# FRUIT CLASSIFICATION USING COVOLUTIONAL NEURAL NETWORK

## A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By MANAL DARWISH

In Partial Fulfillment of the Requirements for the Degree of Master of Science

In

**Computer Engineering** 

NEU

FRUIT CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

NICOSIA, 2020

## FRUIT CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

## A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By MANAL DARWISH

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

NICOSIA, 2020

## Manal Darwish: FRUIT CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

#### Approval of Director of Graduate School of Applied Sciences

#### **Prof. Dr. Nadire CAVUS**

#### We certify this thesis is satisfactory for the award of the degree of Masters of Sciences in Computer Engineering

**Examining Committee in Charge:** 

Prof. Dr. Rahib H. ABİYEV

freek

Assoc. Prof. Dr. Kamil DIMILILER

Assoc. Prof. Dr. Melike ŞAH DİREKOĞLU

Committee Chairman, Head of the Department of Computer Engineering, NEU

Committee Member, Department of Automotive Engineering, NEU

Supervisor, Committee Member, Department of Computer Engineering, NEU



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name: Manal Darwish

Signature:

Date: 1/7/2020

#### ACKNOWLEDGMENTS

I thank the almighty Allah for enabling me to carry this study successfully. A profound gratitude goes to my supervisor for her commitment and support during this study. Lastly, I will also like to appreciate my family for their support and backup as well as every other person that has contributed to the success of this study.

To my Family...

#### ABSTRACT

Food security is a very important topic of discussion in today's society, as improper handling and management of food during production, processing or distribution has caused increased food wastage around the globe. In addition, it has become clear from statistics gotten from surveys by institutions round the world like the Food Bureau of the United States that it is necessary to increase our rate of food production to meet the needs of our rapidly growing population. All of these points to a growing need for optimization of the resources at our disposal and to help in efficient production and management of our food resources.

Machine learning is a powerful tool that has been applied to many fields for the purpose of automation of basic operations and optimization of the results of these operations. In the area of food security, machine learning is being used on the field and in the market place to increase yield and ensure the quality of the fruits and vegetables reaching the consumers. Research is ongoing into the application of machine learning for inspection and grading of fruits and vegetables in retails stores to ensure accurate consistent reports on the quality of the produce sold to consumers. In addition, seeing as the visual assessment is the primary basis for a purchase choice in the market, ensuring the visual quality of fruits and vegetables is important to drive sales.

In this thesis, I would be developing a simple CNN to identify fruits in images. This would be a baseline effort for the development of a fruit classification system that can eventually be developed to identify bad fruits and vegetables and eventually be able to predict the expiration date of a fruit or vegetable. This system would help the human to reduce the time and effort needed for sorting of fruits at supermarkets and eliminates the need for direct contact with a lot of the farm produce along the supply chain. With new strains of viruses and bacteria causes health issues globally, being able to eliminate the unnecessary and inappropriate handling of farm produce by non-professionals through automation could help to solve problems of food poisoning among others.

Keywords: Machine Learning; CNN; Tensor Flow; Fruit Classification

#### ÖZET

Gıda güvenliği, üretim, işleme veya dağıtım sırasında gıdaların yanlış kullanımı ve yönetimi dünya çapında gıda israfının artmasına neden olduğu için günümüz toplumunda çok önemli bir tartışma konusudur. Ayrıca, Amerika Birleşik Devletleri Gıda Bürosu gibi dünyanın dört bir yanındaki kurumlar tarafından yapılan anketlerden elde edilen istatistiklerden, hızla büyüyen nüfusumuzun ihtiyaçlarını karşılamak için gıda üretim oranımızı arttırmanın gerekli olduğu açıkça görülmüştür. Bütün bunlar, elimizdeki kaynakların optimizasyonuna ve gıda kaynaklarımızın verimli bir şekilde üretilmesine ve yönetilmesine yardımcı olmak için artan bir ihtiyaca işaret etmektedir.

Makine öğrenimi, temel işlemlerin otomasyonu ve bu işlemlerin sonuçlarının optimizasyonu amacıyla birçok alana uygulanan güçlü bir araçtır. Gıda güvenliği alanında, verimi artırmak ve tüketicilere ulaşan meyve ve sebzelerin kalitesini sağlamak için sahada ve pazar yerinde makine öğrenimi kullanılmaktadır. Tüketicilere satılan ürünlerin kalitesi hakkında doğru ve tutarlı raporlar sağlamak için perakende mağazalarda meyve ve sebzelerin incelenmesi ve derecelendirilmesi için makine öğrenimi uygulaması konusunda araştırmalar devam etmektedir. Ayrıca, görsel değerlendirme olarak görmek pazardaki satın alma seçiminin temel dayanağıdır, meyve ve sebzelerin görsel kalitesinin sağlanması satışları artırmak için önemlidir.

Bu tezde, görüntülerdeki meyveleri tanımlamak için basit bir CNN geliştireceğim. Bu, sonunda kötü meyve ve sebzeleri tanımlamak için geliştirilebilen ve sonunda bir meyve veya sebzenin son kullanma tarihini tahmin edebilen bir meyve sınıflandırma sisteminin geliştirilmesi için temel bir çaba olacaktır. Bu sistem, insanların süpermarketlerdeki meyvelerin ayrılması için gereken zamanı ve çabayı azaltmasına yardımcı olur ve tedarik zinciri boyunca birçok çiftlik ürünü ile doğrudan temas ihtiyacını ortadan kaldırır. Yeni virüs ve bakteri türleri ile küresel olarak sağlık sorunlarına neden olur, otomasyon yoluyla çiftlik ürünlerinin profesyonel olmayanlar tarafından gereksiz ve uygunsuz bir şekilde işlenmesini ortadan kaldırmak, diğerleri arasında gıda zehirlenmesi sorunlarının çözülmesine yardımcı olabilir.

Anahtar Kelimeler: Makine Öğrenmesi; CNN; Tensör Akışı; Meyve Sınıflandırması

## **TABLE OF CONTENTS**

ACKNOWLEDGMENTS	II
ABSTRACT	iv
ÖZET	v
TABLE OF CONTENTS	vi
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1: INTRODUCTION	1
1.1 Thesis Problem	2
1.2 The Aim of the Thesis	3
1.3 The Importance of the Thesis	3
1.4 Limitations of the Study	4
1.5 Overview of the Thesis	5
CHAPTER 2: LITERATURE REVIEW	6
2.1 Smart Farming	7
2.2 Machine Learning on the field	9
2.3 Machine Learning on the market	11
CHAPTER 3: CONVOLUTIONAL NEURAL NETWORKS	13
3.1 Feature Extraction	14
3.2 Architecture of a Convolutional Neural Network	15
3.2.1 Convolution Layer	16
3.2.2 Pooling Layers	17
3.2.3 Output Layer	17

3.2.4 Activation Function	18
CHAPTER 4: SOFTWARE TOOLS AND CNN ARCHITECTURE	20
4.1 Software Tools	20
4.1.1 Matplotlib	20
4.1.2 NumPy	21
4.1.3 SciPy	21
4.1.4 Keras	21
4.2 Architecture details	22
CHAPTER 5: IMPLEMENTATION AND TRAINING OF THE CNN MODEL	24
5.1 Dataset	24
5.2 Required Modules and Libraries	25
5.3 Displaying and Loading the Dataset	26
5.4 Data Pre-processing	28
5.5 Designing the Model	30
5.6 Compiling and Training the Model	31
CHAPTER 6: EVALUATIONS AND RESULTS	32
6.1 Confusion Matrix	33
6.2 Limitations of the Study	35
CHAPTER 7 CONCLUSION AND FUTURE WORK:	36
7.1 Conclusion	36
7.2 Future Works	36
REFERENCES	37

AP	PENDICES	40
	Appendix 1: Import All Required Models	41
	Appendix 2: Visualizing Samples	42
	Appendix 3: Loading The Data	43
	Appendix 4: Pre-process the Dataset	44
	Appendix 5: Design The CNN	46
	Appendix 6: Train the Model	47
	Appendix 7: Calculate the Accuracy	48
	Appendix 8: Confusion Matrix	49
	Appendix 9: Prediction	50

### LIST OF TABLES

Table 4.1: The detailed architecture of the CNN	23
Table 5.1: Number of images	24

## LIST OF FIGURES

Figure 1.1:	Some sections of smart farming	. 2
Figure 2.1:	Real time event management in smart farming	. 8
Figure 2.2:	Machine learning applied to determine the ripeness of fruits on a tree	. 9
Figure 2.3:	Amazon Fresh user interface	11
Figure 3.1:	Biological neural unit (Neuron)	13
Figure 3.2:	Artificial neural unit (node)	14
Figure 3.3:	Typical structure of a Convolutional Neural Network	16
Figure 3.4:	Convolution and max pooling	17
Figure 3.5:	An arrangement of layers in a simple CNN	19
Figure 4.1:	The hierarchy of TensorFlow APIs	21
Figure 4.2:	Architecture details	22
Figure 5.1:	Importing all the required modules	25
Figure 5.2:	Script display a single sample of data from each of the 5 classes	26
Figure 5.3:	Displayed images for fruits in the 5 classes	27
Figure 5.4:	Script to load the data and split it into training and testing groups	28
Figure 5.5:	Script to convert the dataset to a 4-dimensional array to serve as input for	
	keras	29
Figure 5.6:	Script to make a histogram representation of the images	30
Figure 5.7:	Image histogram for the first image in the array	30
Figure 5.8:	Script to design the CNN model	31
Figure 5.9:	Compiling the CNN model	31
Figure 5.10:	Script to train the CNN model	32
Figure 6.1:	Script to run a prediction using the network	32
Figure 6.2:	Calculating training accuracy	33
Figure 6.3	Calculating testing accuracy	33
Figure 6.4:	Confusion Matrix	33
Figure 6.5:	Confusion Matrix code	34
Figure 6.6:	How to find Confusion Matrix	34
Figure 6.7:	Precision and Recall values	34
Figure 6.8:	Equations for Precision Recall	35

### LIST OF ABBREVIATIONS

AI:	Artificial Intelligent
API:	Application Programmer Interface
CNN:	Convolutional Neural Network
NN:	Neural Network
TPU:	Tensor Processing Unit
AWS:	Amazon Web Services

#### **CHAPTER 1**

#### **INTRODUCTION**

In recent years, there have been a lot of applications of machine learning in the day to day operations of organizations. A lot of these applications are major, such as in autonomous cars, virtual assistants, analysis of video surveillance, and so on. However, some of these applications are less visible but with meaningful impact on the basic operations of some organizations. One of this less visible application is the classification of fruit and vegetables. There have been a number of focuses regarding the applications of machine learning in fruit classification, with the main applications being for fruit identification and categorizing or grading.

The development of smart farming has featured machine learning being used for some data intensive aspects of agriculture. These systems include aspects for data collection on the field, data analysis, monitoring, prediction, and so on. These systems have also aided in agricultural research for studies on farming techniques and practices to help manage limited resources and adapt to climate change. All of these applications and research studies have led to farming with high yield and higher quality of the produce. Some of the aspects involved in smart farming are shown in Figure 1.



Figure 1.1: Some sections of smart farming

Machine learning has lent its strong automation and computation power to large scale farming in many ways, including fruit classification systems that have been used on the field for analysis and prediction of farm yield and disease and defect detection. These systems have also been used for research purposes to help understand and solve problems that plague the agricultural industry.

Also, off the field, for the preservation, transportation and inspection of agricultural produce, machine learning has played a role in helping professionals make more informed and effective decisions. From ware houses, like at Amazon, where fruit detection systems are being used by their produce inspection team to grade produce on a consistent basis (Day one Team, 2019). Computer Vision Scientists at Amazon have also applied artificial intelligence to develop systems that could predict if fruits and vegetables would go bad

soon or if they are sweet. All of the produce is run through a collection of cameras and sensors to gather data. Then this data is then fed into artificial intelligent (AI) systems to make inferences

#### **1.1** Thesis Problem

The way a fruit or vegetable looks is the first basis for which a purchase decision is made. The average person would not be able to track species and genetic traits of a fruit to be sure with any certainty whether a piece of fruit or vegetable is good or would taste sweet. To a good extent however, these conclusions drawn from the basic analysis of the fruit or vegetable by the customer can actually show the quality of the fruit which then means that the supplier or retailer needs to make sure that the fruits and vegetables they put on sale looks desirable, so as to drive up purchases. For this reason, A lot of time and effort is actually put into inspection and grading of fruits and vegetables at supermarkets and stores.

For the sake of quality assurance and as a healthcare measure, certified inspectors also inspect and grade fruits and vegetables. However, inconsistencies in their reports can also lead to high levels of wastage or bad purchases. Every false positive (that is bad fruits or vegetable graded as good) leads to a bad purchase, and every false negative (that is good fruit or vegetable graded as bad) leads to wastage. As a result of this, only about 60% of fruits and vegetables produced in the United States actually make it from the farm to the table (Farm Bureau, n.d.).

Furthermore, fruit identification also reduces the time and effort needed for sorting of fruits at supermarkets and eliminates the need for direct contact with a lot of the farm produce along the supply chain. With new strains of viruses and bacteria causes health issues globally, being able to eliminate the unnecessary and inappropriate handling of farm produce by non-professionals through automation could help to solve problems of food poisoning among others.

#### **1.2** The Aim of the Thesis

The aim of this thesis is to develop a system that can identify specific fruit in a group. I would be training a convolutional neural network (CNN) to be used to solve this problem. For this purpose, around 3,000 fruit images are used for the training of the network and around 1,000 images are used for the testing of the network. In the proposed fruit classification system, first I images are preprocessed; each image is represented as a color image histogram. This would record the tonal distribution of the image. The reason I have chosen this method is because the major signifier of a rotting or spoiling piece of fruit or vegetable is a change in color and color can be used as a criteria for fruit classification (Naik et al, 2017; Nishi et al, 2017). This way, my object identifier would only detect healthy looking fruits and vegetables.

This thesis would hopefully create a way to help in saving time and effort in the sorting process at supermarkets and also help to a certain level in the inspection and grading of fruits and vegetables.

#### **1.2 The Importance of the work**

The entire concept of smart farming has really opened up an opportunity to do more in the agricultural sector. And this is not just a good thing, but a necessary thing. According to the United states Farm Bureau, we would need to be producing 70% more food than we are now to support the world population by the year 2050. This therefore means that we would not only need more farms, but we would also need to be producing more food from each farm land than we are producing now. This is why it is important that we practice more effective farming and that we are able to study our farms and develop strategies to get better yields.

This objective requires increased and effective effort in 3 major areas;

• Increasing the yield of food crops and livestock for each farm

- Maintaining and possibly, increasing the quality of farm produce;
- Reducing the wastage of farm produce

All of these areas are avenues where artificial intelligence can be applied to help get better results. This thesis focuses aiding the reduction of wastage with the added benefit of helping store owners be able to provide quality produce to their customers and ensure the sale of their products.

#### 1.3 Limitations of the Study

Being able to make the object detection model be able to detect when a fruit is bad was an important aspect of this thesis. However, the lack of substantial data in the form of multiple images of spoilt fruits for all my fruit examples was available, so as an alternative, I have preprocessed my data by representing each fruit as an image histogram. This way, fruit with color defects would not be picked.

Organizations like Amazon who use fruit detection systems like the one I am developing have access to considerable large data and are still even training their models on large quantities of already graded fruit samples. My inability to access this level of resources serves as a constraint to my study.

#### **1.5** Overview of the Thesis

In the next chapter of this work, I would be giving a review of the new field of smart farming and the work and research that has already been done for the purpose of effective farming. I would also be giving a brief case study on Amazons fruit sorting system and discussing its benefits In the third chapter, I would be talking about convolutional neural networks which is the type of neural network I am employing here. I would be discussing how they are structured, and how a CNN architecture is decided.

In the fourth chapter, I would be discussing the design of the system and the software tools that I would be using in the development of the system. I would also be discussing my CNN architecture.

In the fifth chapter, I would be giving a walkthrough of my methodology and development process. I would be explaining all parts of my code in this section.

In the sixth chapter, I would be showing my results and running a few tests to show the effectiveness of my model.

Finally, in the seventh chapter, I would be discussing the conclusion and the future works that can be done to improve the system based on my results and conclusions.

#### **CHAPTER 2**

#### LITERATURE REVIEW

A lot of fields and industries have seen the intervention of machine learning and taken advantage of its powerful automation and power to increase efficiency and reduce cost and effort needed. As a result, these areas have had their basic operations revolutionized. Although there have been a few areas that have seen more research and work done by artificial intelligence, there are a few other areas that are also beginning to pick up. One of these areas is Agriculture.

Food security has always been an important point of focus in the world, as hunger and famine continues to grow. This has raised concerns for how we manage our food and how much food we can produce. This problem has been made even bigger as young people have continued to lose interest in farming because of little benefits associated with the profession and negative perceptions that are being spread about it (UN youth Envoy, n.d.). This has made farm owners to turn to cheap unskilled labor to keep their farms operational. This decision has led to poor farming choices and poor strategies that causes a drop in the yield and quality of farm produce.

#### 2.1 Smart Farming

Smart Farming involves the use of information and technology to increase the yield and quality of crops on livestock on a farm. It essentially increases the value of a farm. This is a necessary approach to agriculture as our food security needs continue to grow as our population grows, while problems such as the decline in interest in farming, bad farming practices, and climate change continue to reduce the yield and consequently, the effectiveness of farms.

What smart farming actually offers is an opportunity to have real time monitoring of farms that allows real time situation awareness and the gathering of data on all levels of the farming process for analysis that would lead to the development of strategies to optimize farming operations and help in making proper management decisions (Wolfelt et al, 2014). They have also been known to be used to even estimate farm yield Bargoti et al, 2017; Sa et al, 2016). In some cases, it can also help make real time management decisions.



Figure 2.1: Real time event management in smart farming

As with a number of other industries, smart farming is expected to grow to a point where much more of the processes are automated. Expert systems would dominate the smart analysis and planning section by making well informed inferences based on a well-developed knowledge base trained on extensive data (Ogidan et al, 2018).

The applications of smart farming are more visible on the field, but the effects of all the work also requires corresponding efforts off the field. There also needs to be supporting effort in the area of food transportation, inspection and sales. This thesis discusses the application of machine learning to a practice off the farm – Sorting and grading.

#### 2.2 Machine Learning on the Field

Machine learning applications on the field feature the application of big data. Modern farms are now being equipped with multiple sensors to gather data in real time and give situational awareness of what is happening on the field. This helps the farmers to know precisely what is happening on the field rather than just making guesses and making decisions based on hypothesis. Figure 2.3 shows an example of machine learning on the field to determine the ripeness of fruits on a tree.



Figure 2.2: Machine learning applied to determine the ripeness of fruits on a tree

Although there have been high expectations for the field of big data, there have also been speculations on how long the field would be seen as valuable. Some researchers have even said that the expectations for big data are inflated and would soon fall out of trend (Fenn et al, 2011). In some countries where smart farming concepts and tools have been adopted, there have been a few failures that have made the current reality to vary from the expected results, but this is mostly based on some errors exposed by structural analysis of the systems (Lamprinopoulou et al, 2014). Some of the limitations that have slowed the progress of smart farming include;

- Infrastructural deficiency such as roads, telecommunications infrastructure, research and development facilities, etc. Also, the quality of the system being used falls under this category. Sometimes, the system might be of low quality due to the use of bad data or a poor system architecture.
- Weak institutional involvement in the operations of farms. If there must be improvement in the yield and quality of farm produce, government institutions need to be more involved in the operations of farmers to adjust regulations accordingly and also create an avenue for farmers to share ideas and learn new techniques to aid collective growth. Having just a few farms employ new technology and methods while others stick to business as usual would not help solve the problems.

#### 2.3 Machine Learning on the Market

The Practices on the market level also have a role to play in food security. The way a fruit or vegetable looks is the main factor behind a customer's purchase decisions. This is also why inspection and grading are important to the retailers, as customers would always prefer to purchase highly graded fruits and vegetables.

Amazon have been very open to the idea of using machine learning to aid in their basic operations and have even started up entire services fully based on machine learning, such as automated checkout with Amazon Go, package delivery with Prime Air, style recommendation with Echo Look, virtual assistance with Alexa, fruit sorting with Amazon Fresh, etc. This has helped to increase their efficiency and accuracy, and reduce cost.



Figure 2.3: Amazon Fresh user interface

With Amazon Fresh, they are able to sort through large amounts of fruits and vegetables and grade them on a consistent basis, reducing wastage and ensuring that the fruit that they put up for sale are of high quality.

The system is trained on a very extensive dataset and is able to spot defects. More research on Amazon fresh is also expected to make the system be able to predict if a fruit or vegetable is going to be sweet or not. The system is also being developed to be able to predict when a fruit or vegetable would go bad.

#### **CHAPTER 3**

#### **CONVOLUTIONAL NEURAL NETWORKS**

Neural networks are a method of machine learning that emulates the way the human brain works. The first efforts in the development of a neural network were in 1943 by a neurophysiologist named Warren McCulloch and a mathematician named Walter Pitts. They wrote a paper discussing how neurons work and made and electrical model of a neural network. Donald Hebb then defined some of the more known laws and concepts of Neural networks first with his book in 1949 titled *The Organization of Behavior*, and then with a more publications and articles. Since then, there has been a lot of research that has led to the development of a number of other neural network Models.

Figure 3.1 shows the biological neuron while figure 3.2 shows the artificial neural unit known as a node and shows its similarities to the biological neuron.



Figure 3.1: Biological neural unit (Neuron)



Figure 3.2: Artificial neural unit (node)

Convolutional neural networks are a type of deep neural networks that are mostly used in the field of computer vision. They are composed of fully connected layers of multiple nodes where the output of each node is adjusted according to the input value, and the nodes are lined up so that the output of one node is the input of another node (O'Shea et al, 2015). The entire architecture of a convolutional neural network designed for classification problems can be split into two parts with two objectives. One-part handles feature extraction and selection, while the other part handles classification.

#### **3.1 Feature Extraction**

Neurophysiologists, David H. Hubel and Torsten N. Wiesel in their research explored the human visual cortex and explained that humans see by first extracting local features from small sections of the image coming to our eyes, and then combining these extracted features by pooling. In simple terms, when we see an image, we scan the image and pick out specific features of the image, then we combine all these features and decide what we believe the image is based on the features we have gathered. Feature extraction is a very important part of neural networks because number of features gathered and the usefulness of the feature gathered could affect the efficiency of the neural network. In neural network

applications, Feature extraction methodologies are usually followed by feature selection, to reduce the number of features to a few more important features. This is because having many features increases the volume of the space. This increased volume of the space can make the data sparse unless much more data is made available for analysis. This phenomenon is known as the *curse of dimensionality*.

#### 3.2 Architecture of a Convolutional Neural Network

Convolutional neural networks have an input layer and an output layer with multiple hidden layers. These hidden layers are intertwined and connected fully. The more complicated the convolution is, the more sophisticated the network is, although that would require more computation and more time for training.

There have been improvements on the design of convolutional neural networks over the years, primarily in the layer designs and some optimization techniques to boost speed and accuracy. Improvements have also been aided by the growth of the amount of annotated data and improvement in the power of processing units (Gu et al, 2018).

There are different types of layers in a typical convolutional neural network that serve different purposes within the network, such as the pooling layers, the convolutional layers, the fully connected layer and the output layer. They also consist of an activation function. Figure 3.3 shows a simple typical structure of a CNN.



Figure 3.3: Typical structure of a Convolutional Neural Network

#### **3.2.1 Convolution Layer**

This layer fall under the feature extraction aspect of the CNN. In image classification applications, they are made up of matrices of image pixels known as convolution kernels. These convolution kernels represent specific features of the image and are used to make a feature map by convolving the input with it based on an activation function. Many of these convolutional layers are laid out in series and the output of the previous layers are the inputs of the next layer. Having more convolutional layers improves the accuracy of the network but also increases the computation time and could cause overfitting (Brownlee, 2019).

Overfitting happens when the network is designed too specifically to a particular set of data, that it is no longer flexible enough to accommodate new data and make accurate predictions in the future.

#### **3.2.2 Pooling Layers**

Convolutional layers are normally followed by Pooling layers. The pooling layers are used to offset the complexity of the network caused by the convolution, and also to reduce the possibility of overfitting by downsizing the height and width of the feature map. The depth remains the same, as the pooling operation is performed on each slice of the input feature kernel (Tabian et al, 2019). Pooling serves as a form of feature selection. The major forms of pooling are average pooling and max pooling with max pooling being the most popular of them.

1	0	0	1	0	1	6	3	1			
1	1	0	0	1	7	3	5	2		7	
0	0	1 <sub>x0</sub>	1 *1	0	0	4	1	2	Max Pooling with filter	0	
1	0	0 <sub>×0</sub>	0 🗙	1	9	4	1	3	size (2,2)	9	
0	1	1 🗙	1 📩	0,1	5	8	4	1			
		(a)							(b)		

**Figure 3.4:** Convolution and max pooling (a) convolution operation with the feature kernel (for feature extraction) over an input slice. (b) a form of pooling, known as max pooling with a filter size of (2,2) selecting input with similar positioning (essentially applying the same pooling operation to the parts of the input) from the feature kernel

#### 3.2.3 Output Layer

This layer handles the actual output or inference of the network. It has its own type of activation function. For classification problems, the most popular activation function is known as SoftMax. This activation function normalizes the final output vector and returns the vector with a range of probabilities. The position that has the maximum probability is declared as the predicted class.

#### **3.2.4 Activation Function**

The activation functions are used to perform some operation on the input to a neuron and define the output of the neuron. The determine if a neuron is fired or not. This operation also serves the purpose of causing non linearity between the input and the output. There are different activation functions that are applied for different expected results. The most popular are:

- Sigmoid function
- Tanh function
- ReLu function
- Leaky ReLu function
- SoftMax function

For image classification problems like in these theses, the ReLu activation function is used the most. In ReLu, there are two possible outputs. For negative inputs, the output is equal to zero, while for positive inputs, the output is the input itself. Equation 1 shows the mathematical representation of the ReLu activation function.

$$ReLu(X) = \begin{cases} 0 & x < 0 \\ x & x \ge 0 \end{cases} = max(0, x)$$
(1)

All of these components can be arranged in any order to serve the purpose for which the network is being developed. Trial and error can also be applied to the network to improve the system. Figure 3.5 shows an arrangement of layers for a convolutional neural network used in classifying an image of a car.



Figure 3.5: An arrangement of layers in a simple CNN

#### **CHAPTER 4**

#### SOFTWARE TOOLS AND CNN ARCHITECTURE

A number of tools and libraries were needed for the different stages of the implementation of my system. There were major tools such as the high-level API, *keras* from TensorFlow which were used at the core of the system, and other small libraries like *matplotlib* for displaying images. In this chapter, I would be briefly discussing these tools and the reasons they were chosen.

#### **4.1 Software Tools**

First of all, I decided to write the system in python programming language, because it is one of the simplest programming languages and is easy to use with TensorFlow. Also, there are a number of python libraries that would be helpful for my objective.

#### 4.1.1 Matplotlib

Matplotlib is a python library that can be used to create visual representation of mathematical data or simply static or animated visuals. It also has an object-oriented API that allows you to embed these image representations into other applications. It can be used to make line plots, histograms, scatter plots, polar plots, 3D plots, etc. It is an open source software and therefore has very little restriction on its use. Within my system I would be using matplotlib to display images of fruits where necessary and to plot the image histogram representation of the fruit images.

#### 4.1.2 NumPy

Numpy is a python library used for making, processing and operating on matrices and multidimensional arrays. This would be used to handle and process all data of the array data type within the project and also to help with the convolution and and pooling operations on the data

#### 4.1.3 SciPy

SciPy is a library that is used mainly for citify computations such as interpolation, integration, linear algebra and image processing in python programming language.

#### **4.1.4 Keras**

Keras is a high-level object-oriented API that is a part of TensorFlow. It allows fast prototyping and can be used for major research and production applications. It is user-friendly, modular and easily extendable to allow users to make their own components and use them in their projects. Another high-level API for TensorFlow is the Estimator, although it is no longer popular. The estimator provided major functionality to train, evaluate, predict and export to serve. Figure 4.1 shows the hierarchy of TensorFlow APIs.





#### 4.2 Architecture details

Here I would be discussing the detailed architecture of my CNN. Which is supervised learning algorithm. First, I applied 16 filters at the first convolution, the doubled up until 128 at the fourth convolution. Then I had max pooling layers with pool sizes of 2x2 to down sample the outputs of the convolution layers. I paired the convolution and pooling layers and placed then at intervals four times, and finished off with a dropout layer to avoid overfitting. Then to connect the feature extraction part to the actually classification part for the fully connected layers, I used a flattening layer. I then connected one more hidden layer of 500 nodes with the ReLu activation function.

I chose the Relu activation function because the network will be able to train much faster if we use sigmoid or non-linear functions, because of the computational efficiency without making as significant difference to the accuracy.

Finally, the output layer has 5 nodes for the 5 object classes, with a SoftMax activation function. Table 4.1 shows the details of the CNN architecture. The training carried out in five epochs, Stochastic Gradient Decent is used as a learner with a learning rate 0.001. The total number of the parameters was 12,590,445.



Figure 4.2: Architecture details

Layer	Туре	Output Shape	Parameters	
conv2d_1	Convolution	(none,224,224,16)	208	
max_pooling2d_1	Max pooling	(none,112,112,16)	0	
conv2d_2	Convolution	(none,112,112,32)	2080	
max_pooling2d_2	Max pooling	(none,56,56,32)	0	
conv2d_3	Convolution	(none,56,56,64)	8256	
max_pooling2d_3	Max pooling	(none,28,28,64)	0	
conv2d_4	Convolution	(none,28,28,128)	32896	
max_pooling2d_4	Max pooling	(none,14,14,128)	0	
dropout_1	Dropout	(none,14,14,128)	0	
flatten_1	Flatten (fully	(none,25088)	0	
	connected)			
dense_1	Dense	(none,500)	12544500	
dropout_2	ppout_2 Dropout		0	
dense_2	Dense	(none,5)	2505	

 Table 4.1: The detailed architecture of the CNN

#### **CHAPTER 5**

#### IMPLEMENTATION AND TRAINING OF THE CNN MODEL

After getting choosing the tools for my development and determining my neural network architecture, I went into the development part of my work. I followed a series of steps which would be discussed within this chapter, I did all of my work on Google Colab because it allows me to execute the code on Google Clouds server, and take advantage of Google hardware, such as GPUs and TPUs.

#### 5.1 Dataset

For this sort of applications, considering that they are new and not too popular, there are not so many datasets available for research. However, I was able to access a dataset on Kaggle called Fruits 360. The Fruit 360 dataset consists of 55,199 images of 81 categories of fruits with the size of 100x100. The dataset is a good dataset that is still continuously being improved, with modifications even less than a month before the time of writing this. For my project, I chose to use only 5 classes of fruits (banana, dates, onions, pear and strawberries). This made my selected dataset to consist of 4,002 fruit images. I split the dataset into two groups for training and testing, with 2,952 images for training and 1,050 images for testing.

Table	5.1:	Num	ber of	images
-------	------	-----	--------	--------

Classes	Images	Testing Images	Training Images
Banana	664	170	494
Strawberry	991	250	741
Date	837	247	590
Pear	900	228	672
Onion	610	155	455
Total	4,002	1,050	2,952

#### **5.2 Required Modules and Libraries**

Here, I imported all the Required modules and libraries I need for the development of the system. This includes all libraries discussed in Chapter 4. I had Matplotlib for plotting and displaying images, *NumPy* for the matrix representation and operations involved in the development of the convolutional neural network, and *SciPy* mainly for its image processing capabilities.

I also imported a few modules from TensorFlow's keras API such as *keras.layers*, *keras.preprocessing*, *keras.utils* and *keras.models*. These would be used within the core of the machine learning operations of the system.

I also included a few miscellaneous modules for smaller tasks within the system, such as *glob*, to allow me have pathname pattern expansion like we have in unix, *os*, to allow me have some basic system functionality, and *tqdm*, which is a simple module that provides a progress bar. These modules have no direct impact on the effectiveness or structure of the CNN.



Figure 5.1: Importing all the required modules

#### 5.3 Displaying and Loading the Dataset

After importing all my modules, I went on to use the *matplotlib* plot function to display individual fruit from the annotated classes to make sure the images display and to test the annotations. Figure 5.2 shows the script to display five fruits from each of the 5 classes while Figure 5.3 shows the displayed images.



Figure 5.2: Script display a single sample of data from each of the 5 classes



Figure 5.3: Displayed images for fruits in the 5 classes

After confirming that the images are good and the annotations are correct, I loaded the data. To do this, I created a function called *load\_dataset* where I loaded each image file name into an array of file names and then used *np\_utils.to\_categorical* to map my array vector to a binary matrix of the file name targets. I then split my file names and their corresponding targets into training and testing groups. Figure 5.3 shows the script to load the data from the dataset and split it into training and testing groups at a 70:30 ratio.



Figure 5.4: Script to load the data and split it into training and testing groups

#### **5.4 Data Pre-processing**

Here I had to prepare the data for use within the system. CNNs designed using TensorFlow's keras API requires a 4-dimensional array with the structure [nb\_samples, rows, columns, channels].

*Nb\_samples* represent the number of samples in the dataset. Row, column and channel represent the rows columns and channels of each image sample in the dataset. To create the 4-dimensional array needed for *keras*, I wrote a function called path\_to\_tensor that converts the file names to the 4-dimensional array object. It first loads the image and resizes it to size 224x224 because the images would all have to be the same size. The images would have 3 channels, since we are working with images using the RGB representation. As a result, my returned tensor would have the shape (1,224,224,3). Figure 5.4 shows the script for the conversion.



Figure 5.5: Script to convert the dataset to a 4-dimensional array to serve as input for keras

Then for the purpose of my design, I needed to represent my images as a histogram, highlighting the RGB values, because the colors of the fruit could help show if the fruit was bad or good. Figure 5.5 shows the script to make histogram representations of the images, and Figure 5.6 shows an example histogram for the first image in the array.



Figure 5.6: Script to make a histogram representation of the images



Figure 5.7: Image histogram for the first image in the array

#### **5.5 Designing the Model**

Here, I designed the model based on my determined CNN architecture. I simply had 4 groups of convolution and max pooling pairs, with ReLu activation functions on the convolutions and a pool size of 2x2, and an output layer with SoftMax activation function. Figure 5.7 shows the script for designing the CNN model.



Figure 5.8: Script to design the CNN model

#### 5.6 Compiling and Training the Model

After Creating the CNN model, the next step was to compile and Train the model on the dataset. Figure 5.8 shows the script for compiling the model, and Figure 5.9 shows the script for training the model.



Figure 5.10: Script to train the CNN model

#### **CHAPTER 6**

#### **EVALUATIONS AND RESULTS**

The system was compiled and trained properly. We use 2,952 images for the training of the network over 5 epochs and it took about 97 second for every epoch to train which is reasonable considering the size of the dataset and the complexity of the neural network. I tried to classify images for fruits from the five classes and it took about 0.387 second in average time for classifying a single image (running the classification 10 times and taking its average). For testing, in total, we have 1,050 fruit images.

To run a prediction, I wrote a function called *test\_model*. Figure 6.1 shows the script for this function.



Figure 6.1: Script to run a prediction using the network

I calculated my accuracy for training and testing with formulae as shown in the scripts in Figure 6.2 and Figure 6.3.



Figure 6.2: Calculating training accuracy



Figure 6.3: Calculating Testing accuracy

The training accuracy was 99.8645% and the testing accuracy was 99.2381%.

#### 6.1 Confusion Matrix

The confusion matrix or error matrix is a visual representation of the effectiveness of an algorithm in machine learning or statistical classification. The rows represent the predicted instances and the columns represent the actual instances or ground truths. Figure 6.4 shows the confusion matrix for my model for all of the 5 classes and Figure 6.5 shows the code to derive the confusion matrix using the *sklearn.metric* function *confusion\_matrix*.



Figure 6.4: Confusion Matrix



Figure 6.5: Confusion Matrix code

Confusion matrix also can present precision and recall:

$$Precision = \frac{True \ Positive}{True \ Positive + False \ Positive}$$

Recall = <u>True Positive</u>+False Negative

Figure 6.6: Equations for Precision Recall







Figure 6.8: How to find Confusion Matrix

(TN) True Negative: The actual value was False, and the model predicted False.

(FP) False Positive: The actual value was False, and the model predicted True.

(FN) False Negative: The actual value was True, and the model predicted False.

(TP) True Positive: The actual value was True, and the model predicted True.

#### 6.2 Limitations of the Study

In the development of the system to achieve my objective, I had a few challenges. The major challenge was in making the system to be able to detect bad or spoilt fruit. A lack of datasets created for this purpose made it impossible. However, variations in the color pigment of good fruit and bad fruit can cause bad fruit to not be detected since the images are represented as image histograms. Although this seems like a temporary fix, it would not be as sophisticated as a system trained on data containing images of actual bad fruits. Also, I was not able to test this hypothesis.

Bigger organizations such as Amazon with Amazon Fresh have had to train their networks on Realtime data as they go about serving their customers. This gives them an advantage. Also, considering that there are many ways a bad fruit can look, with discoloration or patches at any point on the fruit, this would require a very extensive dataset.

#### **CHAPTER 7**

#### **CONCLUSION AND FUTURE WORK**

#### 7.1 Conclusion

Application of machine learning for fruit detection and sorting is very valuable in the agricultural sector. This could help a lot in food security and to also allow retailers and suppliers to provide high quality produce to their customers. This is one of the quieter but very important ways we can use machine learning to help in basic operations and increase the quality of goods and services.

The system developed in this thesis takes advantage of the computer vision capabilities of convolutional neural networks to help in fruit classification and sorting, however, the system can still be improved to do even more.

#### 7.2 Future Works

A few organizations like Amazon and some large-scale farms have been able to see the value of including machine learning to aid in their daily operations. This application has shown many benefits to them and helped them reduce cost and increase efficiency, however, there is still more that can be done.

Computer vision experts at Amazon are still doing research into how this system can be improved to predict when a fruit is going to go bad and to tell if a fruit would be sweet. This would require much more data and computation power, the like that provided by Amazon with the Amazon Web Services (AWS).

In conclusion, Fruit classification is an important research are that should be looked into and have more research and development resources because its development and improvement can have a far-reaching effect on agriculture and the quality of fruits provided at markets and retail stores.

#### REFERENCES

- Day One Team. (2019, August 02). Machine Learning: Using algorithms to sort fruit. Retrieved June 01, 2020, from https://www.aboutamazon.eu/innovation/machinelearning-using-algorithms-to-sort-fruit
- *Fast Facts About Agriculture & Food.* (n.d.). Retrieved June 06, 2020, from https://www.fb.org/newsroom/fast-facts
- *Why are rural youth leaving farming?* (n.d.). Retrieved June 06, 2020, from https://www.un.org/youthenvoy/2016/04/why-are-rural-youth-leaving-farming/
- S. Wolfert, D. Goense & C. A. G. Sørensen,(2014) A Future Internet Collaboration Platform for Safe and Healthy Food from Farm to Fork, 2014 Annual SRII Global Conference, San Jose, CA, pp. 266-273, doi: 10.1109/SRII.2014.47.
- Fenn, J., & LeHong, H. (2011). Hype cycle for emerging technologies. Gartner, Stamford.
- Lamprinopoulou, C., Renwick, A., Klerkx, L., Hermans, F., & Roep, D. (2014). Application of an integrated systemic framework for analysing agricultural innovation systems and informing innovation policies: Comparing the Dutch and Scottish agrifood sectors. Agricultural Systems, 129, 40-54. doi:10.1016/j.agsy.2014.05.001
- Ogidan, E. T., Dimililer, K., & Ever, Y. K. (2018). Machine Learning for Expert Systems in Data Analysis. 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). doi:10.1109/ismsit.2018.8567251
- O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. ArXiv, abs/1511.08458.
- Khandelwal, R. (2018, October 18). Convolutional Neural Network(CNN) Simplified. Retrieved June 08, 2020, from https://medium.com/datadriveninvestor/convolutionalneural-network-cnn-simplified-ecafd4ee52c5

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377.

Brownlee, J. (2019). Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python. Machine Learning Mastery.

- Tabian, I., Fu, H., & Khodaei, Z. S. (2019). A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures. *Sensors*, 19(22), 4933. doi:10.3390/s19224933
- CS231n Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved June 10, 2020, from https://cs231n.github.io/convolutional-networks/
- Oltean, M. (2020, May 18). Fruits 360. Retrieved May 25, 2020, from https://www.kaggle.com/moltean/fruits
- Naik, S., & Patel, B. (2017). Machine Vision based Fruit Classification and Grading A Review. International Journal of Computer Applications, 170(9), 22-34. doi:10.5120/ijca2017914937
- Nishi, T., Kurogi, S., & Matsuo, K. (2017). Grading fruits and vegetables using RGB-D images and convolutional neural network. 2017 IEEE Symposium Series on Computational Intelligence (SSCI). doi:10.1109/ssci.2017.8285278
- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & Mccool, C. (2016). DeepFruits: A Fruit Detection System Using Deep Neural Networks. *Sensors*, 16(8), 1222. doi:10.3390/s16081222
- Wang, S., & Chen, Y. (2018). Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. *Multimedia Tools and Applications*, 79(21-22), 15117-15133. doi:10.1007/s11042-018-6661-6

- Bargoti, S., & Underwood, J. P. (2017). Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards. *Journal of Field Robotics*, 34(6), 1039-1060. doi:10.1002/rob.21699
- Femling, F., Olsson, A., & Alonso-Fernandez, F. (2018). Fruit and Vegetable Identification Using Machine Learning for Retail Applications. 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). doi:10.1109/sitis.2018.00013
- Saranya, N., Srinivasan, K., Kumar, S. K., Rukkumani, V., & Ramya, R. (2020). Fruit Classification Using Traditional Machine Learning and Deep Learning Approach. *Computational Vision and Bio-Inspired Computing Advances in Intelligent Systems* and Computing, 79-89. doi:10.1007/978-3-030-37218-7\_10

APPENDICES

## APPENDIX 1 IMPORT ALL REQUIRED MODELS

import matplotlib.pyplot as plt import matplotlib.image as img import numpy as np #from scipy.misc import imresize %matplotlib inline from sklearn.datasets import load files from keras.utils import np utils import numpy as np from glob import glob from keras.preprocessing import image from tqdm import tqdm from IPython.display import display from PIL import Image from keras.layers import Conv2D, MaxPooling2D, GlobalAverageP ooling2D from keras.layers import Dropout, Flatten, Dense from keras.models import Sequential import os from os import listdir from os.path import isfile, join

## APPENDIX 2 VISUALIZING SAMPLES

```
root dir = '/content/drive/My Drive/fruits-360/Training'
rows = 3
cols = 2
fig, ax = plt.subplots(rows, cols, frameon=False, figsize=(15
, 25))
fig.suptitle('Random Image from Each Food Class', fontsize=20
)
sorted food dirs = sorted(os.listdir(root dir))
for i in range(rows):
    for j in range(cols):
        try:
            food dir = sorted food dirs[i*cols + j]
        except:
            break
        all files = os.listdir(os.path.join(root dir, food di
r))
        rand img = np.random.choice(all files)
        img = plt.imread(os.path.join(root dir, food dir, ran
d img))
        ax[i][j].imshow(img)
        ec = (0, .6, .1)
        fc = (0, .7, .2)
        ax[i][j].text(0, -20, food dir, size=10, rotation=0,
                ha="left", va="top",
                bbox=dict(boxstyle="round", ec=ec, fc=fc))
plt.setp(ax, xticks=[], yticks=[])
plt.tight layout(rect=[0, 0.03, 1, 0.95])
```

## APPENDIX 3 LOADING THE DATA

```
# define function to load train and test datasets
def load dataset(path):
    data = load files(path)
    fruit files = np.array(data['filenames'])
    fruit targets = np utils.to categorical(np.array(data['ta
rget']),5)
    return fruit files, fruit targets
# load train and test datasets
train files, train targets = load dataset('/content/drive/My
Drive/fruits-360/Training')
test files, test targets = load dataset('/content/drive/My Dr
ive/fruits-360/Test')
# load list of fruits names
fruit names =[item[:] for item in sorted(glob("/content/drive
/My Drive/fruits-360/Training/*"))]
print(fruit_names)
# print statistics about the dataset
print('There are %d total fruit categories.' % len(fruit name
s))
print('There are %s total fruit images.\n' % len(np.hstack([t
rain_files, test files])))
print('There are %d training fruit images.' % len(train files
))
print('There are %d test fruit images.'% len(test files))
```

## APPENDIX 4 Pre- PROSSEC THE DATASET

```
def path to tensor(img path):
    # loads RGB image as PIL.Image.Image type
    img = image.load img(img path, target size=(224, 224))
    # convert PIL.Image.Image type to 3D tensor with shape (2
24, 224, 3)
    x = image.img to array(img)
    # convert 3D tensor to 4D tensor with shape (1, 224, 224,
 3) and return 4D tensor
    return np.expand dims(x, axis=0)
def paths to tensor(img paths):
    list of tensors = [path to tensor(img path) for img path
in tqdm(img paths)]
    return np.vstack(list_of_tensors)
from PIL import ImageFile
ImageFile.LOAD TRUNCATED IMAGES = True
# pre-process the data for Keras
train tensors = paths to tensor(train files).astype('float32'
)/255
test tensors = paths to tensor(test files).astype('float32')/
255
```

```
import cv2
import numpy as np
```

```
from matplotlib import pyplot as plt
img = cv2.imread('/content/drive/My Drive/fruits-
360/Training/Pear/150_100.jpg')
color = ('b','g','r')
for i,col in enumerate(color):
    histr = cv2.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.plot(histr,color = col)
    plt.xlim([0,256])
plt.show()
```

## APPENDIX 5 DESIGN THE CNN

```
#Defining the architecture
model = Sequential()
model.add(Conv2D(filters=16, kernel size=2, padding='same',ac
tivation='relu',input shape=(224,224,3)))
model.add(MaxPooling2D(pool size=2))
model.add(Conv2D(filters=32, kernel size=2, padding='same',ac
tivation='relu'))
model.add(MaxPooling2D(pool size=2))
model.add(Conv2D(filters=64, kernel size=2, padding='same',ac
tivation='relu'))
model.add(MaxPooling2D(pool size=2))
model.add(Conv2D(filters=128, kernel size=2, padding='same',a
ctivation='relu'))
model.add(MaxPooling2D(pool size=2))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(500, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(5, activation='softmax'))
model.summary()
```

```
model.compile(optimizer='rmsprop', loss='categorical_crossent
ropy', metrics=['accuracy'])
```

## APPENDIX 6 TRAIN THE MODEL

from keras.callbacks import ModelCheckpoint

### TODO: specify the number of epochs that you would like to use to train the model.

epochs = 5

### Do NOT modify the code below this line.

checkpointer = ModelCheckpoint(filepath='/content/drive/My Dr ive/saved models/weights.best.from scratch.hdf5',

```
verbose=1, save_best_only=True
```

)

model.fit(train\_tensors, train\_targets,

```
validation_split=0.2,
```

epochs=epochs, batch\_size=20, callbacks=[checkpoint
er], verbose=1)

model.load\_weights('/content/drive/My Drive/saved\_models/weig hts.best.from\_scratch.hdf5')

## APPENDIX 7 CALCULATE THE ACCURECY

# get index of predicted fruit for each image in train set fruit\_classification = [np.argmax(model.predict(np.expand\_dim s(tensor, axis=0))) for tensor in train tensors]

# report train accuracy

train\_accuracy = 100\*np.sum(np.array(fruit\_classification)==n
p.argmax(train\_targets, axis=1))/len(fruit\_classification)
print('Train accuracy: %.4f%%' % train accuracy)

# get index of predicted fruitfor each image in test set

fruit\_classification\_test = [np.argmax(model.predict(np.expan
d\_dims(tensor, axis=0))) for tensor in test\_tensors]

# report test accuracy

test\_accuracy = 100\*np.sum(np.array(fruit\_classification\_test
) == np.argmax(test\_targets, axis=1))/len(fruit\_classification\_
test)

print('Test accuracy: %.4f%%' % test\_accuracy)

## APPENDIX 8 CONFUSION MATRIX

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import metrics
from sklearn.metrics import confusion matrix , plot confusion
matrix
import seaborn as sns
import matplotlib.pyplot as plt
cm = metrics.confusion matrix(np.array(fruit classification t
est), np.argmax(test targets, axis=1))
ax= plt.subplot()
sns.heatmap(cm, annot=True); #annot=True to annotate cells
ax.set xlabel('Predicted labels');ax.set ylabel('True labels'
);
ax.set title('Confusion Matrix');
pcm = np.array(fruit classification test)
true pos = np.diag(pcm)
false pos = np.sum(pcm, axis=0) - true pos
false neg = np.sum(pcm, axis=0) - true pos
precision = np.sum(true pos / (true pos + false pos))
recall = np.sum(true pos / (true pos + false neg))
```

print('%.4f'%precision)

print('%.4f'%recall)

## APPENDIX 9 PREDICTION

#loading the best model

model.load\_weights('/content/drive/My Drive/saved\_models/weig hts.best.from\_scratch.hdf5')

```
def test_model(img_path):
    img=cv2.imread(img_path)
    plt.imshow(img)
    plt.show()
    bottleneck=path_to_tensor(img_path).astype('float32')/255
    predicted_vector=model.predict(bottleneck)
    return fruit names[np.argmax(predicted vector)]
```

```
#for single image
import time
start = time.time()
testimagepath="/content/drive/My Drive/test-
images/banan1.png"
```

```
detected=test_model(testimagepath)
print('Detected Class: ',detected.split('/')[-1])
print(f'Time:{time.time() - start}')
```

#for multiple images
import time

```
start = time.time()
folderpath="/content/drive/My Drive/test-images"
testimages=glob(folderpath+"/*")
for timage in testimages:
    print("Input Image: ",timage)
    detected_class=test_model(timage)
    print("Detected Class: ",detected_class.split('/')[-
1],"\n")
```

print(f'Time:{time.time() - start}')

#### **APPENDIX 10**

### **Similarity Report**

Chapters	Percentages
Abstract.doc/docx	0%
Chapter 1.doc/docx	0%
Chapter 2.doc/docx	3%
Chapter 3.doc/docx	12%
Chapter 4.doc/docx	13%
Chapter 5.doc/docx	1%
Chapter 6.doc/docx	3%
Chapter 7.doc/docx	0%
*All.doc/docx	8%

\*All.doc/docx document must include all your thesis chapters (except cover page, table of contents, acknowledge, declaration, references, appendix, list of figures, list of tables, and abbreviations list ).

$\leftarrow \rightarrow$	O A https://www.turnitin.com	v/t_inbox.asp?r=97.9723168899859&svr=34⟨=en_us8	kaid=85058509&fo	=0&pg=1&ro=4				六 佐 @ @ …
	Stella A	нланкероп	36%			u	1343501266	14-JUN-2020
	Ziad A	abstract	0%			0	1334352313	29-May-2020
	Ziad A	Chapter 1	1%			0	1334029753	29-May-2020
	Ziad A	chapter 2	3%			0	1334030202	29-May-2020
	Ziad A	Chapter 3	7%			0	1334030611	29-May-2020
	Ziad A	Chapter 4	10%			0	1334352642	29-May-2020
	Ziad A	Chapter 5	8%			٥	1334031082	29-May-2020
	Ziad A	Chapter 6	0%			٥	1334031305	29-May-2020
	Ziad A	ICAFS2020	14%			0	1351291577	29-Jun-2020
	Ziad A	Overall Thesis Final	6%			0	1334353155	29-May-2020
	Ziad A	Thesis	11%			0	1326888278	18-May-2020
	Barakat D	FONES	20%			0	1350012379	26-Jun-2020
	Cem D	ICAFS2020	15%	-	-	0	1350740333	28-Jun-2020
	Manal D	abstract	0% 🔳	-		0	1344735615	16-Jun-2020
	Manal D	Chapter 1	0%	-		0	1341828688	11-Jun-2020
	Manal D	Chapter 2	3%	-		0	1341830506	11-Jun-2020
	Manal D	Chapter 3	12%			0	1341830676	11-Jun-2020
•	Manal D	Chapter 4	13%	-		0	1344734612	16-Jun-2020
	Manal D	Chapter 5	1%	-	-	0	1344734836	16-Jun-2020
	Manal D	Chapter 6	3%	-	-	0	1344735100	16-Jun-2020
	Manal D	Chapter 7	0%	-		0	1344735263	16-Jun-2020
	Manal D	Overall Thesis	8%	-		0	1352218137	01-Jul-2020
								PAGE: 1 2 3

Final Thesis (2) (1).docx     Open file     Final Thesis 4 (1).docx     Open file					Show
Type here to search	H 🟦 💽	0 💼 💻	🚾 🔲 🛷 4	) ~ 葉顎 (mole) = NG	5:09 PN 7/1/202

Regards,



#### **APPENDIX 11**

#### **Ethics Approval**

#### ETHICAL APROVAL DOCUMENT

Date: \_30\_\_/\_6\_\_/\_2020\_\_\_\_

#### To the Graduate School of Applied Sciences

The research project titled "Fruit Classification using Convolutional Neural Networks......" has been evaluated. Since the researcher(s) will not collect primary data from humans, animals, plants or earth, this project does not need to go through the ethics committee.

**Title: Assoc Prof Dr** 

Name Surname: Melike Sah Direkoglu

Signature:

Role in the Research Project: Supervisor