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 FUFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE MOUNTAIN TOP UNIVERSITY, IBAFO, OGUN STATE, NIGERIA

CERTIFICATION

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 fulfilment of the requirements for the degree of BACHELOR OF SCIENCE

(Computer Science)

_____________________________ **(Signature and Date)**

Prof. A. I. AKINWANDE

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 MTU 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 MTU 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 system designed to simulate the way the human brain analyzes and processes information. It is the foundation of artificial intelligence (AI) and solves problems that by human or statistical standards, would prove impossible or difficult.

CHAPTER ONE

INTRODUCTION

1.1 Background to study

Deep leaning also known as an artificial intelligence function that imitates the functions of the human brain in processing data and generating patterns for use in decision making is also known as the deep neural network. Deep learning is a subset of artificial intelligence machine learning that 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 of a neural network with a supervised leaning algorithm such as BP means finding the weights of the links connecting the nodes using a set of training examples. An error function in the form of the number of the squares belonging to the errors between the actual output from the training set and the computed outputs is minimized iteratively. The learning or training rule specifies how the weights are modified in each iteration.

In computer vision, image classification refers to a method that can identify an image according to its visual content. An image classification algorithm, for example, could be designed to tell whether or not an image contains a human figure. While for humans, detecting an object is trivial, a robust picture classification is still a great challenge in computer vision applications. 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 type of hidden layers it consists of. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers, and normalization layers. 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 MTU 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 MTU grows to the extent of having over 10, 000 students

1.3 Aim and Objectives

The aim of this study is to use deep learning techniques 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 a framework that serves as the 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 of deep 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 representation learning.

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 data scraping used for extracting data from websites.

Machine Learning: Machine learning is the study of computer algorithms that improve automatically through experience.

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 an important role. The method of sorting all the pixels in an image into a finite number of individual groups is digital image classification. (Rashmi & Mandar, 2011). Different advanced techniques such as Artificial Neural Networks in image classification (ANN). Support Vector Machines (SVM), Fuzzy measures, Genetic Algorithm (GA), Fuzzy support Vector Machines (FSVM) and Genetics Algorithm with Neural Networks are being developed for image classification. According to (Seetha & Muralikrishna, 2008) artificial neural networks can handle non-convex decisions.

Traditional machine learning methods (such as multilayer perceptron machines, support vector machines, etc.) to deal with a small number of samples and computing units, they often use shallow structures. The efficiency and generalization ability of complex classification problems are obviously inadequate when the target items have rich definitions. The field of image processing, the convolutional neural network (CNN) created in recent years has been widely used because it is good at dealing with problems of image classification and recognition and has greatly improved the accuracy of many machine learning tasks. It has become a deep learning model that is strong and universal. (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 very interested in machine learning 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. Context-based convolution neural network algorithms that represent the most advanced depth learning methods and classical non-neural network algorithms are recognized with deep structure and pixel-based multilayer perceptron (MLP) with shallow structure (Zhang et al., 2018). In a succinct and successful way the two algorithms with very different behaviors are combined and a rule-based decision fusion 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 image recognition and classification, which they have proven very successful. Besides powering vision in robots and robots, CNNs have been effective in recognizing faces, objects 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.

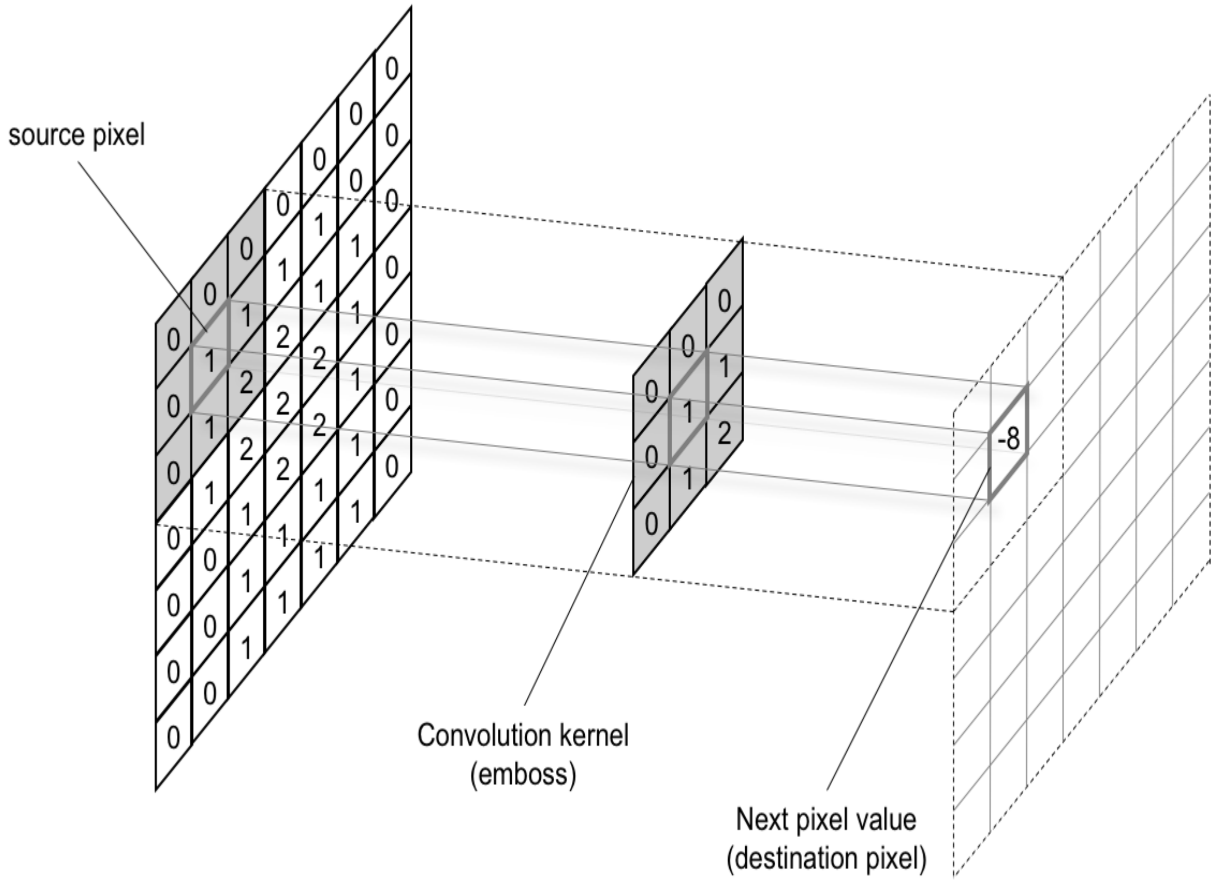


Figure 2.1: Convolutional Neural Network

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.

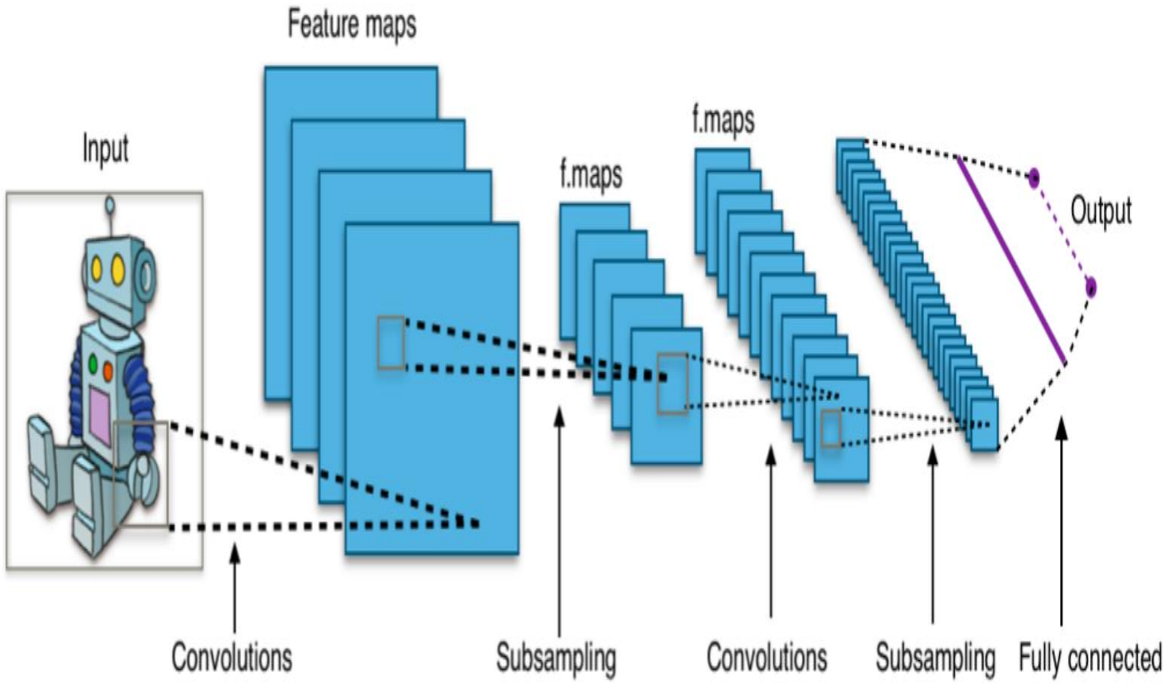


Figure 2.2: Convolutional Neural Network (different layers)

2.2.2 MACHINE LEARNING

Machine Learning (ML)is the data-based field in AI which provides systems the ability to automatically learn and improve from practice without being programmed specifically. The computer is fed with lots of data which it learns from. It finds patterns in the data and can be used for prediction, clustering or classification. The process of learning first begins with observation of data. The primary aim of machine learning is to allow the computer to learn automatically 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 targets for any new input after adequate training. The algorithm used can also equate its output with the right 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 to describe a hidden structure from unlabelled data. It identifies patterns in the data and can be used to cluster the attributes in the data into groups.

Reinforcement Learning: In this case, the agent interacts with its environment by producing actions and discovers errors 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 technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in the computer or made it easier 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 web scraping has

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 frequent 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 problem that focuses on storing the weights and prejudices, in other words, information acquired while solving one issue and applying it to a separate but related problem. For example, knowledge gained while learning to recognize dollars, could apply when trying to recognize naira. This is a very efficient way to build accurate models in a timesaving way (Rawat & Wang 2017). With transfer learning, you start from patterns that have been learned while solving a different problem instead of beginning the learning process from scratch. Previous learning is therefore leveraged and we stop starting from scratch.

In computer vision. Transfer learning is typically conveyed by the use of pre-trained models. A pre-trained model is a model trained to solve a problem similar to the one we want to solve on a large benchmark dataset. (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 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 Directed Acyclic Graph Support Vector Machines (DAG SVM). It suggests a weighted multi-class classification technique which divides the input space into several subspaces. A DAG SVM is trained and its probability density function (pdf) is guessed for each subspace in the training process of the technique. The test step is determined using the pdf of the subspace to match the value of each input pattern to each subspace as the weight of each DAG SVM. 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 can be used to analyze 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 new data fusion scheme 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 advanced classification approaches and the methods used to improve the accuracy of classification.

(Jipsa & Vkarunakaran, 2012) did a survey on image classification method and find Image classification is one of the most complex areas in image processing. It is more complex and difficult to classify if it is blurry and noisy material is included. There are many methods for classifying images and they provide good classification results, but when the image includes fuzzy and noisy information, they fail to provide satisfactory classification results. Supervised and unsupervised classification are the two primary methods for image classification. Both classifications have their own advantages and disadvantages. 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. Traditional classification approaches deal poorly on content based image classification tasks being one of the reasons of high dimensionality of the space of a function. Color image classification is performed in this paper on features derived from color component histograms. 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 design an efficient linear transformation operator that can optimize the inter-class to intra-class distance ratio while meeting the constraint of aligning the different class centers in different directions after transformation. Its key benefit over the conventional linear discrimination study of Fisher is that it is possible to accomplish the classification simultaneously with the transformation. The CLDA, i.e. the class spectral signatures, is a supervised approach. (Qian, 2007).

(Kumar et al., 2012) Its basic concept is 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 different class centers in different directions after transformation. Its key benefit over the conventional linear discrimination study of Fisher is that it is possible to accomplish the classification simultaneously with the transformation. The CLDA, i.e. the class spectral signatures, is a supervised approach.

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 characteristics of the picture classification. Two kinds of ants that have different search strategies and refreshing mechanisms are described. New categories are defined by stochastic ants; category tables are constructed and the clustering center of each category is calculated. The analysis shows that the ant-colony algorithm increases performance and precision.

CHAPTER THREE
METHODOLOGY

3.1 Introduction

This section shows the methodology design used to develop the image classification model in a detailed way. 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. I will take several pictures of these books with 'noise'. Noise in the sense that the pictures will be taken in different backgrounds. Pictures of the books will be taken with other objects to help increase the accuracy of the model so that it can identify these books even if a part of it is showing or it is really showing. I will take the pictures in different lightning backgrounds so that the model would be able to identify these objects even when there is poor lightning.

I will manually split the images into two directories, one for training and one for validation with the validation data being 20% of the whole data set. I will make sure the validation images are selected randomly.

3.3 Data Preprocessing

Some of the pictures I will take for the Image classification might be faulty like they might be dark, blurry, e.tc so I will have to sort them out and delete them. In Data preprocessing I will also need to convert the images 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 raw data into a comprehensible format. Real-world data is often 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.. I will need to clean the data 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

I will increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. Even when I have lots of data, it will help to increase the amount of relevant data in my dataset. This is related to the way with which neural networks learn. I will also use Data augmentation 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

I will create two models for this research purpose.

The first model will have its layers constructed by me, while the other would be a pre-trained model. I will use ImageDataGenerator to implement data augmentation for my training data and will shuffle generator. I will create a model 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 number of nodes as the layers increase. I will also use dropouts 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.

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

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

I will also load a pre-trained model, 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, I will also plot the training accuracy alongside the validation accuracy, and plot the training loss alongside the validation loss.

3.4.1 Convolutional neural network

With convolutional neural networks I will automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. 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, assign importance to various aspects/objects in the image (learnable weights and biases) and be able to distinguish one from the other. As compared to other classification algorithms, the pre-processing required in a ConvNet is much lower. Although filters are hand-engineered in primitive processes, ConvNets have the ability to learn these filters/characteristics with adequate training. (Sumit, 2018).

Convolutional Neural Networks perform better than other Deep Neural Network architectures because of their unique process. Instead of looking at the image one pixel at a time, CNNs group several pixels together (an example 3×3 pixel like in the image above) so they can understand a temporal pattern.

CNNs can see a group of pixels forming a line or a curve in another way. Due to the deep design of Deep Neural Networks, not a group of pixels, but groups of lines and curves forming 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 I will change the model in response to the estimated error each time the model weights are updated. Choosing learning rate can be challenging as a value too small may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process.

In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of loss function.

Learning rate is a hyper-parameter that controls the weights of our neural network with respect to the loss gradient. It defines how quickly the neural network updates the concepts it has learned. A desirable learning rate is low enough that the network converges to something useful, but high enough that it can be trained in a reasonable amount of time. The weights of a neural network cannot be calculated using an analytical method. Instead, the weights must be discovered via an empirical optimization procedure called stochastic gradient descent.

The problem of optimization addressed by stochastic gradient descent for neural networks is difficult and the solution space (weight sets) can consist of several good solutions (called global optima) as well as easy to find, but low in ability solutions (called local optima). During each stage of this search process, or the phase size, the amount of change to the model is called the "learning rate" and provides perhaps the most significant hyperparameter to tune into your neural network in order to achieve good performance on your problem. (Jason, 2019)

3.5 Training the Model

In training the model, I will first compile the model. The goal of training a model is to find a set of weights and biases that have low loss, on average. The process of training the ML model involves providing the ML algorithm (that is, the *learning algorithm*) with training data to learn from. The training data will contain the correct answer, which is known as a *target* or *target attribute*. The learning algorithm will then find patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

3.5.1Transfer Learning

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

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems. (Jason, 2017).

With transfer learning I will decrease the trained time for a neural network model and make it 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 performance of the model with parameters such as accuracy/recall. etc., to that of pre-determined accuracy with the model already built and moved into production. This is the most trivial of different techniques which could be used for blackbox testing (Kumar, 2018)

I will use the test data to check the accuracy of the model and display the result

CHAPTER FOUR

SIMULATIONS RESULTS AND DISCUSSIONS

4.1 INTRODCTION

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 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]:  1 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 4.1: 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.

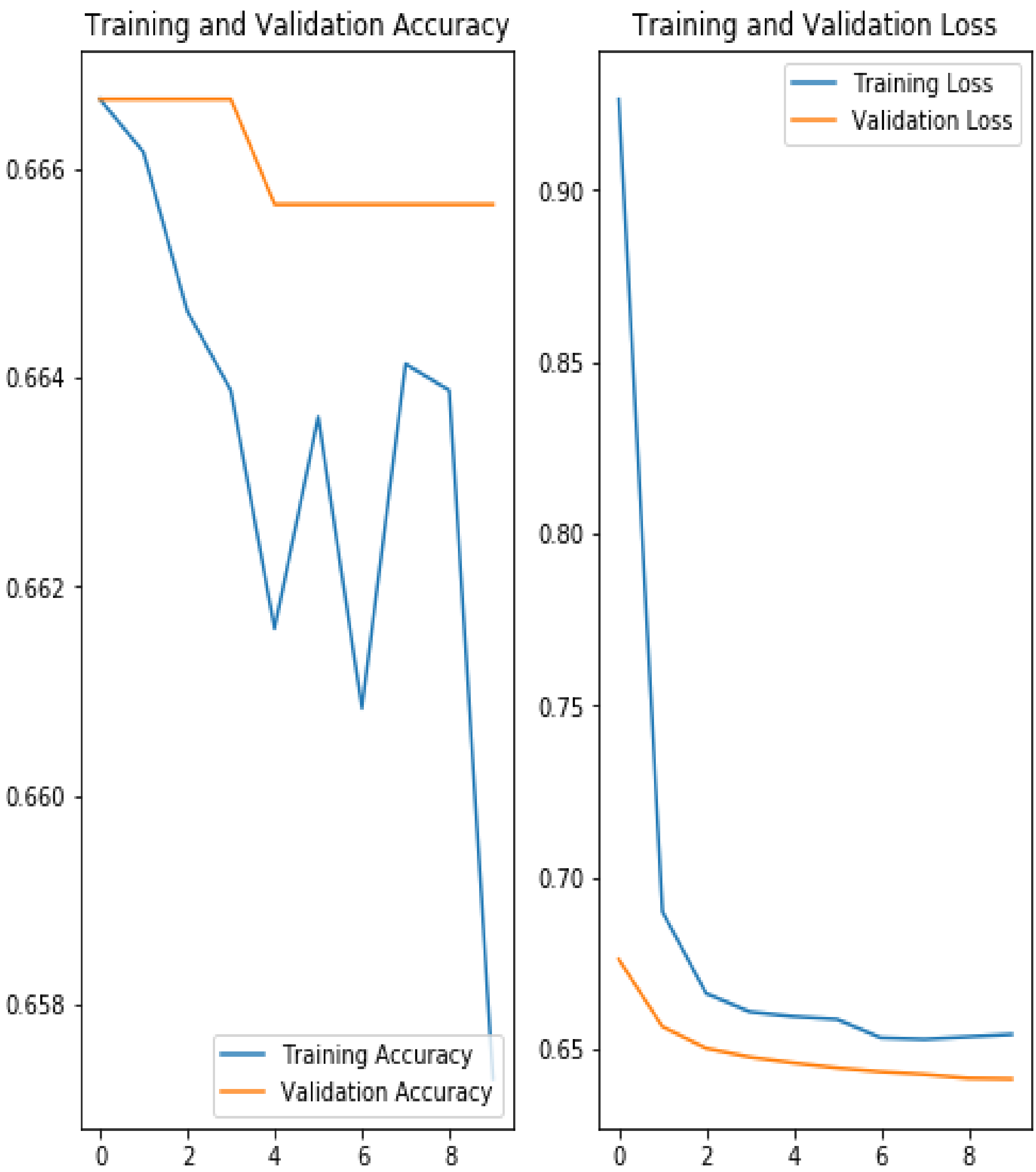


Figure 4.2: Training and Validation accuracy & loss

4.5 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.

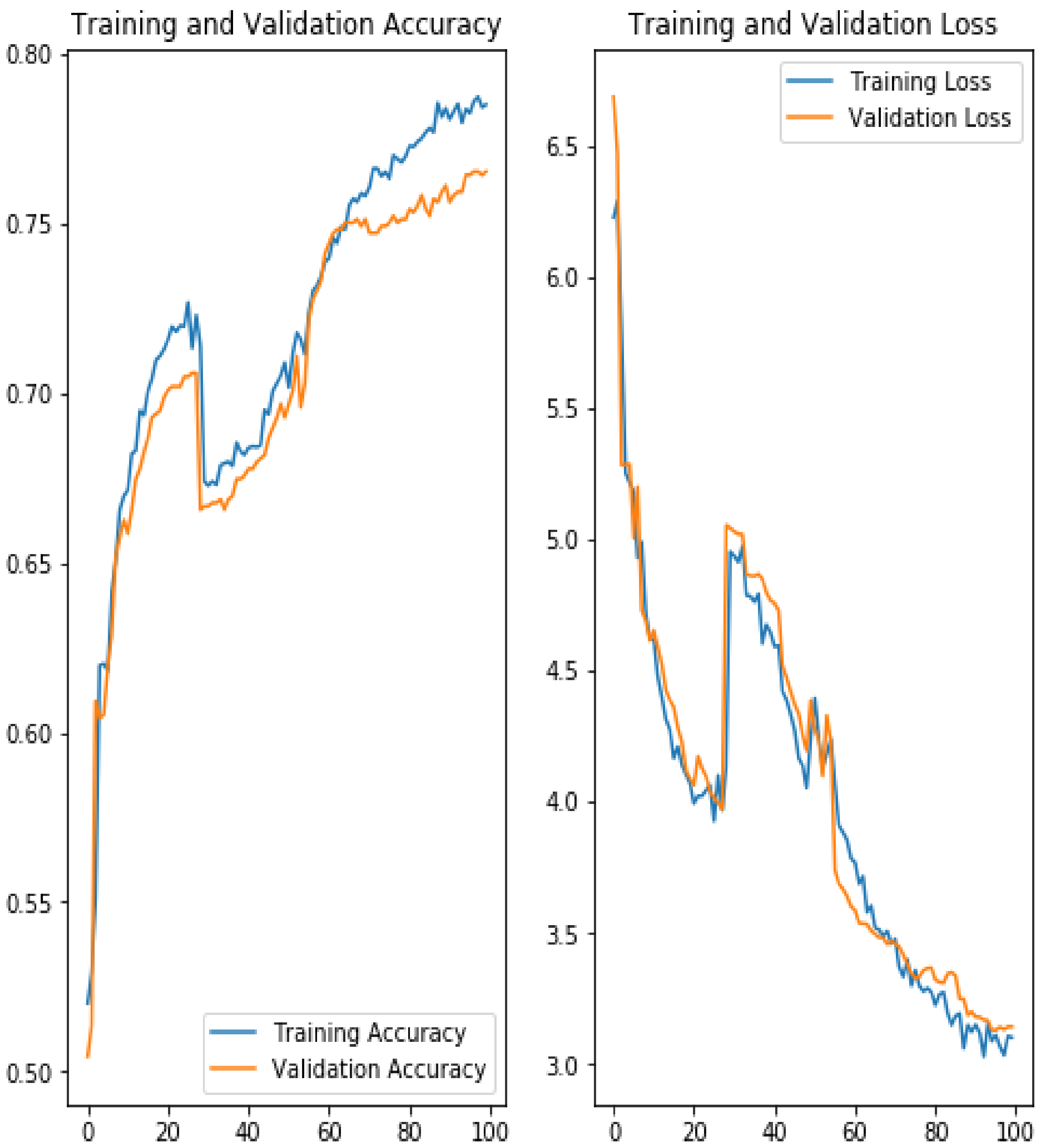


Figure 4.3: Training and Validation accuracy & loss

4.6 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.

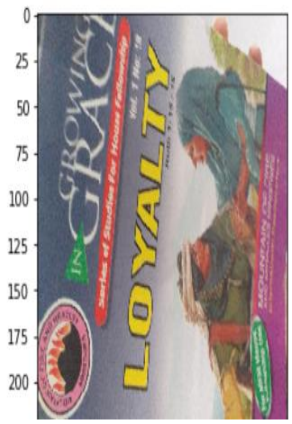
For loyalty book;

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
```



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

Figure 4.4: Picture of the first chapel book

For Rudiments of salvation.

```
In [51]:  1 plt.imshow(test_gen[0][0][11])
          2 print(df.iloc[11])
```

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

```
In [52]:  1 plt.imshow(test_gen[0][0][12])
          2 print(df.iloc[12])
```

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

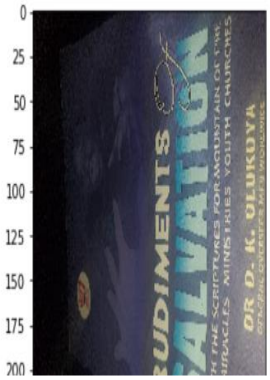


Figure 4.5: Picture of the second chapel book

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
```

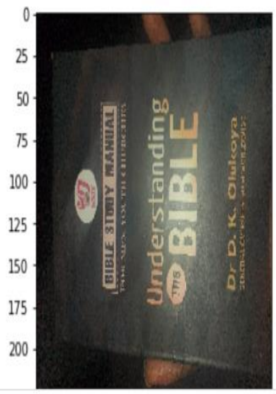


Figure 4.6: Picture of the Third chapel book

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 RECOMMENDATION

5.1 SUMMARY

Having seen the challenges encountered at the MTU 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 MTU 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 MTU 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 main contribution of knowledge was the ability to develop a model to help avoid stress at the MTU chapel using Artificial Neural network. With the help and use of Deep Learning technique, the codes worked successfully and ready for use. 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 to carry 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 detect objects such as stop signs and traffic lights

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 Image Classification using 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 designed to simulate the manner in which the human brain analyzes and processes 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 local tensor singular value decompositions (IEEE, international workshop on computational advances in multi-sensor adaptive processing. IEEE, Willemstad, 2018), pp.1-5.
2. X. Wang, C. Cheng, et al, Zero-shot image classification based on deep feature extraction. 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 large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248–255). Ieee.
9. 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).
10. 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).
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 survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345–1359.

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

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

16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (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. Deep learning for computer vision: A brief review. Computational intelligence and neuroscience, 2018.

19. Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320–3328).

20. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In European conference on computer vision (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()
```