

2018

## Spoilage Detection in Raspberry Fruit Based on Spectral Imaging Using Convolutional Neural Networks

Karthik Kuchangi Jothi Prakash

*Technological University Dublin, Ireland*

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomdis>

 Part of the [Computer Engineering Commons](#)

---

### Recommended Citation

Prakash, K.K.J. (2018). Spoilage Detection in Raspberry Fruit Based on Spectral Imaging Using Convolutional Neural Networks. *Dissertation M.Sc. in Computing (Data Analytics)*, August 2018.

This Conference Paper is brought to you for free and open access by the School of Computer Science at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [aisling.coyne@tudublin.ie](mailto:aisling.coyne@tudublin.ie), [vera.kilshaw@tudublin.ie](mailto:vera.kilshaw@tudublin.ie).

**Spoilage detection in raspberry fruit based  
on Spectral imaging  
using  
Convolutional neural networks**



**Karthik Kuchangi Jothi Prakash**

*D17123395*

A dissertation submitted in the partial fulfillment of the requirements of  
Dublin Institute of Technology for the degree of  
M.Sc. in Computing (Data Analytics)

August 2018

## **Declaration**

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Karthik Kuchangi JothiPrakash

Date: 04/09/2018

## Abstract

Effective spoilage detection of perishable food items like fruits and vegetables is essential for retailers who stock and sell large quantities of these items. This research is aimed at developing a non-destructive, rapid and accurate method which is based on Spectral Imaging (SI) used in tandem with Convolutional Neural Network (CNN) to predict whether the fruit is fresh or rotten. The study also aims to determine the number of days before which the fruit rots. This research employs a primary, quantitative and inductive methods to investigate the Deep Learning based approach to detect fruit spoilage.

Raspberry fruit in particular has been chosen for the experiment. Baskets of raspberries from three different stores were bought and stored in the refrigerator at four-degree Celsius. Images of these baskets was captured on a daily basis using an RGB digital camera until all the baskets of fruits were rotten. The study employs a Supervised learning-based classification approach where-by the data is labelled based on the physical appearance of fruits in the basket. The results show that a Spectral imaging technique used along with a CNN yields a good accuracy of 86% with the F1 score of 0.82 to classify the fruits as Good or Bad but does not fare well in estimating the number of days before the fruit actually rots. The ability of CNN to process and identify patterns in a SI to detect spoilage in fruits would help fruit retail operators to optimize their business chain.

Keywords – *Convolutional Neural Network, Image classification, Image processing, Pattern recognition, Spectral Imaging, Spoilage detection*

## Acknowledgements

Firstly, I would like to thank my supervisor **Dr. Robert John Ross** for his tremendous support and motivation during the course of this research. His valuable suggestions and guidance at all times kept me motivated and on track to complete the project.

I would like to thank all the lectures from the School of Computing in Dublin Institute of Technology who have imparted knowledge, been kind and helpful throughout my time in college.

Finally, I would like to acknowledge the love and support that my family showered during my studies. A big hug and thanks to my father **K. Jothi Prakash**, my mother **B.S. Aruna** and my loving wife **Apurva**. This would not have been possible without them.

## Table of Contents

Declaration .....	i
Abstract .....	ii
Acknowledgements .....	iii
List of Figures .....	vi
List of Tables .....	vii
List of Acronyms .....	viii
Chapter 1 .....	1
Introduction .....	1
1.1 Background .....	1
1.2 Research Problem .....	2
1.3 Research Objectives .....	3
1.4 Research Methodologies .....	4
1.5 Scope and Limitations .....	4
1.6 Document Outline .....	5
Chapter 2 .....	7
Literature Review .....	7
2.1 Detecting fungal infections in fruits and other food products .....	7
2.2 Use of Hyperspectral images to analyze food products .....	9
2.3 Applications of Deep Learning techniques in Agriculture .....	11
2.4 Efficiency of Convolutional Neural Network in Image processing tasks .....	13
2.5 Summary of the literature, limitations and gaps .....	14
Chapter 3 .....	16
Design and Methodology .....	16
3.1 Problem Definition .....	16
3.2 Data collection .....	16
3.3 Data processing and profiling .....	17
3.4 Modeling .....	18
3.4.1 Convolutional Layer .....	19
3.4.2 Pooling Layer .....	20
3.4.3 Fully-Connected Layer .....	20
3.5 Evaluation .....	20

Chapter 4.....	23
Implementation and results .....	23
4.1 Data collection .....	23
4.2 Image segmentation .....	26
4.2.1 Increasing the size of the data .....	27
4.2.2 Highlighting the region of interest and ease of data labelling .....	28
4.3 Data profiling .....	30
4.3.1 Irregular image sizes .....	30
4.3.2 Minimum fruit coverage in the image .....	30
4.3.3 Identifying and removing pixelated images.....	32
4.4 Data labelling .....	32
4.4.1 Labels to identify good vs bad fruits.....	33
4.4.2 Labels to identify the numbers of days left before the fruit rots.....	35
4.5 Dataset creation.....	36
4.6 Importing data using TFRecords .....	36
4.7 Modelling.....	37
4.8 Predicting the current state of the fruit .....	39
4.9 Estimating the number of days before the fruit rots .....	40
4.10 Summary .....	42
Chapter 5 .....	43
Evaluation and Analysis .....	43
5.1 CNN architecture Evaluation .....	43
5.2 Evaluating the model to predict the current condition of the fruit.....	44
5.3 Evaluating the model to estimate the number of days left before rotting .....	45
5.4 Strengths and limitations of the result .....	46
Chapter 6 .....	47
Conclusion .....	47
6.1 Problem Definition.....	47
6.2 Design, experimentation, evaluation and results .....	47
6.3 Contributions and Impact.....	48
6.5 Future work and recommendations.....	49
References .....	50
Appendix.....	54

## List of Figures

2.1	Layout of the literature review.....	7
3.1	High level design of the research.....	17
3.2	Components of a Convolutional Neural Network.....	19
4.1	Image of a raspberry basket on day one of the experiment.....	25
4.2	Image of a raspberry basket on day twelve of the experiment.....	26
4.3	The segmented images of a raspberry basket.....	27
4.4	Original Image of the raspberry containing a decay which is not highlighted.....	29
4.5	A segmented image of the original image where the decay region is highlighted..	29
4.6	A segmented image whose background information is greater than threshold.....	31
4.7	A segmented image whose background information is on the border of the threshold and hence considered for the experiment.....	31
4.8	A segmented image that is pixelated or blurred.....	32
4.9	A set of Images that were labelled as good.....	33
4.10	Another set of images labelled as good.....	34
4.11	A set of Images labelled as bad.....	34
4.12	Another set of images labelled as bad.....	35
4.13	The graph of the training accuracy of the model.....	40
4.14	A scatter plot showing the actual and predicted values from the model.....	41
5.1	A boxplot based on the comparison of the accuracies of the model for different epochs.....	44



## List of Tables

3.1	Confusion Matrix.....	22
4.1	Specification of the Convolutional layers used in the experiment.....	38
4.2	The maximum, minimum and average Accuracy and F1score of the model to predict current condition of the fruit.....	40
4.3	Accuracy and F1scores of the model to estimate number of days.....	41

## List of Acronyms

<b>ANN</b>	Artificial Neural Network
<b>BNN</b>	Back Propagation Neural Network
<b>CNN</b>	Convolutional Neural Network
<b>DL</b>	Deep Learning
<b>FN</b>	False Negative
<b>FP</b>	False Positive
<b>HIS</b>	Hyperspectral Images
<b>IR</b>	Infrared
<b>ML</b>	Machine Learning
<b>NIRS</b>	Near Infrared Spectroscopy
<b>PLS</b>	Partial least squares
<b>SI</b>	Spectral Imaging
<b>SVM</b>	Support Vector Machines
<b>TN</b>	True Negative
<b>TP</b>	True Positive
<b>UV</b>	Ultraviolet
<b>VIS/NIR</b>	Visible/Near infrared

# **Chapter 1**

## **Introduction**

This section introduces to the background scope of the study, followed by the research question, the hypothesis, the research objectives and the research methodologies that are used in this study. An overview of the thesis along with its contents is also presented.

### **1.1 Background**

Consumption of fresh fruits and vegetables provide a lot of nutritional benefits to humans in the form of proteins, vitamins, calcium, and potassium to name a few. Cultivation of fruits is increasing in many countries due to a steady increase in consumption of fresh and processed fruits. As a result of this, many European countries and the United States cultivate a large quantity of Raspberries (Giovanelli, Limbo & Buratti, 2014) which contribute significantly to the economy.

Red raspberries are considered as one of the most popular berries among the customers for their delicious sweet taste and also for their rich content of health beneficial nutrients. (Rao & Snyder, 2010). Many studies have shown that raspberries act as a major source of antioxidants because of their high concentration of polyphenolic compounds like flavonols, anthocyanins, catechins, ellagic acid derivatives and ascorbic acid (Beekwilder et al., 2005).

The shelf life of red raspberries is limited to a few days postharvest. They lose their firmness, and the susceptibility to rot is very high if it is not preserved properly. Structurally, a raspberry fruit is very fragile when compared to other fruits like Apple or Mango. Retailers that sell the raspberries must inspect the baskets in which they are stored before they put it up on the shelf. This requires a lot of human effort and time since they have to inspect a large number of baskets on a daily basis to make sure that the fruit is in a healthy condition.

Computer vision or Image processing-based systems find many applications in the food and agricultural industries. They are used primarily in the domain of classification of

products and also quality control (Moallen, Serajoddin & Pourghassem, 2017). Computer vision has replaced visual inspection in various agricultural commodity grading systems because of their accuracy, speed and also, they are economical (Donis-Gonzalez et al., 2012). It is also a non-destructive method as it limits the physical contact of the food with the handler while inspection which may otherwise lead to physical distortion of the food. It captures the external aspects of the agricultural product like the color, shape, and size.

In recent years, Deep learning (DL) techniques are growing in popularity in the agricultural domain and also in the field of image processing (Kamilaris and Prenafeta-Boldu, 2018). DL is used in the field of agriculture to analyze agricultural data, understand the problems associated and provide solutions to the issues which range from soil fertility, fungal infection in food products and so on. Deep learning techniques allow the implementation of CNN, which are used in many image processing tasks across various domains like healthcare and also agriculture.

Traditionally in the field of agriculture, computer vision techniques have been implemented with machine learning models like Support Vector Machine (SVM) (Zhang Chu et al, 2016) to classify the ripeness of the fruits. To increase the accuracy of classification, Deep learning techniques using CNN are used. CNN used along with Spectral/Hyperspectral images has the potential to efficiently transform the agricultural model, thus reducing human effort.

## **1.2 Research Problem**

The project aims to train a Deep learning model using Convolutional Neural Network and to determine the condition of the raspberry fruit. The dataset consists of the images obtained from raspberry fruits which were collected over a time span of twelve days using a digital camera. This is a supervised learning technique which has independent and dependent variables. The three channels of the image namely Red, Green, and Blue (RGB) which is made up of pixels are the independent features with the dependent feature being the condition of the fruit (either good or bad). The research question that is answered from this study can be stated as:

*“Can Convolutional Neural Networks be able to detect the spoilage in a raspberry fruit effectively using Spectral Imaging?”*

The result obtained from this study would be a proof of concept for the use of Convolutional neural networks along with spectral imaging in the detection of fruit spoilage.

### **1.3 Research Objectives**

The research objective of this study can be framed from the hypothesis:

**H<sub>0</sub>:** Convolutional Neural network cannot detect the patterns from a spectral image accurately, to predict the current state of a raspberry fruit.

**H<sub>1</sub>:** Convolutional Neural network can detect the patterns from a spectral image accurately, to predict the current state of a raspberry fruit.

**H<sub>0</sub>:** Convolutional Neural networks cannot detect the patterns from a spectral image accurately, to estimate the number of days left before the raspberry’s rot.

**H<sub>1</sub>:** Convolutional Neural networks can detect the patterns from a spectral image accurately, to estimate the number of days left before the raspberry’s rot.

The overriding objective is to design and execute experiments that seek to reject the null hypothesis. The experiment is performed as follows:

- Investigate and document the state of art in detecting spoilage of food items and also the current application of Spectral imaging and Deep learning techniques in the field of agriculture.
- Create the dataset by capturing images from baskets of raspberries bought from different stores and labelling these images by the virtue of visual perception.
- Perform data profiling on the dataset.
- Built a Deep learning-based model to detect the spoilage in a raspberry fruit using Convolutional Neural Network.
- Evaluate the model for its ability to detect the spoilage using F1 scores and the classification accuracy.

## **1.4 Research Methodologies**

The study is aimed at predicting the condition of a raspberry fruit which involves collecting data in the form of images and hence classifies as a primary research. Images from baskets of raspberries are captured using an RGB camera. As a part of the primary research, literature related to the existing techniques to detect spoilage in food products, use of hyperspectral images in agriculture and image processing using convolutional neural networks are studied to make up a comprehensive literature review.

The research is a quantitative and empirical research. The performance of the CNN classifier will be evaluated based on the accuracy and F1 score. On the basis of these results, a conclusion will be drawn as to whether or not a CNN model can predict the condition of a raspberry fruit and hence falls under inductive research.

## **1.5 Scope and Limitations**

The study will focus on classifying the raspberry fruit as either fresh or rotten, and also estimating the numbers of days left for the raspberry to rot. It is achieved by implementing Deep learning techniques on the images of raspberry fruits. The dataset that is used in the experiment was prepared by capturing the images of raspberry fruits using an RGB camera and labelling the images based on visual inspection. A convolutional neural network is used to build a model to detect the patterns from the image and predict the condition of the fruit. The model will be trained and evaluated against the F1 scores and the percentage of its classification accuracy.

Raspberries bought from different stores are used in the experiment. Data is generalized and is not restricted to any one particular shop. Images are captured under varied lighting conditions so that the brightness of the image does not alter the model's performance.

The camera used to capture the image is a normal digital camera which produces an RGB image. Unlike a multispectral camera which captures the external and internal aspects of the fruit, a digital camera captures only the exterior aspects of the fruit like its color, size or shape. So, the model is trained to predict the condition of the fruit-based only on its exteriors

and does not consider the internal aspects of the fruit. Secondly, the labels are assigned to the images based on the visual perception of the condition of the fruit as observed by the researcher.

The fruits were stored in a refrigerator throughout the experiment and therefore the effects of the environmental factors are not taken into consideration. Furthermore, a relatively small sample size of 1128 images are available to carry out the experiment. There is an imbalance in the dataset which has greater number of images labelled as Good as against Bad.

## **1.6 Document Outline**

The remaining chapters of the thesis shall be organized in a manner as described below:

- Chapter 2 (Literature Review) will provide an overview of the current state of the art techniques used to detect spoilage in food products. A systematic literature review is conducted to gather insights into the use of Hyperspectral images, and also the Deep learning techniques in the field of Agriculture. The use of Convolutional Neural Networks in many image processing, pattern recognition studies is also reviewed. The gaps existing in the literature are highlighted in order to support the research question for this study.
- Chapter 3 (Design and Methodology) will provide details of the design of this experiment. The steps that are involved to carry out the study are mentioned. Details about the data collection process and the techniques that are proposed to be implemented for this research is listed and described.
- Chapter 4 (Implementation and results) details the implementation of each step carried out in the experiment in depth. The results of the experiment are also listed out in this chapter.
- Chapter 5 (Evaluation and Analysis) will apply the criteria of evaluation that is elaborated in Chapter 3 against the results that are generated in Chapter 4. It will then state as to whether or not to reject or accept the hypothesis.

- Chapter 6 (Conclusion) discusses the contribution that is made to the existing knowledge body by giving a synopsis of the results that are obtained from the experiment. The chapter also outlines the future work that may be undertaken in this regard.



## Chapter 2

### Literature Review

This chapter will cover literature work regarding the existing techniques in spoilage detection, hyperspectral images and their use in food products, advances of Deep learning techniques in agriculture and the use of Convolutional Neural networks in tasks relating to image processing. The chapter is divided into different sections and the layout of the sections is as shown in figure 2.1.

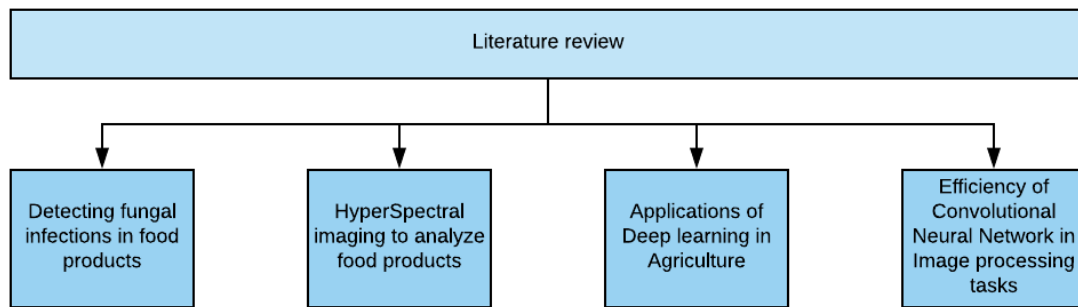


Figure 2.1: Layout of the Literature review

#### 2.1 Detecting fungal infections in fruits and other food products

Detecting rotten and fungus infected fruit at an early stage is essential for producers, retailers and consumers. Fungal contamination can spread from a single fruit to a whole basket of fruits in transportation or storage. Presence of fungus on the fruit surface causes loss of quality and also diminishes the safety of the consumer.

There are many methods to detect spoilage or fungal infections in food products. Traditional methods include using microbiological or physicochemical techniques but require a lot of labor, professional experience, greater cost, time consuming and is limited to test a restricted number of fruit batches at a time (Sanzani et al., 2016). With a mass production of fruits, there has been a number of researches carried out to develop non-

destructive, rapid and real time monitoring of different parameters of fruits that affect their quality. (Brosnan & Sun, 2004)

Detecting the spoilage in a fruit can be carried out in two ways. One way is to examine the internal quality of the fruit and the other is to examine the external quality. Gomez et al., (2006) used a non-destructive optical method which was based on visible/near-infrared (Vis/NIR) spectroscopy to evaluate the internal quality of the fruit like starch content, oil content and water content. Evidence was found that the internal qualities of the fruit had an effect on the spectral measurements thus proving that Vis/NIR technology could be used to detect the freshness of the fruit.

There are many ways to detect the external quality of the fruit. One of the traditional methods to detect the external spoilage in a fruit involves the use of Ultraviolet (UV) rays. Fruits are inspected manually under the UV rays to detect damage to the skin. In the case of orange fruits, the damaged skin of the oranges is quite often fluoresced by UV rays (Kondo et al., 2009). Use of UV rays is potentially harmful for humans as long exposure to it can cause damage to the skin leading to premature aging or cancer.

Using a soft X-ray imaging on wheat grains and processing this image in various statistical discriminant classifiers like linear classifier and quadratic classifiers, Narvankar et al. (2009) was able to achieve an accuracy of 90% in classifying wheat grain infected with fungus as against those which were not. The study showed that wheat grains that were infected with fungus had a higher density in comparison with healthy grains. This density change could be identified by comparing the features that were extracted from the X-ray images of the infected and healthy wheat grains. The study concluded that soft X-ray imaging used along with statistical classifiers outperformed the Artificial Neural Network (ANN) classifiers. Use of X-rays involves relatively high cost and also care needs to be taken to shield from the radiations produced by the X-rays.

Thermal images are usually known to be used in identifying temperature variations of the objects in an image. Thermal imaging technique has also been used to detect fungal infections in food products. Chelladurai et al. (2010) captured thermal images of wheat grains that were infected with different types of fungus. They varied heating and cooling rates while capturing the thermal images to differentiate the healthy and fungal infected

tissues of wheat grains based on their chemical composition. They obtained a classification accuracy of 96% in identifying fungal infected wheat grains. However, the study could not differentiate between various fungal species as the temperature profiles of the samples that was infected by various fungal species was the same.

## **2.2 Use of Hyperspectral images to analyze food products**

Hyperspectral images (HSI) provide more detailed information than a normal color image as it acquires extra spectral channels that may be Ultraviolet (UV) or Infrared (IR) in addition to the visual primary colors Red, Green and Blue. As a result, each pixel in the HSI contains more spectral information which can be used to characterize the objects with great detail and precision. An HSI is capable of capturing information that is beyond the visible range of a human eye. As a result, HSI is used in a wide range of applications ranging from satellite based remote sensing, military operations, healthcare and also agriculture. Due to the rich information content provided by the HSI, they suit well for image processing tasks.

Conventional methods like human inspection which are used to detect the firmness and other attributes in a fruit are considered as destructive methods. Verifying individual fruit is difficult and requires sampling of fruits. Use of Near-Infrared Spectroscopy (NIRS) technique provides a non-destructive and accurate measure to verify individual fruits for their external qualities like the firmness and also internal qualities like the Ph, acidity and sugar content in commercial operations (Rungpichayapichet, P., Mahayothee, B., Nagle, M., Khuwijitjaru, P. & Müller, J., 2016). The study used NIRS to examine the quality of mangoes postharvest using reflectance spectra which were in the region of 700-1100nm. Calibration models were developed using partial least squares (PLS) regression analysis to detect firmness, total soluble solids ripening index in mangoes. It was found that the model depended on the harvest year of the mangoes and using mangoes from two to three different harvest years increased the prediction accuracy of the model. The study concluded that NIRS could be used as a reliable and non-destructive technique to assess the quality of mangoes.

Bruising of vegetables and fruits is a common type of damage that mainly occurs due to collision or compression between the products or during packing and transportation.

Many times, a bruised fruit or vegetable is vulnerable to fungal and bacterial infections which may result in rotting and thereby affecting the quality of other fruits stored along with it (Dintwa, E., Van Zeebroeck, M., Ramon, H. & Tijssens, E., 2008). Identifying the bruised product is important and necessary to improve the overall quality and also to reduce the cross infection amongst them. But detecting and identifying the bruises is a difficult task. Ye, D et al. (2018) proposed a hyperspectral image-based technique to detect the bruises in potatoes. The study divided the potato samples into three different levels based on the degree of the bruises on them. The hyperspectral image-based data was pre-processed and the grid search algorithm was applied to the data to optimize the modeling parameters. The model was not only successful in detecting the minor bruises on the potatoes but also could classify the bruise levels with the help of hyperspectral images, thereby providing evidence that hyperspectral images indeed can be used to extract minute information from the data.

Fungal infections in fruits, is difficult to be spotted by human eye in their early stages. Strawberries, like raspberries is a popular fruit which is mainly attributed to their nutritional values and the sweet taste. However, the shelf life of a strawberry is short due to their high metabolic rate at the time of storage. This attribute of the strawberries makes them susceptible to infections and tissue damage by different fungi and bacteria. Hence detecting the fungal infected fruit early is important. Hyperspectral imaging technique was applied on the strawberries which included the fresh and also fungal infected samples. 19 wavelengths were identified from the original spectra as most appropriate for distinguishing between the samples. A Backpropagation Neural Network (BNN) model provided the most accurate results. The study was also able to detect the fungal infections 24 hours prior to which the symptoms of decay was visible to human eye (Siedliska, A. et al, 2018).

Along with fruits and vegetables, people all around the globe enjoy meat products. Meat has been an important aspect of human diet for many centuries for their nutritional benefits. People prefer different kinds of meat and with greater access to information in the recent times, consumers are concerned and want to be aware of the kind of meat they consume. Therefore, it is important to label the meat accurately thereby helping the customers select the appropriate type of meat that suits them well. Kamruzzaman, M. et al. (2012) used multivariate analysis in tandem with hyperspectral imaging technique to identify different

types of red meat. Hyperspectral images were captured from the longissimus dorsi muscles of beef, lamb and pork. The spectral data from these images were extracted and Principal component analysis (PCA) was used to analyze the data. Six wavelengths were identified by applying the second derivate and these wavelengths were used as inputs to pattern recognition algorithms that classified the meat samples with Partial least square discriminant analysis yielded a classification accuracy of 98.67% on the meat samples. The study concluded that use of hyperspectral images with multivariate analysis and image processing had a good potential to identify and label the red meat.

### **2.3 Applications of Deep Learning techniques in Agriculture**

Deep learning (DL) is an extension of Machine learning (ML) in terms of adding depth or increased complexity into a model. It transforms the data using different functions that allows data to be represented in a hierarchical way through multiple levels of abstraction (Schmidhuber, J., 2015). DL is advantageous for its feature learning abilities. It extracts features from the raw data automatically in such a way that features of higher levels of hierarchy are formed from the composition of the lower level features. DL models can solve complex problems well and fast which allows massive parallelization (Pan & Yang, 2010). The complex models employed in DL has the capability to increase the classification accuracy in a classification problem or reduce the error in regression problems, provided they are trained on adequately large datasets.

With these advantages, DL is used in many domains to solve complex problems. They can be applied on data in various forms like speech, audio, images and also natural language. Hence, they are used in the fields of healthcare, creating chatbots and so on. DL techniques have also been used successfully to solve many agricultural problems.

For a fruit to grow healthy in a plant, certain portion of the flowers and fruitlets must be removed in an early stage in the growing season. The proportion of flowers and fruitlets that has to be removed from the plant is estimated on the basis of the bloom intensity, i.e the number of flowers that are present in the orchid. Dias, P.A., et al., (2018) proposed an automated technique of estimating the bloom intensity using a Convolutional Neural

Network based DL technique. Images of apple tree were used to train a CNN to detect flowers in the tree. The model was robust enough to changes in illumination of images in the dataset and could achieve a precision and recall of greater than 90% on a challenging test dataset. They concluded that the proposed DL based approach was better than many baseline methods in terms of the generalization capabilities.

There are many pieces of literature which show the successful use of DL in many pattern recognition, image processing and classification tasks. Pattern recognition has been used in vegetable pathology along with the use of ML algorithms to detect patterns in the image and predict accurate results for the related problem. One such study which includes the use of DL in vegetable pathology was undertaken by Lu, Yi, Zeng, Liu and Zhang (2017) on detecting the diseases that affects a rice plant. For this purpose, images from healthy and diseased rice leaves and stem was captured. They were fed into a CNN to train a model. Different hyperparameters were tried and tested in the CNN to obtain optimal prediction accuracy. The model was able to successfully classify ten rice diseases associated with the rice plants. It was found that CNN models can be effectively used to identify rice diseases through image recognition as it had a better training performance and also the convergence rate was much quicker compared to other models.

The yield from any agricultural product, mainly depends on keeping out the weeds from infecting the plants. Different weed control techniques like the use of chemicals or mechanical weeding is applied to limit the damage of weeds on the plants. For effective weed control, it is important to know the kind of weed species that are present in an agricultural field and treat them accordingly. Computer vision techniques have been used to identify the weed species and has been successful (Søgaard, H., 2005). Using computer vision itself, only a limited number of plant species can be identified. Using Deep learning techniques on color images from plant species, a greater variety of plant species can be identified correctly which in turn helps in effective weed management and a good crop yield (Dyrmann, M., Karstoft, H., & Midtiby, H., 2016).

## **2.4 Efficiency of Convolutional Neural Network in Image processing tasks**

A convolutional neural network is one of the well-known deep learning architectures. Several works of literature have proposed a CNN model as the suitable framework for many biological vision problems. A CNN takes inspiration from the mechanism of natural vision perception found in living creatures. Its ability to obtain representations of the original image effectively helps in recognizing the visual patterns from the raw pixels of the image directly with minimal or no preprocessing (Gu, J., et al., 2018).

A large number of variations with respect to the architecture of the CNN can be found in the literature. It mainly varies with respect to the depth of the network, different activation functions used, different loss functions, different techniques for regularization and optimization. However, the components that are used in all the architectures is similar and it comprises of Convolutional layers, Pooling layers and fully connected layers.

In a CNN, multiple convolutions are performed on the images which creates numerous representations of the dataset that is used for learning. It starts with a general representation at the initial layers and becomes specific as the network grows deeper. The convolutional layer and the pooling layer perform the task of feature extraction while the fully connected layer performs the task of classification. The input and output of a CNN is a vector (Kamilaris, A., & Prenafeta-Boldú, F., 2018).

CNN's have a wide application in the field of Earth imagery through analysis of images that are acquired remotely. Processing hyperspectral images and multispectral images lead to a large dimension of information resulting in an increased processing time. A processing time that is acceptable and also getting a good classification accuracy in sensory image classification task depends on the selection of parameters in a CNN (Maggiori, Tarabalka, Charpiat & Alliez, 2017).

Due to the proven success of CNN in image processing tasks, they have been widely used in various domains like Cancer detection (Yang, X., Liu, C., Wang, Z., Yang, J., Min, H., Wang, L., & Cheng, K., 2017), football count and to estimate the density (Sindagi, V., & Patel, V., 2018), for detecting online food materials (Chen, H., Xu, J., Xiao, G., Wu, Q., &

Zhang, S., 2018), in the field of agriculture (Sa et al., 2016) and also in livestock management (Santoni, M., Sensuse, D., Arymurthy, A., & Fanany, M., 2015) to name a few.

## **2.5 Summary of the literature, limitations and gaps**

This chapter provided an extensive review of the various methods that are used to detect spoilage in fruits, use of Hyperspectral images in detecting spoilage, advances of Deep learning in agriculture and also the use of CNN's in Image processing tasks. The process of detecting spoilage can be broadly classified into two groups. One is destructive and the other is non-destructive. Destructive techniques involve human interference in the process of inspection of products. These methods can be laborious, time consuming and also prone to errors. In case of inspecting structurally fragile products like fruits, human handling can cause distortion on the surface of the fruits due to continuous handling.

Non-destructive techniques that make use of UV rays, soft X-ray and thermal imaging are used that does not involve handling the products manually. These techniques are more reliable than human inspection. However continuous exposure to these rays can have an effect on the skin of humans involved in this process. Spoilage in food products can be found based on the external aspects or internal aspects of the food. Hyperspectral imaging can be used to detect the internal quality of the food products.

Hyperspectral images consist of information that is not visible through human eye. These images consist of multiple wavelengths that cannot be detected by human eye thus helpful in inspecting the internal qualities of the fruit. Use of a hyperspectral camera can be expensive depending on the output wavelengths that are obtained from the hyperspectral images.

Deep learning techniques are growing in popularity in the field of agriculture based on the literatures reviewed. Due to their ability to classify images precisely, they have been used to solve many agricultural problems which involves capturing images. A hierarchical structure that is followed in a Deep learning technique enables the features to be extracted precisely thus increasing the accuracy of the prediction models.



In particular, deep learning architecture based on Convolutional Neural networks have been proven to have a good track record in image processing tasks. Limited research has been carried out on using a non-destructive technique like CNN using the spectral images to detect spoilage in a raspberry fruit. Given the popularity, short life span and a fragile structure of raspberry's a non-destructive, efficient and less laborious technique to detect a rotten raspberry would be beneficial if researched further.

## **Chapter 3**

### **Design and Methodology**

An overview of the experimental design which is undertaken in this study to answer the research question is explained in this chapter. The experiment is carried out in phases starting from Understanding the problem, collecting data, cleaning the data, Modeling and then evaluation. A digital camera will be used to collect data in the form of images. Python programming language along with the Tensorflow library will be used to build the model.

The key focus of the research is to build a model that predicts the condition of a raspberry based on the images collected from baskets of raspberry fruits over a period of time. An overview of the experiment is illustrated in figure 3.1.

#### **3.1 Problem Definition**

The main aim of this study is to predict the condition of raspberry fruits which have a short life span. To achieve this, a Deep learning based Convolutional neural network model will be trained on the images, collected from a digital camera and labelled according to the researcher's visual perspective of the image. The performance of the Deep learning model will be evaluated based on F1 score and classification accuracy.

#### **3.2 Data collection**

The dataset for this experiment does not exist and has to be created. The dataset will be created by capturing the images of the raspberry fruit. Baskets of raspberry fruits that have the same expiry date will be bought from different retail outlets and stored in a refrigerator at 4 degree Celsius. Images of each raspberry basket will be captured by removing the baskets from the refrigerator and placing them horizontally on a white background. A digital camera with 13 Mega pixel capacity will be used to capture the images.

The image consists of three channels namely Red, Green and Blue. The baskets of raspberry fruit are expected to rot over a span of time which is evident by the growth of fungus or signs of decay or distortion of the outer surface of the fruit. All the information will be

captured in the form of images. Once the images are collected, it will be pre-processed and then labelled to create the dataset.

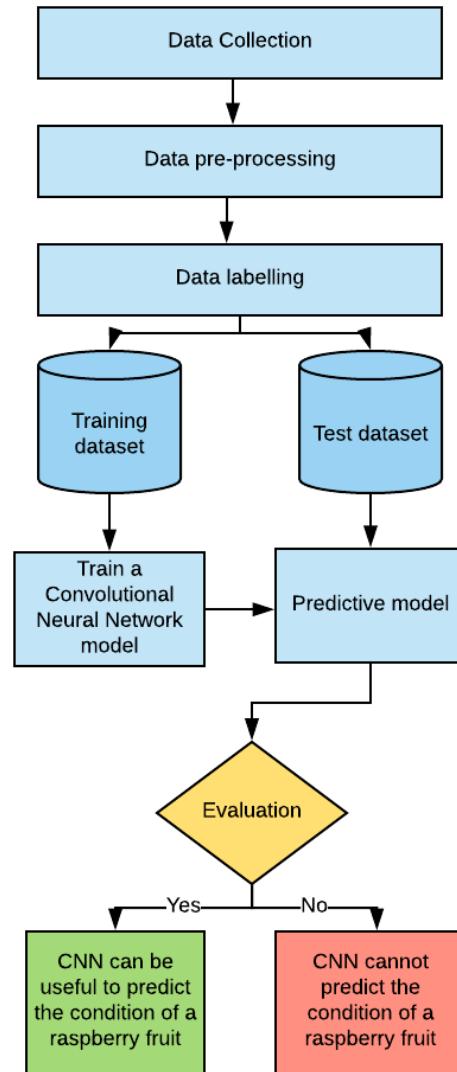


Figure 3.1: A high level design of the research

### 3.3 Data processing and profiling

All the images that are collected from the raspberries are of dimension 4160 x 3120 pixels. The images will be broken down or segmented into smaller dimensions to increase the number of images and also to facilitate the model to learn the patterns from the image easily

whereby the region of interest in the image is highlighted. Python programming language along with OpenCV library will be used to read the images from the system and perform segmentation.

Once the images are segmented, they will be analyzed to check for inconsistent dimensions, background information and also the quality of the pixels in the image. Images that do not meet the above standards will not be used for the study.

After the images are segmented and profiled, labels will be attached to each image as per the researcher's visual interpretation of the quality of the fruits in that image. Signs like the presence of fungus, distortion of the fruit and rotting on the surface of the fruit will be considered while labelling the images as Good or Bad. The images will also be labelled which contains the number of days taken by it to rot.

A dataset will be created in a TFRecord file format which is a simple binary file format that is used to train a model which uses Tensorflow framework. 70% of the data will be used to train the model and the remaining 30% will be used for testing the model's performance.

### **3.4 Modeling**

The research aims to build a model to predict the condition of a raspberry fruit using Deep Learning techniques. A Convolutional Neural Network is used for this purpose. Python programming language and Tensorflow library will be used to implement the model. The results will be generated on a cluster provided by Dublin Institute of Technology which is equipped with two 8-core Intel Xeon with 2.1GHz and Dual Tesla K40 and K80 Graphics card.

CNN's are feed-forward neural networks consisting of various combination of convolutional layers, max pooling layer and fully connected layers. They exploit the spatially local correlation in a way by enforcing a connectivity pattern locally between the neurons of the adjacent layer (Hu, Huang, Wei, Zhang & Li, 2015). The first layer in a CNN is an image processing layer. It consists of pre-defined filters which will be kept constant throughout the training. Additional information like the edges and gradients along with the

raw input image itself is provided to the network using this layer. It extracts contrast and help to improve the recognition rate (Schmidhuber, 2015).

The layers of the Convolutional Neural Network are:

- i. Convolutional Layer
- ii. Pooling Layer
- iii. Fully-Connected layer

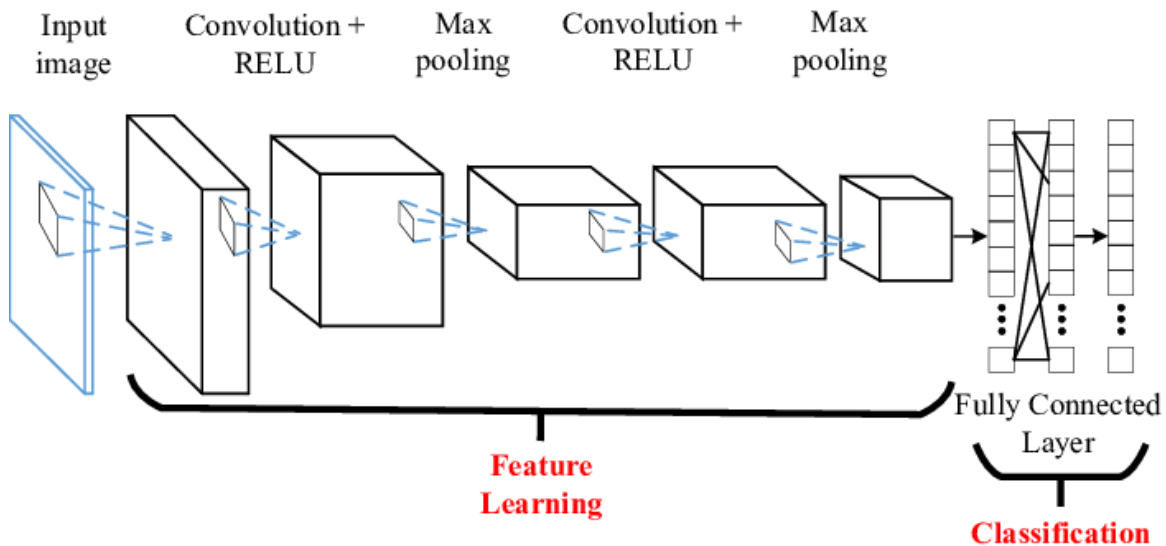


Figure 3.2: Components of a Convolutional Neural Network

(Kamencay, Benco, Mizdos & Radil, 2017)

### 3.4.1 Convolutional Layer

The neurons in the Convolutional layer are sparsely connected to the neurons in the subsequent layer. Clones of the neurons in the Convolutional layer is introduced to make sure that the entire input is covered. This is achieved by the use of Filters.

- **Filters** - Filters is used for simultaneously learning by the use of simple feature detectors that are scanned over the input vector. Each filter uses the same weight and hence can be thought of as clones of each other. The width and height of the filters are set according to our requirements. The output after the filter is applied results in a Feature map.

- **Feature maps** – Typically in a Convolutional layer, many filters are used. Each of these filters returns or outputs an individual feature map.

The output dimensions of a Convolutional layer depend on 3 parameters. The width and the height give the information about the dimensions of the feature map. The depth is the number of filters in the convolutional layer.

The sliding window of the filters over the image can be controlled by the parameter called **stride**. Using this parameter, the overlap between the input image and filter can be varied which results in different sizes of the feature maps. Multiple convolutional layers can be used in a CNN.

### **3.4.2 Pooling Layer**

The feature maps are fed as inputs to the pooling layer. Having numerous filters in the convolutional layer results in a large number of output values which will be used by layers further down in the network. Pooling layer is used to subsample the feature maps in order to reduce their size.

Pooling is done by taking a number of units and returning an aggregate of those units. One method of aggregating the values is by making use of max-pooling. Max-pooling calculates the aggregate by taking the maximum value across the units. It helps in reducing overfitting.

### **3.4.3 Fully-Connected Layer**

The neurons of fully connected layer in a CNN is connected to all the neurons in the prior layer. The convolutional layer detects the local features while the fully connected layer is used to detect the global feature in an image. Activation functions like the softmax are applied in this layer in order to output the probabilities of the predicted classes.

## **3.5 Evaluation**

This experiment uses the F1 score and the accuracy to evaluate the performance of the model. The model is trained on 70% of the data while 30% is used for evaluation. The metrics used to evaluate the model in this classification task are:

- Accuracy or Classification accuracy (CA),
- Precision
- Recall
- F1 score

Accuracy of the model will be calculated as the percentage of correct prediction of the top class (the class having the highest probability as indicated by the CNN model) and the target class assigned by the author before-hand are the same. It can be represented by the below equation.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.1)$$

where,

TP (True Positive) represents the positive instances that are correctly classified as positive,

TN (True Negative) represents the negative instances that are correctly classified as negative,

FP (False Positive) represents the negative instances that are wrongly classified as positive,

FN (False Negative) represents the positive instances that are wrongly classified as negative.

Precision is calculated as the fraction of True Positives (TP) from the sum of the relevant classes, i.e the sum of the True Positives and the False positives. It can be represented by the below equation.

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

Recall is calculated as the fraction of True Positives from the sum of True positives and False Negatives. It can be represented by the below equation.

$$Recall = \frac{TP}{TP + FN}$$

(3.3)

F1 score will be used in this experiment as the dataset is imbalanced. F1 Score is interpreted as the harmonic mean of Precision and Recall. It can be represented by the below equation.

$$F1\ Score = \frac{2 * (TP * FP)}{TP + FP}$$

(3.4)

A confusion matrix is also an evaluation metric that is used to describe the performance of a classification task. Precision, Recall and Accuracy can all be calculated with the help of a Confusion matrix.

	<b>Positive (Actual)</b>	<b>Negative (Actual)</b>
<b>Positive (Predicted)</b>	True Positive (TP)	False Positive (FP)
<b>Negative (Predicted)</b>	False Negative (FN)	True Negative (TN)

Table 3.1: Confusion Matrix



## **Chapter 4**

### **Implementation and results**

This chapter will detail out the steps that are involved in capturing the images, analyzing them, attaching labels to the images and then creating the dataset from these labelled images. A model development and also the testing process for the Convolution Neural Network will be presented.

#### **4.1 Data collection**

The dataset used for this experiment is a primary dataset consisting of a set of images that was created by capturing the images of baskets of raspberries with the use of a digital camera. The dataset has features in the form of an image and corresponding labels to indicate its freshness and also the number of days left before fruit goes off. The image is represented in the form of a matrix of pixels. Since the image is a color image, it is constituted by three channels namely Red, Green and Blue (RGB). Each pixel value is a vector of three numbers. The label which is used to represent the current condition of the fruit is a numeric, discrete variable consisting of values 1 and 0 which represents whether the fruit is fresh or spoilt respectively. Another label which indicates the number of days before which the fruit turns bad is also numeric, discrete variable with values ranging from 0 to 8.

Raspberries used in this experiment are bought from Aldi, Lidl and Tesco. These supermarkets store large quantities of fruits on the shelf. The raspberries were kept in normal room temperature in all the three supermarkets. Two baskets of raspberries are used from each store. A total of six baskets of raspberries that have the same expiry date was collected from the three stores. Using fruits from different vendors for the analysis ensures the data is diverse in terms of their origin, the quality of the fruits and also, the shelving policies of different stores that varies in comparison to each other.

Each of the six baskets were tagged manually with a unique number to ensure that the data is collected in an efficient manner. This helped to track the degradation of each basket of fruit over the duration of the experiment. The two baskets that were bought from Aldi, Tesco and Lidl were labelled as A1, A2, T1, T2, L1 and L2 respectively.

The basket of fruits was kept in the refrigerator once it was bought from the stores. The temperature inside the refrigerator was maintained at four degrees Celsius. The rationale behind keeping the fruits in the refrigerator was to simulate an industry standard environment where fruits are stored in a cooler temperature to increase the lifespan and also negate the effect of the environmental factors on the fruit.

The next phase of the experiment was to capture the images of the fruits that were bought. For this purpose, a 13 Mega pixel camera with ISO 100-1600 was used. The camera outputs a JPEG image of dimension 4160 x 3120 pixels. The fruit baskets were taken out of the refrigerator every day to capture the images and later kept back. Once the fruits were taken out of the refrigerator, each basket was placed on a clear white background one at a time and the image was captured. The camera was held perpendicular to the fruit basket and the distance between the fruit and the camera was approximately the same for every image. A few images were clicked under natural sunlight and a few others under room light to make sure that the samples were collected under different lighting conditions. This ensured that the model was insensitive to the different lighting conditions which could have an effect on the results.

The process of capturing the images was carried out for a duration of twelve days. Some of the baskets of fruits started to show signs of rotting as early as day two of the experiment, while the last basket of fruit rotted on day eight. Even though all the baskets were rotten by day 8, the images of all the baskets of fruits were captured until day twelve. An excel sheet was maintained to record the condition of each basket of fruit from the start till the end of the data collection process. It contained three columns. The first column contained the unique number that was assigned to each basket earlier. The second column contained the day number while the third column contained information related to the condition of the fruit basket (either good or bad) on that particular day. At this stage of the experiment the basket was considered as good if all the fruits in the basket were visibly healthy and did not show any signs of fungus or decay. The basket was considered as bad even if a single fruit in the basket was decayed or had traces of fungus. This was a temporary set of labels that was used and the actual labels that are assigned to the image is explained in the subsequent sections. The excel sheet was updated on a daily basis.

Once the images were captured on any given day, they were transferred from the camera to the local system. While saving the image in the system, it was renamed in the format of the basket number followed by an underscore and the corresponding day on which the image was captured. For example, the image from the first basket of Aldi captured on day two was named as A1\_2. This helped to relate the information stored in the excel with the actual image.



Figure 4.1: Image of a raspberry basket on day one of the experiment



Figure 4.2: Image of a raspberry basket on day twelve of the experiment

## 4.2 Image segmentation

Once the images were captured for a duration of twelve days, the next step of the experiment was to create smaller segments of images from one large image that was obtained from the camera. Image segmentation is defined as the process of partitioning a large image into several smaller images. Image segmentation in this case is carried out to achieve three main goals.

- i. Increase the size of the dataset
- ii. Facilitate the process of learning by highlighting the region of interest.
- iii. Ease the process of labelling the data by concentrating on small specific chunks of the image rather than considering the entire image.

An image that is segmented into smaller images is shown in figure 4.3.

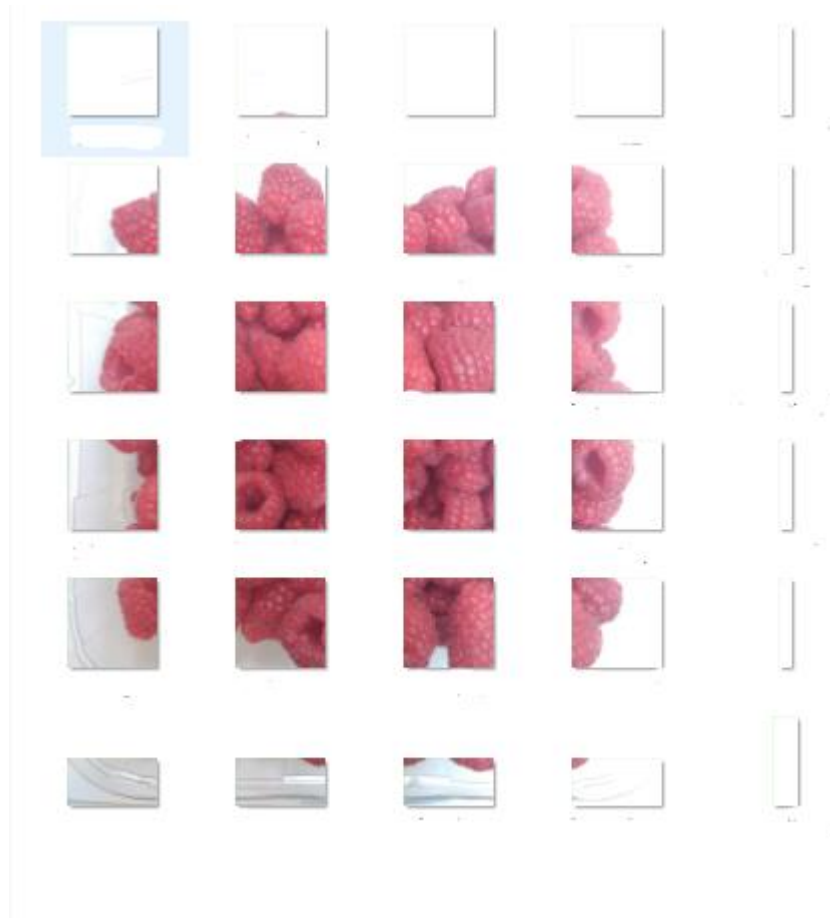


Figure 4.3: The segmented images of a raspberry basket

#### 4.2.1 Increasing the size of the data

Yalcin, (2017) in his study involving the detection of phenological changes in plants made use of image segmentation in order to increase the size of the dataset. In this experiment, the number of images that were collected from the six baskets of raspberries for twelve days was 72 images. In order to increase the instance of the images in the dataset, image segmentation was taken up. Various segment sizes like 500 x 500, 600 x 600, 650 x 650 were tried and tested. But a segment size of 750 pixels by 750 pixels was selected. The objective behind selecting the appropriate segment size was to make sure that the smaller segmented images can fit in exactly one full raspberry fruit from the basket that contained many individual raspberries. This objective could not be met fully as the individual raspberries in the basket was not arranged in a grid format but were naturally stacked making

it difficult to extract only a single fruit per image. The size 750 x 750 was the closest to fulfill this objective and hence was selected.

#### **4.2.2 Highlighting the region of interest and ease of data labelling**

Creating smaller segments of images helps in highlighting the region of interest in a Deep learning application (Mohanty, Hughes & Salathé, 2016). By creating smaller segments of the original image, the area that contained the decay in the fruit was more highlighted than using the original image. This helps to identify the decay in an efficient manner and thereby aid in the process of learning.

Segmenting the image also enables the data to be labelled easily as against using the full-sized image (Chen et al., 2017). The original image contained only small part that showed signs of decay. Though major portion of the image did not show any signs of decay, the entire image had to be labelled as Bad which could confuse the model and affect its performance. Segmenting the image helped overcome this problem as it facilitated the labelling of smaller bits of the image which also helped the learning process.

OpenCV library package provided by Python programming language was used to read the images from the local system into the python environment. Once read, these images were segmented into 750 x 750 dimensions and saved back into the local system. The naming of the segmented images extended the names of their original image with a number starting from one for the first segmented image while the final segmented image had the number of the last segment of their respective original image. This was done so that the label that contains the information regarding the number of days prior to which the basket of fruit rots could be assigned in an efficient and accurate manner to the segmented images based on the information that was previously stored in the excel.



Figure 4.4: Original Image of the raspberry containing a decay which is not highlighted



Figure 4.5: A segmented image of the original image where the decay region is highlighted

### **4.3 Data profiling**

This part of the report explains the process undertaken to obtain a clean set of images to be used for training the model.

#### **4.3.1 Irregular image sizes**

The segmentation process was initiated from the top left corner of the original image. With the original image of size 4160 x 3120, and the segmentation dimensions set to 750 x 750, some of the segmented images were not exactly of size 750 x 750. In particular, the segmented images that belonged to the extreme right and also to the extreme bottom of the original image were not of size 750 x 750. This is because neither 4160 nor 3120 is completely divisible by 750. Each of the original image yielded thirty segmented images out of which twenty images were of size 750 x 750, four images of size 410 x 750, five images of 750 x 120 and one image of 410 x 120.

To maintain a uniform size of the images to feed as input to the CNN, the images that were less than 750 x 750 were deleted and not considered for the experiment. The process of deletion was done manually after taking note of their sizes. These images mostly contained the background information and not the actual fruit itself so deleting them did not result in any major loss of information.

#### **4.3.2 Minimum fruit coverage in the image**

Removing the background information helps in increasing the accuracy of prediction (Mohanty, Hughes & Salathé, 2016). After filtering out the images that were not of size 750 x 750, the next step was to identify images that had more background information and less information about the fruit. A threshold of 50% background information was decided and images that contained background information more than this threshold were not considered. The process of deciding the percentage of background information in an image was through visual inspection of the image. Some images that were on the borderline of this threshold and difficult to decide were retained and used in the experiment.



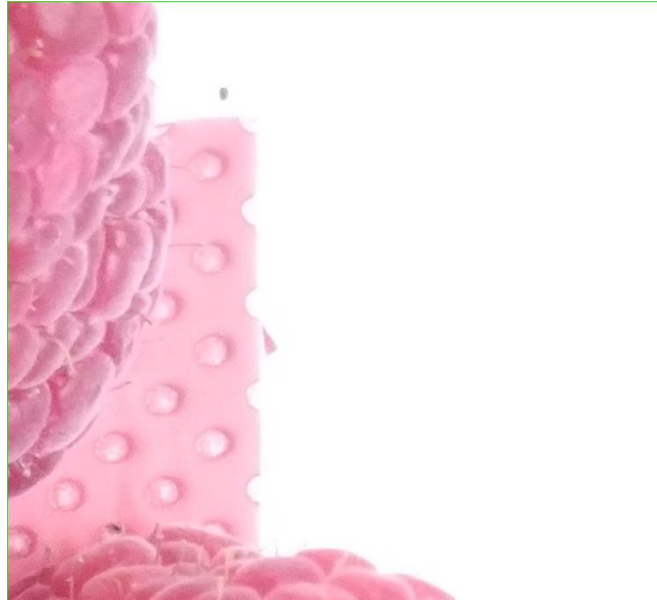


Figure 4.6: A segmented image whose background information is greater than threshold



Figure 4.7: A segmented image whose background information is on the borderline of the threshold and hence considered for the experiment

### 4.3.3 Identifying and removing pixelated images

Pixelation is a term that describes blur sections in an image. Object in a blurred image is difficult to detect, see or identify. Blur images increase ambiguity and uncertainty. Use of a blur image in object detection, recognition or identification, results in a poor performance (Takahshi & Watanabe, 2015). In this experiment, a few segmented images were also blur. To minimize the effect of blur images on the performance of the CNN model, the images that were identified as blur, were removed from the collection of images and not used in the experiment. All the images from the collection of the segmented images were inspected and the blur images were identified through visual inspection.

Once the images were segmented and the profiling was done, the final set of images that were to be used in the experiment was available.



Figure 4.8: A segmented image that is pixelated or blurred

## 4.4 Data labelling

After the images were segmented and profiled, a total of 1128 images were available. The experiment aimed to achieve two goals. The first goal was to identify if the fruits were good or bad. The second goal was to estimate the number of days before which the fruits would

turn bad. Each image was assigned with two different set of labels to achieve the above-mentioned goals.

#### **4.4.1 Labels to identify good vs bad fruits**

The images were labelled as good and bad based on the visual perception of the researcher. If the fruits in the image appeared to have some traces of fungus or decay, the image was labelled as bad. If the image did not show any signs of fungus or decay, the image was labelled as good. The overall area of the image that contained signs of decay was not considered as a factor to label the image. An image in which the decayed area of the fruit spread across a larger region of the image was labelled at par with an image whose decayed area was limited to a small region. The labels for this part of the experiment is a binary variable containing values 0 and 1 to represent the Bad and Good images respectively. A comma separated value file was used to store the names of the images and the corresponding label information.



Figure 4.9: Images that are labelled as good

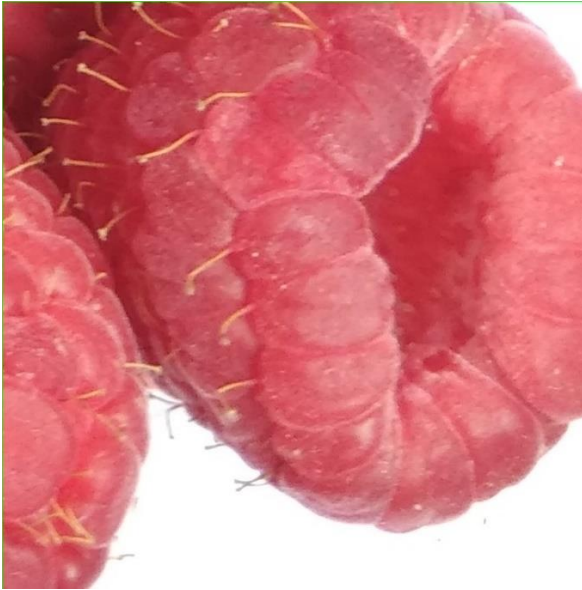


Figure 4.10: Another set of images that are labelled as Good



Figure 4.11: Images that are labelled as Bad





Figure 4.12: Another set of images that are labelled as Bad

#### **4.4.2 Labels to identify the numbers of days left before the fruit rots**

In the later part of the experiment, the images were labelled using the information that was collected and stored in an excel at the time of image collection. The excel contained the information regarding the condition of the basket of fruits on any given day of the experiment. For this experiment, the segments of the original images, that was labelled as Good at the time of data collection was only used. Images that were labelled as bad at the time of capturing the image was not used.

Suppose at the time of data collection, an image of a particular basket of raspberries say 'X' was labelled as good right from day 1 of the experiment based on its physical appearance and then, on day N, due to the presence of fungus or signs of rotting, the basket 'X' was labelled Bad. At this point, the image of basket 'X' on day N-1 was labelled as 1, image on day N-2 was labelled as 2, image on day N-3 as 3 and so on and so forth till the first day of the experiment. Their corresponding segmented images obtained from basket 'X' that remained after data profiling, was labelled using the same logic. As mentioned previously, the segmented images were named based on the basket number, information about the day of the experiment, followed by the segment number (Segments of images from the first

basket that was bought from Aldi and its picture taken on day 2 of the experiment was named from A1\_2\_1 to A1\_2\_30). Using this naming convention as the reference, the original image and their segmented images were correlated and the segmented images were labelled accordingly.

The labels consisted of discrete values in the range of 1 to 8. Concept of binning was used and images with labels 7 and 8 were grouped together and assigned a new label value of 0. Here values 7 and 8 are the number of days left for the fruit to rot and it indicates that the fruits are fresh and have sufficient time to be consumed. Images with labels 4, 5 and 6 were grouped and assigned a new label value of 1 which indicated that the fruits are relatively fresh and it needs to be consumed quickly. Images with labels 1,2 and 3 constituted the last group and assigned the new label value of 2 which indicates that the fruits do not have much time left before it rots. A total of 351 images were used as a part of this experiment.

#### **4.5 Dataset creation**

The images that were used to conduct the part of the experiment to check whether the fruit was Good or Bad was stored in a folder. These images were sorted according to their names. A comma separated value (csv) file was used to store the names of these images along with their corresponding label information. This file was stored inside the same folder that contained the images.

The same process was also carried out for the other part of the experiment to predict the number of days before the spoilage of the fruit. The images along with the csv file was stored in a folder.

#### **4.6 Importing data using TFRecords**

There are multiple ways to create a dataset that is read by a CNN. In this experiment, a tfrecord file was used to create a dataset and later on, this data was used as a part of the computational graph. Tensorflow framework provides various input pipelines to read the data from the disk into memory. TFrecords is one such input pipeline that converts the dataset into a binary file. Since the images and their corresponding labels are stored separately, using a TFRecord pipeline stores the image and their label as a binary file in one

block of memory. The TFRecord creates a binary file and is stored with .tfrecord as the file extension.

The data was divided into training and test datasets. 70% of the data was used for training purpose and 30% was used for testing purpose. Two separate tfrecord files were created for training and test datasets.

## **4.7 Modelling**

The hierarchical architecture that is used in a CNN is gradually proved to be a successful and efficient way to learn visual representations (Hu, W. et al., 2015). A CNN with a strong predictive ability requires that an effective optimization of the hyperparameters. A number of parameters must be set while creating a CNN which includes selecting the optimal number of layers in the CNN, the number of neurons in each layer, selecting the most appropriate optimization algorithm, a suitable learning rate, batch size and also the number of epochs of training to name a few.

Bergstra et al. (2011) while considering about optimizing hyperparameters noted that human judgment has been historically used to set the hyperparameters due to its relative efficiency in computational environments. Hyperparameters as a part of this experiment is set based on human judgment.

The architecture of the proposed CNN to classify the fruits consisted of 9 layers made up by the input layer (I1), First Convolutional layer (C1), followed by the max pooling layer (M1), Second Convolutional layer (C2), second max pooling layer (M2), Third Convolutional Layer (C3), third max pooling layer (M3), followed by the fully connected layer (FC1) and the output layer (O1).

In this architecture, layers C1, M1, C2, M2, C3 and M3 could be viewed as the trainable feature extractors to the Spectral image. Layer FC1 is the trainable classifier to these feature extractors. A total of 112 features can be extracted from the input image. In the first Convolutional layer, 16 features each of size 20 x 20 could be extracted from the input image. The second convolutional layer has 32 feature detectors of size 15 x 15 and the last

Convolutional layer has 64 feature detectors of size 6 x 6. The stride of 4 x 4 is used across all the convolutional layers.

	Number of filters	Size of the filters	Stride
Convolutional Layer 1	16	20 x 20	4 x 4
Convolutional layer 2	32	15 x 15	4 x 4
Convolutional layer 3	64	6 x 6	4 x 4

Table 4.1: Specification of the Convolutional layers used in the experiment

Classifying a Spectral image using a CNN requires the parameters regarding the number of channels of the image and the number of output classes. The parameter for the number of channels is set to 3 as the image has three channels namely Red, Green and Blue. The parameter for the number of output classes was set to 2 for classifying the image as good or Bad. The same parameter was set to 3 for classifying the number of days to rot.

Tensorflow neural network wrapper is used in this project to train and test the model. The hyperparameters, batch size and number of epochs are set to 64 and 50 respectively. Batch size parameter is used to adjust the error rate of the model after the number of samples specified is trained. The model is trained iteratively for numerous times which is specified by the number of epochs.

The training process is carried out in two steps:

- i. Forward Propagation
- ii. Back Propagation

The forward propagation is used to compute the classification result with respect to the input data based on the current trainable network parameters such as weight. The initial weights are randomly selected and used in the network. The back propagation is used to update these



trainable network parameters in order to reduce the discrepancy between the actual classification output and the expected classification output.

This study employs a 9-layer CNN network consisting of 750 x 750 input units (as size of the input image is 750 x 750) in the input layer(I1) and 2 output units in the Output layer (O1). There are several hidden layers C1, M1, C2, M2, C3, M3 and FC1. If we assume  $x_i$  as the input to layer  $i$ , then  $x_{i+1}$  is calculated as

$$x_{i+1} = f_i(W_i x_i + b_i) \quad (4.1)$$

where,  $W_i$  is the weight matrix of layer  $i$ , and  $b_i$  is the bias unit.

$F_i()$  is the activation function used in the experiment. A Rectified Linear Units (ReLU) activation function is used in layers C1, C2, C3 and FC1. A max pooling function is used in the pooling layers M1, M2 and M3. The output of the Fully connected layer (FC1) is fed to the output layer (O1) using a Softmax function.

In the backpropagation stage of the training, the weights of the model are adjusted and updated using an Adam Optimizer algorithm. A cross entropy loss function is used as the cost function. As the model is trained over a number of iterations, the accuracy of the model increases and the training is stopped at 50 iterations and evaluated against the test dataset.

#### **4.8 Predicting the current state of the fruit**

The architecture of the model that was used to predict the current state of the fruit as good or bad is as discussed in section 4.7. A total of 1128 images were used as a part of this section of the experiment. The data labels for the images is as discussed in section 4.4.1.

The dataset was split into training and test. 790 images were used to train the model and 338 images were used to test the classification accuracy. The accuracy of the Convolutional Neural network model varied between 82% and 90% on the test dataset with an average of 86%.

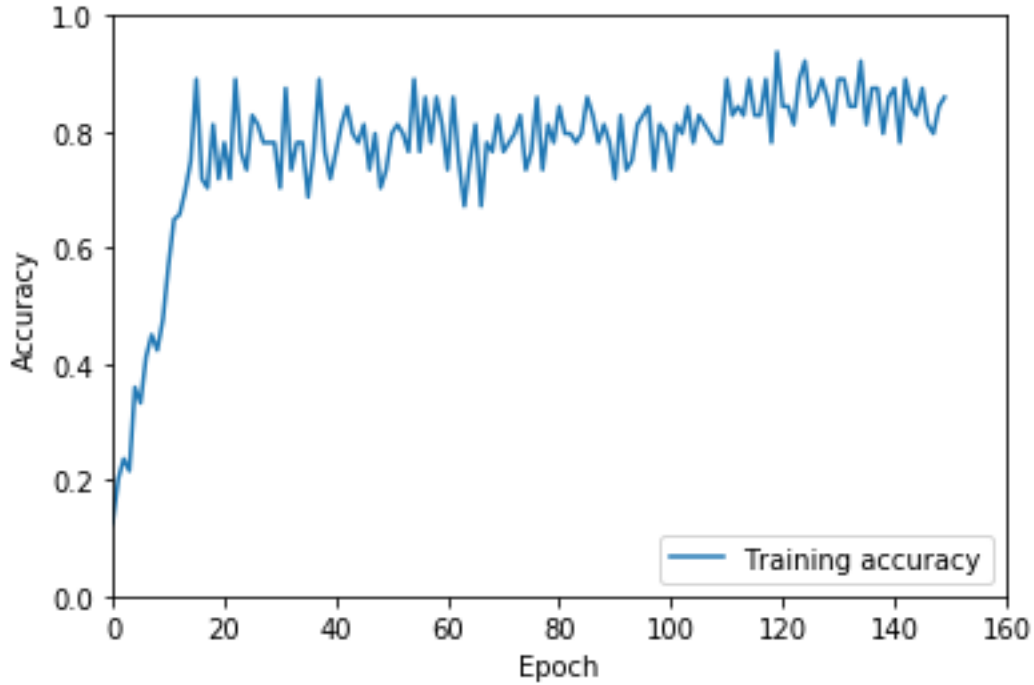


Figure 4.13: The graph of the training accuracy of the model.

With the data consisting of 990 images that were labelled Good and 138 images as Bad, the F1 scores on an average was 0.82.

	Minimum	Maximum	Average
Accuracy	82.35%	90.26	86%
F1 score	0.79	0.85	0.82

Table 4.2: The maximum, minimum and average Accuracy and F1score of the model to predict current condition of the fruit

#### 4.9 Estimating the number of days before the fruit rots

The same architecture of the CNN was used as defined in section 4.7 was used to estimate the number of days left for the fruit to rot. The data labels for the images is explained in section 4.4.2.

A total of 351 images were used for this experiment out of which 32 images had a label of 0 (indicates the fruits are fresh and have a sufficient amount of time to be consumed), 85 images had a label of 1 (indicates the fruits are relatively fresh and needs to be consumed

quickly) and 234 images had label 2 (to indicate that the fruit does not have much time left before it rots).

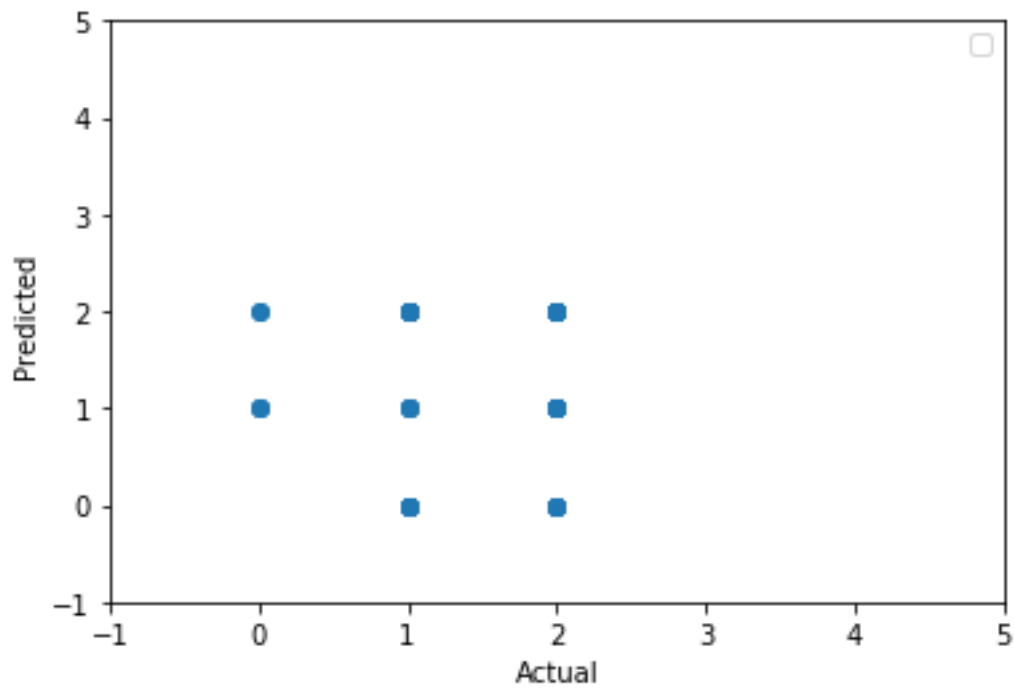


Figure 4.14: A scatter plot showing the actual and predicted values from the model

The performance of the model when predicting the values is as shown in figure 4.14. The accuracy and the F1 score of the model was very low and the model could not produce the expected results.

	Values
Accuracy	47%
F1 score	0.53

Table 4.3: Accuracy and F1scores of the model to estimate number of days

## **4.10 Summary**

This chapter provided a detailed explanation of the steps that were implemented in this experiment to answer the research question. The data collection process, processing of the collected data, adding annotations to the images, the architecture of the deep learning model and the results obtained from the experiments are covered. A detailed analysis of the results is discussed and evaluated in chapter 5 which will be used to accept or reject the hypothesis.

## **Chapter 5**

### **Evaluation and Analysis**

This chapter builds on the detailed evaluation and analysis of the results obtained from the experiment documented in Chapter 4. The reason why the current architecture of the model was chosen over other architectures that were tested is also briefed. The results of these variations are evaluated against the accuracy and F1 scores of the model. The outline of how this study helps to extend the knowledge of the existing body is also discussed.

#### **5.1 CNN architecture Evaluation**

The images used in this experiment were of relatively bigger size (i.e 750 x 750). To minimize the number of parameters of the model, an architecture that consisted of three convolutional layers, three pooling layers and one fully connected layer was used. Even though the complete image itself was 750 x 750, the part of the image that consisted of the decay was in some cases restricted to a very small number of pixels.

Different pool sizes and strides were used to determine the optimum fit that yielded better accuracies. A pool size of 15x15 in the first convolutional layer, 10x10 in the second and 5x5 in the third with a stride of 8x8 classified 78% of the instances correctly while a pool size of 20x20 and 15x15 and 6x6 in the first second and third convolutional layers provided a better accuracy than the earlier setup.

A batch size of 100 and 64 was tried and it did not affect the accuracy to a great extent and hence a batch size of 64 was maintained. A learning rate schedule implemented with an exponential decay was used in which the initial learning rate was set to 0.001 and a decay rate of 0.97 was used. Using an adaptive learning rate was effective than using a constant learning rate.

An epoch is defined as one complete set where the entire dataset has undergone the process of forward and back propagation. It was noticed that the number of epochs had an impact on the model. The accuracy of the model when trained with a lesser number of epochs had a lesser accuracy as against training the model using a greater number of epochs. The

classification accuracy of the model to detect the current condition of the fruit on the test dataset using epochs 10, 25 and 50 is as shown in figure 5.1. It can be seen clearly that the model with a higher epoch has a greater accuracy compared to lower epochs.

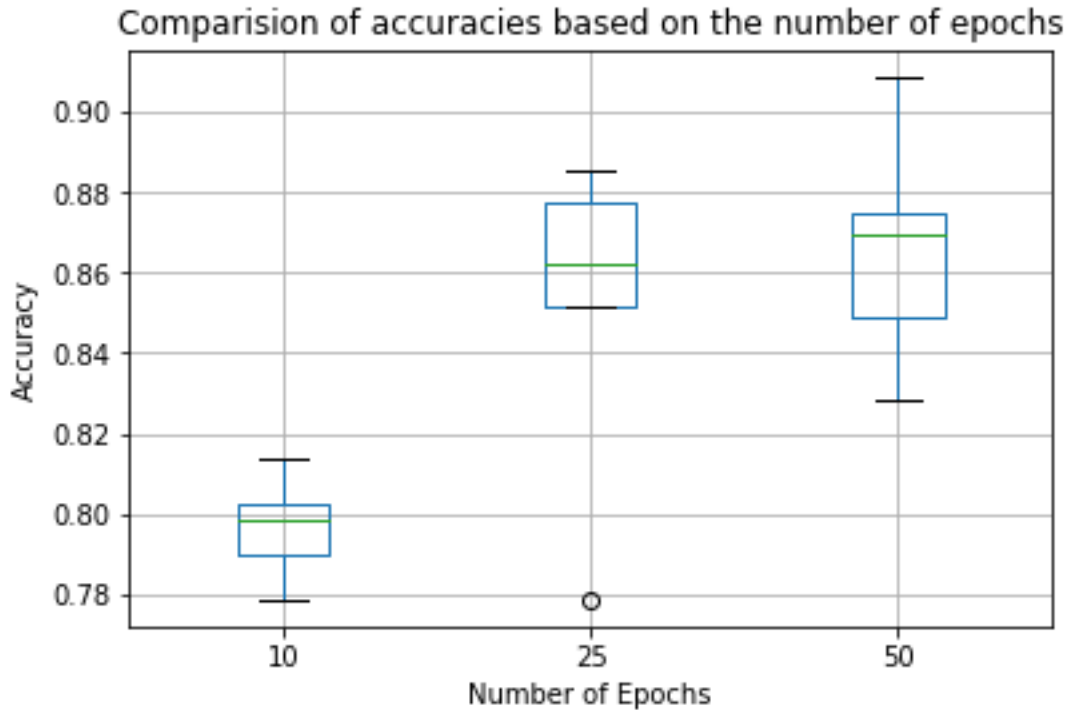


Figure 5.1: A boxplot based on the comparison of the accuracies of the model for different epochs

To avoid overfitting of the model, dropouts were implemented on the Fully connected layer. A dropout of 0.5 was used which resulted in an increase of accuracy in the range of 2 to 3 percent on the test dataset. The model showed good convergence rate for a small number of epochs.

## 5.2 Evaluating the model to predict the current condition of the fruit

The data that was used for this experiment was imbalanced. There was a total of 990 images that were labelled as good and another 138 images that were labelled as bad. The imbalance in the dataset did not impact the performance of the model much as indicated by the F1 score. A F1 score of 0.82 indicated that the model was able to classify the bad images as bad and good images as good, 86% of the time.

Even though some of the images contained background noise along with the fruit, the model was successful in identifying and classifying the images correctly. Using the blurred images in the dataset to train the model had an impact on the performance and hence was not used. The accuracy of the model during the training phase is as shown in figure 4.13 in section 4.8.

With an average accuracy of 86% and an average F1 score of 0.82, the hypothesis can be accepted and it is concluded that the Convolutional neural networks can detect the patterns from the images of raspberry fruit and predict the current condition of the fruit.

### **5.3 Evaluating the model to estimate the number of days left before rotting**

The architecture of the model that was used to estimate the number of days left before rotting was the same as the model used to predict the current condition of the fruit. Limited data was available for this experiment as the images from the raspberry basket only until it showed any signs of rotting was used. There was an imbalance in the data with 32 images with label 0, 85 images with label 1 and 234 images with label 2 were available.

The number of bins into which the images were assigned was also varied and tested. A two-bin approach was tested where the original images with label 1-6 were grouped into one bin and images with label 7 and 8 was grouped into another bin. The effect of this change on the accuracy of the model was negligible.

A scatter plot showing the actual values of the label and the prediction from the model is as shown in figure 4.14 in section 4.9. It can be seen that the model is not able to detect the changes and make correct predictions. The fruits showed gradual discoloring over the period which could not be recognized by the model.

The F1 scores and the accuracy of the model as seen in table 4.3 also suggest that the model failed to estimate the number of days left for the fruit to rot. On the basis of this, the hypothesis can be rejected and it is concluded that the Convolutional Neural Network cannot detect patterns from the raspberry fruit to estimate the number of days left before the fruit rots.

## **5.4 Strengths and limitations of the result**

The ability of Convolutional Neural networks to process images of raspberry's and predict their condition is studied as a part of this research study. The experiments are conducted to answer two questions that are similar and relate to the condition of the fruit while differs from the data that is used and also the labels.

The research is carried out on the data that was collected by the researcher for this experiment. This can be considered as the strength of the research as understanding and processing the data becomes easier. The images used for the experiment all have standard sizes which makes it easier to feed them into a CNN model.

The architecture of the models used for the experiment was selected after testing various configurations of the Convolutional layers and pooling layers which is used for feature detection. This also adds as a strength of the research. The same architecture can be used for various other datasets in the same domain to predict spoilage of different fruits as the model is not data specific.

The model is also trained on images which were captured under natural light and also artificial lights. The variation in the lighting does not affect the performance of the model which will be useful in industrial applications. The model also has a good convergence rate for a small number of epochs of training.

The imbalance in the datasets is one of the limitations of the model. In order to make the model more robust, a balanced dataset can be used. The model could not detect the changes to predict the number of days before rotting. The image used is an RGB image which captures only the exterior aspects of the fruit. Using a hyperspectral camera which captures more information which a human eye cannot detect could be used to capture the images and feed them into the model to see if the changes could be detected.

The number of images that the models are trained on is less. Using more data can help to train the model over a larger corpus of data which can help balance the data set and also improve the performance of the model. Data is not pre-processed to remove the background noise and irrelevant information in the image which could also be considered as a drawback.



## **Chapter 6**

### **Conclusion**

A brief overview of the overall work performed, the results that are achieved in the experiment and the conclusions reached are provided in this chapter. The later part of the chapter will assess the impact and the contributions of this research and also outline the areas of work which needs to be addressed in the future arising from the findings of this research.

#### **6.1 Problem Definition**

This research aimed to determine if Convolutional Neural Networks could determine the spoilage in raspberry's using Spectral imaging. In order to address the research question, data had to be collected from the raspberries in the form of images and build a Convolutional neural network model to find patterns from the image to determine spoilage in the fruit.

#### **6.2 Design, experimentation, evaluation and results**

The research project began by a thorough review of the literature to gain knowledge about the techniques that are used to detect spoilage in food products and also the ability of Convolutional Neural networks to process images and find patterns in them to solve problems. The literature review also covered the use of Deep learning techniques and Hyperspectral images in the field of Agriculture.

Having reviewed the literatures related to the research question, the focus of the experiment was to collect data and prepare a dataset. Images were collected from the basket of raspberry's using a digital camera over a period of time. Once the images were collected, they had to be pre-processed to break them into smaller segments and also remove some images from the collection of images.

The next step of the experiment was to annotate these images. The images were planned to be used in determining the current state of the fruit (either good or bad) and also estimate

the number of days before which the fruits would turn bad. So, the images were labelled with two separate labels with each label addressing the above problems. The labelling was done according to the researcher's visual perception about the quality of the fruit in the image.

With the dataset ready to be used, the experiment moved on to designing and building a Convolutional Neural network. The architecture of the neural network regarding the number of layers that the network should have, hyperparameters like the batch size, number of iterations were decided based on the researcher's judgment and experimenting with various values for the hyperparameter. The activation functions and also the optimization algorithms were chosen and the model was built using the Tensorflow framework in Python development environment.

The CNN model developed achieved a considerable level of predictive power to support the claim that a neural network can predict the current state of the raspberry fruit. The accuracy and also the F1 score metrics provided sufficient proof to claim the results. However, the CNN model could not estimate the number of days before which the fruit rots.

### **6.3 Contributions and Impact**

Having pointed out in the literature review about the limited research into the use of Deep learning techniques like Convolutional Neural Network to detect spoilage in fruits like raspberry, this research makes use of the capabilities of Convolutional Neural networks to address the problem.

The research also provides a non-destructive solution to find spoilage making use of computer vision and thus overcomes the problem of manual inspection which is time consuming and also subject to errors. This research project has developed and implemented a deep learning model to find patterns from a raspberry image and demonstrated that deep learning techniques has the potential to detect the current state of the fruit.

A framework has been established and a baseline model has been set from this research which could be used as a starting point for future research in this field and also be used for practical deployment.

## **6.5 Future work and recommendations**

The significant limitation of this research is it uses only the exterior aspects of the raspberry to make decisions about its quality. Future work involves the use of a hyperspectral camera to capture the images of the fruits. A multispectral image gives more information than a normal RGB which could help to maximize the predictive power of the model.

The dataset used for this research is imbalanced with majority of the images labelled as Good. More images can be clicked of fruits once they have gone bad to get an equal distribution of good and bad images which will help to train the model more effectively.

Other aspect that the research does not consider is the environmental factors that affect the fruit. The fruits were kept in the refrigerator throughout the experiment. In order to measure the effects of the environmental factors, future study will consider keeping the fruits outside the refrigerator in normal room temperature and observe the effects of sunlight, oxygen and other factors on the fruit.

The images have been labelled according to the researcher's perspective. Going forward, more scientific methods could be used to label the data. Also, the background information in the image is retained in this research. As a future study, the image may be pre-processed to remove the background information from the image and concentrate only on the fruits which could have a positive impact on the classification accuracy of the model.

## References

- Beekwilder, J., Hall, R., & Ric Vos, C. (2005). Identification and dietary relevance of antioxidants from raspberry. *Biofactors*, 23(4), 197-205.
- Bergstra, J. Bardenet, R. Bengio, Y. Kegl, B. (2011). Algorithms for Hyperparameter optimisation. Proceedings of Advances in Neural Information Processing Systems 24 (NIPS 2011). Retrieved from: <http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization>
- Brosnan, T., & Sun, D.W., (2004). Improving quality inspection of food products by computer vision. *Journal of Food and Engineering*. 61 (1), 3–16.
- Chelladurai, V., Jayas, D., & White, N. (2010). Thermal imaging for detecting fungal infection in stored wheat. *Journal Of Stored Products Research*, 46(3), 174-179.
- Chen, H., Xu, J., Xiao, G., Wu, Q., & Zhang, S. (2018). Fast auto-clean CNN model for online prediction of food materials. *Journal Of Parallel And Distributed Computing*, 117, 218-227.
- Chen, S., Shivakumar, S., Dcunha, S., Das, J., Okon, E., & Qu, C. et al. (2017). Counting Apples and Oranges With Deep Learning: A Data-Driven Approach. *IEEE Robotics And Automation Letters*, 2(2), 781-788.
- Dias, P., Tabb, A., & Medeiros, H. (2018). Apple flower detection using deep convolutional networks. *Computers In Industry*, 99, 17-28.
- Dintwa, E., Van Zeebroeck, M., Ramon, H., & Tijskens, E. (2008). Finite element analysis of the dynamic collision of apple fruit. *Postharvest Biology And Technology*, 49(2), 260-276.
- Donis-González, I., Guyer, D., Leiva-Valenzuela, G., & Burns, J. (2013). Assessment of chestnut (*Castanea spp.*) slice quality using color images. *Journal Of Food Engineering*, 115(3), 407-414.
- Dyrmann, M., Karstoft, H., & Midtiby, H. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*, 151, 72-80.

- Giovanelli, G., Limbo, S., & Buratti, S. (2014). Effects of new packaging solutions on physico-chemical, nutritional and aromatic characteristics of red raspberries (*Rubus idaeus* L.) in postharvest storage. *Postharvest Biology And Technology*, 98, 72-81.
- Gómez, A., He, Y., & Pereira, A. (2006). Non-destructive measurement of acidity, soluble solids and firmness of Satsuma mandarin using Vis/NIR-spectroscopy techniques. *Journal Of Food Engineering*, 77(2), 313-319.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., & Shuai, B. et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
- Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep Convolutional Neural Networks for Hyperspectral Image Classification. *Journal of Sensors*.
- Kamencay, P., Benco, M., Mizdos, T., & Radil, R. (2017). A New Method for Face Recognition Using Convolutional Neural Network. *Advances In Electrical And Electronic Engineering*, 15(4). doi: 10.15598/aece.v15i4.2389
- Kamilaris, A., & Prenafeta-Boldú, F. (2018). Deep learning in agriculture: A survey. *Computers And Electronics In Agriculture*, 147, 70-90.
- Kamruzzaman, M., Barbin, D., ElMasry, G., Sun, D., & Allen, P. (2012). Potential of hyperspectral imaging and pattern recognition for categorization and authentication of red meat. *Innovative Food Science & Emerging Technologies*, 16, 316-325.
- Kondo, N., Kuramoto, M., Shimizu, H., Ogawa, Y., Kurita, M., & Nishizu, T. et al. (2009). Identification of Fluorescent Substance in Mandarin Orange Skin for Machine Vision System to Detect Rotten Citrus Fruits. *Engineering In Agriculture, Environment And Food*, 2(2), 54-59.
- Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267, 378-384.
- Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions On Geoscience And Remote Sensing*, 55(2), 645-657.

- Moallem, P., Serajoddin, A., & Pourghassem, H. (2017). Computer vision-based apple grading for golden delicious apples based on surface features. *Information Processing In Agriculture*, 4(1), 33-40.
- Mohanty, S., Hughes, D., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers In Plant Science*, 7.
- Narvankar, D., Singh, C., Jayas, D., & White, N. (2009). Assessment of soft X-ray imaging for detection of fungal infection in wheat. *Biosystems Engineering*, 103(1), 49-56.
- Pan, S., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions On Knowledge And Data Engineering*, 22(10), 1345-1359.
- Rao, A.V., & Snyder, D.M. (2010). Raspberries and human health: a review. *J. Agric. Food Chem.* 58, 3871–3883.
- Rungpichayapichet, P., Mahayothee, B., Nagle, M., Khuwijitjaru, P., & Müller, J. (2016). Robust NIRS models for non-destructive prediction of postharvest fruit ripeness and quality in mango. *Postharvest Biology And Technology*, 111, 31-40.
- Santoni, M., Sensuse, D., Arymurthy, A., & Fanany, M. (2015). Cattle Race Classification Using Gray Level Co-occurrence Matrix Convolutional Neural Networks. *Procedia Computer Science*, 59, 493-502.
- Sanzani, S.M., Reverberi, M., & Geisen, R. (2016). Mycotoxins in harvested fruit and vegetables: insights in producing fungi biological role, conducive conditions, and tools to manage postharvest contamination. *Postharvest Biol. Technol.* 122, 95–105.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117.
- Siedliska, A., Baranowski, P., Zubik, M., Mazurek, W., & Sosnowska, B. (2018). Detection of fungal infections in strawberry fruit by VNIR/SWIR hyperspectral imaging. *Postharvest Biology And Technology*, 139, 115-126.
- Sindagi, V., & Patel, V. (2018). A survey of recent advances in CNN-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107, 3-16.

Søgaard, H. (2005). Weed Classification by Active Shape Models. *Biosystems Engineering*, 91(3), 271-281.

Takahshi, K., & Watanabe, K. (2015). Effects of image blur on visual perception and affective response. *2015 7Th International Conference On Knowledge And Smart Technology (KST)*. doi: 10.1109/kst.2015.7051480

Yalcin, H. (2017). Plant phenology recognition using deep learning: Deep-Pheno. *2017 6Th International Conference On Agro-Geoinformatics*. doi: 10.1109/agro-geoinformatics.2017.8046996

Yang, X., Liu, C., Wang, Z., Yang, J., Min, H., Wang, L., & Cheng, K. (2017). Co-trained convolutional neural networks for automated detection of prostate cancer in multi-parametric MRI. *Medical Image Analysis*, 42, 212-227.

Ye, D., Sun, L., Tan, W., Che, W., & Yang, M. (2018). Detecting and classifying minor bruised potato based on hyperspectral imaging. *Chemometrics And Intelligent Laboratory Systems*, 177, 129-139.

zhang, C., Guo, C., Liu, F., Kong, W., He, Y., & Lou, B. (2016). Hyperspectral imaging analysis for ripeness evaluation of strawberry with support vector machine. *Journal Of Food Engineering*, 179, 11-18.

## Appendix

### Python code to create tfrecords dataset

```
from random import shuffle
import glob
import cv2
import tensorflow as tf
import sys
import numpy as np
shuffle_data = True # shuffle the addresses before saving
raspberry_images =
'C:/Users/Dell/Desktop/Masters/Dissertation/Data_Cropped_Images_Copy/*.jpg'
# read addresses and labels from the 'train' folder
addrs = glob.glob(raspberry_images)
import pandas as pd
Data = pd.read_csv('C:/Users/Dell/Desktop/Masters/Dissertation/Images_data.csv')
labels = Data['Condition'] # 0 = Bad, 1 = Good
# to shuffle data
if shuffle_data:
    c = list(zip(addrs, labels))
    shuffle(c)
    addrs, labels = zip(*c)

# Divide the hata into 70% training and 30% test
train_addrs = addrs[0:int(0.7*len(addrs))]
train_labels = labels[0:int(0.7*len(labels))]
#val_addrs = addrs[int(0.6*len(addrs)):int(0.8*len(addrs))]
#val_labels = labels[int(0.6*len(addrs)):int(0.8*len(addrs))]
test_addrs = addrs[int(0.7*len(addrs)):]
```



```

test_labels = labels[int(0.7*len(labels)):]

def load_image(addr):
    # cv2 load images as BGR, convert it to RGB
    img = cv2.imread(addr)
    img = cv2.resize(img, (750, 750), interpolation=cv2.INTER_CUBIC)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = img.astype(np.float32)
    return img

def _int64_feature(value):
    return tf.train.Feature(int64_list=tf.train.Int64List(value=[value]))

def _bytes_feature(value):
    return tf.train.Feature(bytes_list=tf.train.BytesList(value=[value]))

train_filename = 'train.tfrecords' # address to save the TFRecords file
# open the TFRecords file
writer = tf.python_io.TFRecordWriter(train_filename)
for i in range(len(train_addrs)):
    # print how many images are saved every 1000 images
    if not i % 50:
        print ('Train data: { }/{ }'.format(i, len(train_addrs)))
        sys.stdout.flush()
    # Load the image
    img = load_image(train_addrs[i])
    label = train_labels[i]
    # Create a feature
    feature = {'train/label': _int64_feature(label),

```

```

        'train/image': _bytes_feature(tf.compat.as_bytes(img.tostring()))}

# Create an example protocol buffer
example = tf.train.Example(features=tf.train.Features(feature=feature))

# Serialize to string and write on the file
writer.write(example.SerializeToString())

writer.close()
sys.stdout.flush()

# open the TFRecords file
test_filename = 'test.tfrecords' # address to save the TFRecords file
writer = tf.python_io.TFRecordWriter(test_filename)
for i in range(len(test_addrs)):
    # print how many images are saved every 1000 images
    if not i % 50:
        print ('Test data: {}/{}'.format(i, len(test_addrs)))
        sys.stdout.flush()
    # Load the image
    img = load_image(test_addrs[i])
    label = test_labels[i]
    # Create a feature
    feature = {'test/label': _int64_feature(label),
              'test/image': _bytes_feature(tf.compat.as_bytes(img.tostring()))}
    # Create an example protocol buffer
    example = tf.train.Example(features=tf.train.Features(feature=feature))
    # Serialize to string and write on the file
    writer.write(example.SerializeToString())

```

```
writer.close()
sys.stdout.flush()
```

## Python code to the CNN model

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix

train_data_path = 'train_new.tfrecords' # address to save the hdf5 file
test_data_path = 'test_new.tfrecords'

def weight_variable(shape):
    with tf.variable_scope("weight",reuse=tf.AUTO_REUSE):
        return
    tf.get_variable("weight",shape=shape,initializer=tf.truncated_normal_initializer(stddev=0.1))

def weight_variable1(shape):
    with tf.variable_scope("weight1",reuse=tf.AUTO_REUSE):
        return
    tf.get_variable("weight1",shape=shape,initializer=tf.truncated_normal_initializer(stddev=0.1))

def weight_variable2(shape):
    with tf.variable_scope("weight2",reuse=tf.AUTO_REUSE):
        return
    tf.get_variable("weight2",shape=shape,initializer=tf.truncated_normal_initializer(stddev=0.1))
```

```

def weight_variable3(shape):
    with tf.variable_scope("weight3",reuse=tf.AUTO_REUSE):
        return
tf.get_variable("weight3",shape=shape,initializer=tf.truncated_normal_initializer(stddev=
0.1))

def weight_variable4(shape):
    with tf.variable_scope("weight4",reuse=tf.AUTO_REUSE):
        return
tf.get_variable("weight4",shape=shape,initializer=tf.truncated_normal_initializer(stddev=
0.1))

def bias_variable(shape):
    with tf.variable_scope("bias",reuse=tf.AUTO_REUSE):
        return tf.get_variable("bias",shape=shape,initializer=tf.constant_initializer(0.1))

def bias_variable1(shape):
    with tf.variable_scope("bias1",reuse=tf.AUTO_REUSE):
        return tf.get_variable("bias1",shape=shape,initializer=tf.constant_initializer(0.1))

def bias_variable2(shape):
    with tf.variable_scope("bias2",reuse=tf.AUTO_REUSE):
        return tf.get_variable("bias2",shape=shape,initializer=tf.constant_initializer(0.1))

def bias_variable3(shape):
    with tf.variable_scope("bias3",reuse=tf.AUTO_REUSE):
        return tf.get_variable("bias3",shape=shape,initializer=tf.constant_initializer(0.1))

def bias_variable4(shape):

```

```

with tf.variable_scope("bias4",reuse=tf.AUTO_REUSE):
    return tf.get_variable("bias4",shape=shape,initializer=tf.constant_initializer(0.1))

def conv2d(x, W):
    return tf.nn.conv2d(x, W, strides=[1, 4, 4, 1], padding='SAME')

def max_pool_2x2(x):
    x = tf.nn.dropout(x, 0.85)
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],strides=[1, 2, 2, 1], padding='SAME')

def my_cnn(x,num_classes, is_training):

    # Conv Layer 1: with 16 filters of size 20 x 20
    W_conv1 = weight_variable([20, 20, 3, 16])
    b_conv1 = bias_variable([16])
    x_image = tf.reshape(x, [-1, 750, 750, 3])

    h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
    # Pool
    h_pool1 = max_pool_2x2(h_conv1)

    # Conv Layer 2: with 32 filters of size 15 x 15
    W_conv2 = weight_variable1([15, 15, 16, 32])

    b_conv2 = bias_variable1([32])
    h_conv2 = tf.nn.relu(conv2d(h_pool1, W_conv2) + b_conv2)
    # Pool

```

```

h_pool2 = max_pool_2x2(h_conv2)

# Conv Layer 3: with 64 filter of size 6 x 6
W_conv3 = weight_variable2([6, 6, 32, 64])

b_conv3 = bias_variable2([64])

h_conv3 = tf.nn.relu(conv2d(h_pool2, W_conv3) + b_conv3)
h_pool3 = max_pool_2x2(h_conv3)

W_fc1 = weight_variable3([2*2*64, 64])
b_fc1 = bias_variable3([64])

# flattening output of pool layer to feed in FC layer
h_pool3_flat = tf.reshape(h_pool3, [-1, 2*2*64])

# FC layer
h_fc1 = tf.nn.relu(tf.matmul(h_pool3_flat, W_fc1) + b_fc1)

# Dropout
h_fc1_drop = tf.nn.dropout(h_fc1, 0.5)

W_fc2 = weight_variable4([64, 2])
b_fc2 = bias_variable4([2])

# Output
y_conv = tf.matmul(h_fc1_drop, W_fc2) + b_fc2

```

```

        return y_conv

with tf.Graph().as_default():

    feature = {'train/image': tf.FixedLenFeature([], tf.string),
               'train/label': tf.FixedLenFeature([], tf.int64)}

    feature_test = {'test/image': tf.FixedLenFeature([], tf.string),
                    'test/label': tf.FixedLenFeature([], tf.int64)}

    # Create a list of filenames and pass it to a queue
    filename_queue = tf.train.string_input_producer([train_data_path])
    filename_queue_test = tf.train.string_input_producer([test_data_path])

    # Define a reader and read the next record
    reader = tf.TFRecordReader()
    reader_test = tf.TFRecordReader()

    _, serialized_example = reader.read(filename_queue)
    _, serialized_example_test = reader_test.read(filename_queue_test)

    # Decode the record read by the reader
    features = tf.parse_single_example(serialized_example, features=feature)
    features_test = tf.parse_single_example(serialized_example_test, features=feature_test)

    # Convert the image data from string back to the numbers
    image = tf.decode_raw(features['train/image'], tf.float32)
    image_test = tf.decode_raw(features_test['test/image'], tf.float32)

    # Cast label data into int32

```

```

label = tf.cast(features['train/label'], tf.int32)
label_test = tf.cast(features_test['test/label'], tf.int32)

# Reshape image data into the original shape
image = tf.reshape(image, [750, 750, 3])
image_test = tf.reshape(image_test, [750,750,3])
label = tf.reshape(label,[1])
label_test = tf.reshape(label_test, [1])

num_of_training_records = 0
for record in tf.python_io.tf_record_iterator(train_data_path):
    num_of_training_records +=1
print(num_of_training_records)
num_of_test_records = 0
for record in tf.python_io.tf_record_iterator(test_data_path):
    num_of_test_records +=1
print(num_of_test_records)
batch_size = 64
# Any preprocessing here ...

# Creates batches by randomly shuffling tensors
images, labels = tf.train.shuffle_batch([image, label], batch_size=batch_size,
capacity=100, num_threads=1, min_after_dequeue=80)
images_test, labels_test = tf.train.shuffle_batch([image_test, label_test],
batch_size=batch_size, capacity=100, num_threads=1, min_after_dequeue=80)

num_classes = 2
lbl = tf.one_hot(labels,num_classes)
lbl_test = tf.one_hot(labels_test,num_classes)

```



```

lbl = tf.squeeze(lbl)
lbl_test = tf.squeeze(lbl_test)

global_step = tf.Variable(0)

learning_rate = tf.train.exponential_decay(1e-4, global_step=global_step,
decay_steps=10000, decay_rate=0.97)

function_to_map = lambda x:my_cnn(x,num_classes,is_training=True)
logits = tf.map_fn(function_to_map,images)
logits = tf.squeeze(logits)

function_to_map_test = lambda x:my_cnn(x,num_classes,is_training=True)
logits_test = tf.map_fn(function_to_map_test,images_test)
logits_test = tf.squeeze(logits_test)
y_pred_test = tf.argmax(logits_test,1)
y_true_test = tf.argmax(lbl_test,1)

cross_entropy =
tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=logits,labels=lbl))

train_step = tf.train.AdamOptimizer(learning_rate).minimize(cross_entropy,
global_step)

correct_prediction = tf.equal(tf.argmax(logits, 1), tf.argmax(lbl, 1))

accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
correct_prediction_test = tf.equal(tf.argmax(logits_test, 1), tf.argmax(lbl_test, 1))

```

```

accuracy_test = tf.reduce_mean(tf.cast(correct_prediction_test, tf.float32))

saver = tf.train.Saver()

# Initialize all global and local variables

with tf.Session() as sess:
    init_op = tf.group(tf.global_variables_initializer(), tf.local_variables_initializer())
    sess.run(init_op)

    # Create a coordinator and run all QueueRunner objects
    coord = tf.train.Coordinator()
    threads = tf.train.start_queue_runners(coord=coord, sess=sess)

    num_epoch = 1

    f1score = []
    y_true1 = []
    y_pred1 = []
    training_error = []
    test_accuracy = []

    for i in range(num_epoch):
        for j in range(0, num_of_training_records, batch_size):
            if j % 384 == 0:
                train_accuracy = sess.run(accuracy)
                training_error.append(1 - train_accuracy)
                print("step: %d, training accuracy: %g" % (j, train_accuracy))
                ts = sess.run(train_step)

        for i in range(0, num_of_test_records, batch_size):
            test_accuracy.append(sess.run(accuracy_test))

            y_true, y_pred = sess.run([y_true_test, y_pred_test])
            f1score.append(f1_score(y_true, y_pred, average='weighted'))

            cm = confusion_matrix(y_true, y_pred)
            print(cm)

```

```
print ('test accuracy %g' % np.mean(test_accuracy))
print('f1 score %g' % np.mean(f1score))

# Stop the threads
coord.request_stop()

# Wait for threads to stop
coord.join(threads)
save_path = saver.save(sess, "saved_models/Raspberry_MODEL.ckpt")
print(saver.saver_def.filename_tensor_name)
print(saver.saver_def.restore_op_name)
sess.close()
```