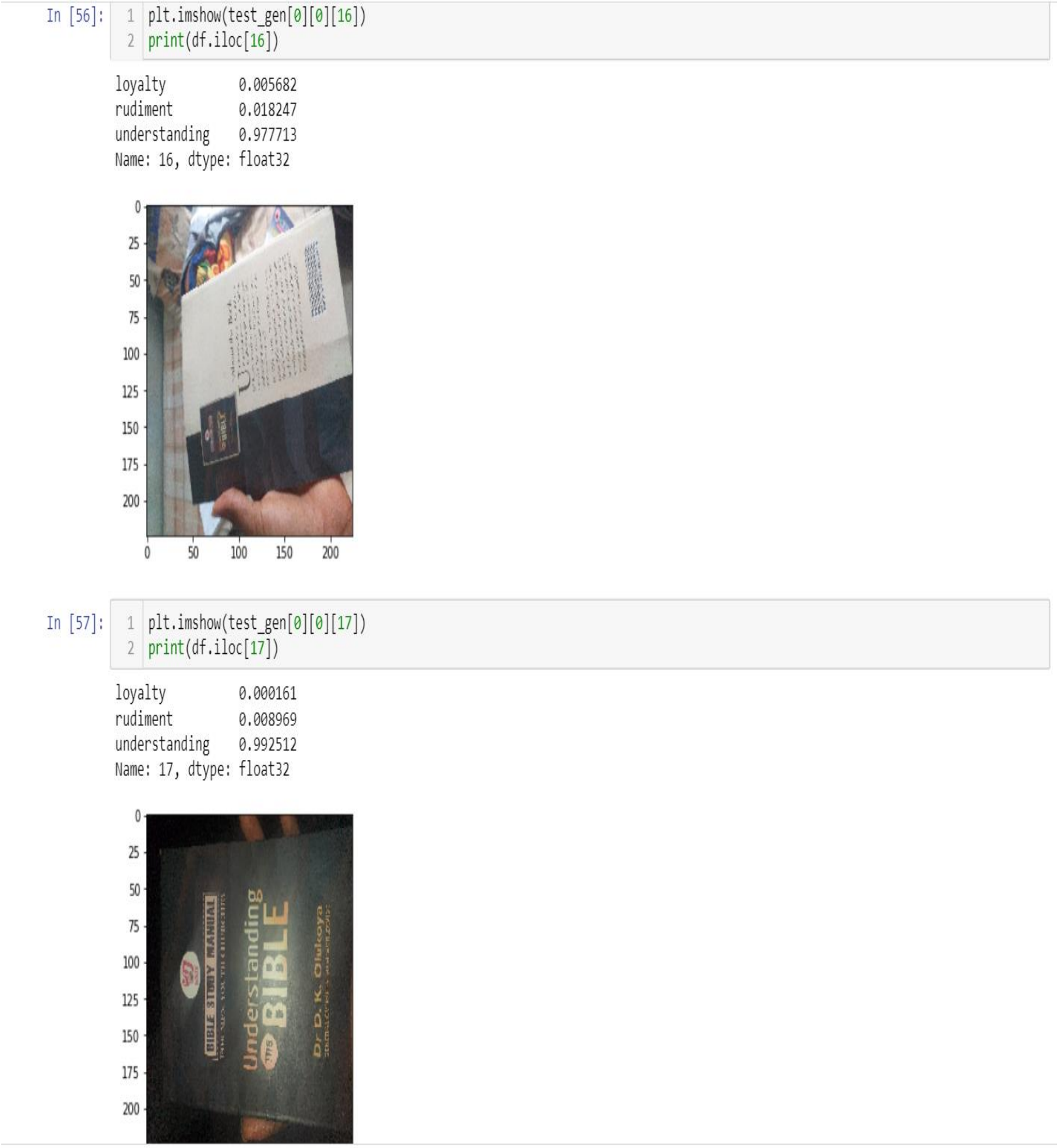
```
In [52]: plt.imshow(test_gen[0][0][12])
         print(df.iloc[12])
```

```
loyalty          0.004750
rudiment         0.764428
understanding    0.232464
Name: 12, dtype: float32
```

Figure 4.6: Picture of the second chapel book

As shown above, the model was 71% sure the first image is the rudiment book and 76% sure of the second image.

For Understanding the Bible.



```
In [56]: plt.imshow(test_gen[0][0][16])
         print(df.iloc[16])
```

```
loyalty          0.005682
rudiment         0.018247
understanding    0.977713
Name: 16, dtype: float32
```

```
In [57]: plt.imshow(test_gen[0][0][17])
         print(df.iloc[17])
```

```
loyalty          0.000161
rudiment         0.008969
understanding    0.992512
Name: 17, dtype: float32
```

Figure 4.7: Picture of the Third chapel book

As shown above, the model is 98% sure the first image is understanding the Bible, and 99% sure of the second image.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

Having seen the challenges encountered at the Mountain Top University chapel when checking for those present with their chapel books, the proposed system is set to eradicate the issue associated with the manual way, the chaplaincy unit check for students present with their Mountain Top University chapel books but by the introduction of Image Classification Model which will make the chaplaincy officials to stop one-by-one checking of chapel books or manual, which will take a lot of time especially in a situation where Mountain Top University grows to the extent of having over 10, 000 students. This will be so much hard checking students with or without chapel books but, with the use of this Image Classification Model, the chaplain or any of the chaplaincy officials will implement the model in a device then, from their offices it would detect those with or without the current chapel books.

5.1 Contribution to Knowledge

The key contribution is the knowledge was the ability to develop a model to help avoid stress at the Mountain Top University chapel using Artificial Neural network. The codes were effectively functioning and ready for use with the aid and use of the Deep Learning technique. Even, when a new chapel books the same process can be done to the new books and it will as well show automatically whether the students came to the chapel along with the book or not.

5.2 Limitations

i. Inability for carrying out the full Implementation in an application due to insufficient funds for the deployment of the model

ii. It can’t be used on tablets, mobile devices or om PC.

5.3 Recommendation for future study

It is recommended that researchers should study techniques in deep learning to show the possibility of automatically detecting objects such as stop signs and lights for traffic.

5.4 Conclusion

This project shows problems of checking each student one-by-one, making sure they all come with their chapel manual which can result to time wastage. With using Image Classification artificial neural network, instead of humans doing it, a system will be in chapel to see any of the students and know if he/she has the chapel book needed. In conclusion, artificial neural network (ANN) is when the computer system is aimed at simulating, the manner in analyzing and processing the human brain information. It is the foundation of **artificial** intelligence (AI) and solves problems that would prove impossible or difficult by human or statistical standards.

REFERENCES

1. E. Newman, M. Kilmer, L. Horesh, Image classification using singular value decompositions of local tensor images (IEEE, international workshop on Multi-Sensor Adaptive Computing Computational Advances. IEEE, Willemstad, 2018), pp.1-5.
2. X. Wang, C. Cheng, et al, Zero-shot classification of images based on extraction of deep features United Kingdom: IEEE Transactions on Cognitive & Developmental Systems, 2018.
3. A.A.M. Al-Saffar, H. Tao, M.A. Talab, Review of deep convolutional neural network in image classification (International conference on radar, antenna, microwave, electronics, and telecommunications. IEEE, Jakarta, 2018), pp.26-31.
4. 1. Bengio, Y., 2009. Learning deep architectures for AI. Foundations and trends in Machine Learning, 2(1), pp.1–127.
5. Canziani, A., Paszke, A. and Culurciello, E., 2016. An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678.
6. Chollet, F., 2015. Keras.
7. Chollet, F., 2017. Deep learning with python. Manning Publications Co..
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A database of large-scale hierarchical images. In Computer Vision and identification of patterns, 2009. CVPR 2009. IEEE Conference on (pp. 248–255). Ieee.

9. He, K., Zhang, X., Ren, S. and Sun, J., 2016. For image recognition, deep residual learning. In IEEE conference proceedings on machine vision and pattern recognition (pp. 770–778).

10. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Deep convolutional neural networks Imagenet classification. In Advances in applications of neuronal knowledge processing (pp. 1097–1105).

11. LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.

12. Lin, M., Chen, Q. and Yan, S., 2013. Network in network. arXiv preprint arXiv:1312.4400.

13. Pan, S.J. and Yang, Q., 2010. A Transfer Learning survey. IEEE Information and Data Engineering Transactions, 22(10), pp.1345–1359.

14. Rawat, W. and Wang, Z., 2017. Deep convolutional neural networks for classifying images: A comprehensive review. Neural computation, 29(9), pp.2352–2449.

15. Simonyan, K. and Zisserman, A., 2014. Quite deep convolutional large-scale image recognition networks arXiv preprint arXiv:1409.1556.

16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the architecture of initiation for computer vision. In Proceedings on computer vision and pattern recognition at the IEEE conference (pp. 2818–2826).

17. Tang, Y., 2013. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.

18. Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E., 2018. For computer vision, deep learning: A brief review. Computational intelligence and neuroscience, 2018.

19. Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. Why features in deep neural networks are transferable? In Advances in applications of neuronal knowledge processing (pp. 3320–3328).

20. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing convolutional networks and understanding them. Conference on Computer Vision in Europe (pp. 818–833). Springer, Cham.

APPPENDIX

SOURCE CODE

```
from __future__ import division, absolute_import, print_function, unicode_literals

import tensorflow as tf

import tensorflow_hub as hub

import os

import glob

import shutil

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import SGD

import numpy as np

import matplotlib.pyplot as plt

train_dir=r"D:\Downloads\deep learning projects\book classification\training"

val_dir=r"D:\Downloads\deep learning projects\book classification\val"

batch_size=100

img=224

IMG=224

image_gen=ImageDataGenerator(rescale=1./255, horizontal_flip=True, zoom_range=0.5,
rotation_range=0.5,

                    width_shift_range=0.2, height_shift_range=0.2)
```

```
train_data_gen=image_gen.flow_from_directory(shuffle=True, target_size=(img, img),
batch_size=batch_size,
                                               directory=train_dir)
image_gen=ImageDataGenerator(rescale=1./255)


val_data_gen=image_gen.flow_from_directory(target_size=(img,img), directory=val_dir,
batch_size=batch_size,
                                             shuffle=False, class_mode='categorical')
model=tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, 3,input_shape=(img, img, 3), padding='same'),
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'),
    tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu'),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu'),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu'),
    tf.keras.layers.Conv2D(256,3, padding='same', activation='relu'),
    tf.keras.layers.Conv2D(256,3, padding='same', activation='relu'),

    tf.keras.layers.MaxPooling2D((2,2)),

    tf.keras.layers.Conv2D(512,3, padding='same', activation='relu'),

    tf.keras.layers.Conv2D(512,3, padding='same', activation='relu'),

    tf.keras.layers.MaxPooling2D((2,2)),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(512,activation='relu'),

    tf.keras.layers.Dropout(0.3),

    tf.keras.layers.Dense(512, activation='relu'),

    tf.keras.layers.Dropout(0.3),

    tf.keras.layers.Dense(4096, activation='relu'),

    tf.keras.layers.Dense(3, activation='softmax')

])

model.summary()

opt = SGD(lr=0.0001)

model.compile(loss='categorical_crossentropy',

              optimizer=opt, metrics=['accuracy'])

EPOCHS = 10



history = model.fit(train_data_gen,
                    epochs=EPOCHS,

                    validation_data=val_data_gen)

acc = history.history['accuracy']

val_acc = history.history['val_accuracy']



loss = history.history['loss']

val_loss = history.history['val_loss']



epochs_range = range(EPOCHS)



plt.figure(figsize=(8, 8))

plt.subplot(1, 2, 1)

plt.plot(epochs_range, acc, label='Training Accuracy')

plt.plot(epochs_range, val_acc, label='Validation Accuracy')

plt.legend(loc='lower right')

plt.title('Training and Validation Accuracy')



plt.subplot(1, 2, 2)

plt.plot(epochs_range, loss, label='Training Loss')

plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')

plt.title('Training and Validation Loss')

plt.show()

image_gen_train = ImageDataGenerator(rescale=1./255, horizontal_flip=True,
zoom_range=0.45, rotation_range=0.45,

                        width_shift_range=0.15, height_shift_range=0.15)



train_data_gen=image_gen_train.flow_from_directory(shuffle=True, target_size=(IMG,
IMG), directory=train_dir,

                                  batch_size=batch_size)

mage_gen_train = ImageDataGenerator(rescale=1./255)



val_data_gen=image_gen_train.flow_from_directory(target_size=(IMG, IMG),
directory=val_dir,

                                 batch_size=batch_size, class_mode='categorical')

url=r"https://tfhub.dev/google/tf2-preview/mobilenet_v2/classification/2"

feature_extractor=hub.KerasLayer(url, input_shape=(IMG,IMG,3))

model=tf.keras.Sequential([

    feature_extractor,

    tf.keras.layers.Dense(3,activation='softmax')
])
#model.summary
model.compile(
  optimizer=SGD(lr=0.01),
  loss='categorical_crossentropy',
  metrics=['accuracy'])


EPOCHS = 1
history = model.fit(train_data_gen,
                    epochs=EPOCHS,
                    validation_data=val_data_gen)
import pandas as pd
test_dir=r"C:\Users\holar\Downloads\book classification\test"
image_gen=ImageDataGenerator(rescale=1./255)
test_gen=image_gen.flow_from_directory(directory=test_dir, target_size=(224,224),
shuffle=False)
predict=model.predict(test_gen)
predict=predict/predict.max()
names=['loyalty','rudiment','understanding']
df=pd.DataFrame(predict,columns=[names])
```

```
print(df)

plt.imshow(test_gen[0][0][0])

print(df.iloc[0])

plt.imshow(test_gen[0][0][1])

print(df.iloc[1])

plt.imshow(test_gen[0][0][2])

print(df.iloc[2])

plt.imshow(test_gen[0][0][3])

print(df.iloc[3])

plt.imshow(test_gen[0][0][4])

print(df.iloc[4])

plt.imshow(test_gen[0][0][5])

print(df.iloc[5]

plt.imshow(test_gen[0][0][6])

print(df.iloc[6])

plt.imshow(test_gen[0][0][7])

print(df.iloc[7])

plt.imshow(test_gen[0][0][8])

print(df.iloc[8])

plt.imshow(test_gen[0][0][9])
```

```
print(df.iloc[9])

plt.imshow(test_gen[0][0][10])

print(df.iloc[10])

plt.imshow(test_gen[0][0][11])

print(df.iloc[11])

plt.imshow(test_gen[0][0][12])

print(df.iloc[12])

plt.imshow(test_gen[0][0][13])

print(df.iloc[13])

plt.imshow(test_gen[0][0][14])

print(df.iloc[14])

plt.imshow(test_gen[0][0][15])

print(df.iloc[15])

plt.imshow(test_gen[0][0][16])

print(df.iloc[16])

plt.imshow(test_gen[0][0][17])

print(df.iloc[17])

plt.imshow(test_gen[0][0][18])

print(df.iloc[18])

acc = history.history['accuracy']
```

```
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(EPOCHS)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```