DEEP LEARNING METHODS FOR FILTER EXTRACTION

A Thesis Presented to the Department of

Computer Science

African University of Science and Technology

In Partial Fulfilment of the Requirements for the Degree of

Master of Computer Science

By

Falalu Ibrahim Lawan

Abuja, Nigeria

October, 2017.

CERTIFICATION

This is to certify that the thesis titled “Deep Learning Methods for Filter Extraction” submitted to the school of postgraduate studies, African University of Science and Technology (AUST) Abuja, Nigeria for the award of the Master's degree is a record of original research carried out by Falalu Ibrahim Lawan in the Department of Computer Science.

DEEP LEARNING FOR FILTER EXTRACTIONS

By

Falalu Ibrahim Lawan

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED: _____

Supervisor, Professor Lehel Csato

_____

Co-supervisor

_____

Head, Department of Petroleum Engineering

APPROVED: _____

Chief Academic Officer

_____

December 9th, 2017

ABSTRACT

With the exponential growth of the information technology, nowadays tremendous amounts of data including images, audio, text and videos, up to millions or billions, are collected for training machine learning models. Deep neural networks (DNNs) are one of the widely used methods today. Large companies in the uses these methods to recommends buyers with products, filter junk email or text-based hate speeches, understand and translate major languages in real time, and so on.

Inspired by the trend, our work is dedicated to developing and training a deep neural network to extract meaningful patterns from a set of labelled data i.e. making generalizations. We show that DNNs can learn feature representations that can be successfully applied in a wide spectrum of application domains. We show how DNNs are applied to classification problems – grading of fresh tomato fruits based on their physical qualities using supervised learning approach.

ACKNOWLEDGEMENTS

I would like to express my gratitude to African University of Science and Technology for their devotion my studies, my supervisor, Professor Csato Lehel, for introducing me to the world of Artificial Intelligence and the Machine Learning universe, his endless enthusiasm and guidance through independent research, which rendered this thesis successful. I would like especially thank Professor Amos DAVID, Head of the Department for his guidance. Also, to my roommate Lukman Ismaila, and also Muawiya Modibbo, Chalse I. Saidu, Sa'ad Usman and the entire colleagues of mine for the wise comments and encouragement or even constructive criticisms. To the AUST Artificial Intelligence for their contributions towards expanding my knowledge in AI. Also, to all people I have met in AUST at the master level, for letting me learnt and endured their diversified ethnic and religious background I did not know. Thanks to my family and friends, for supporting me and being with me physically or otherwise. Thanks, a lot!

DEDICATION

To Maryam (Hajjo), my entire family, Aisha Ibrahim, Abdurrahman Muhammed and friends who teach or taught me in one way or the other.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABC	Artificial Bee Colony
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUST	African University of Science and Technology
BLAS	Basic Linear Algebra Subroutines
CNN	Convolutional Neural Network
DNN	Deep neural networks
GDA	Generalized discriminant analysis
HOD	Head of the Department
HSV	Hue, saturation, value
JIT	Just-in-time
LDA	Linear discriminant analysis
LLE	Locally linear embedding
ML	Machine learning
MRI	Magnetic resonance imaging
NN	Neural network
PCA	Principal component analysis
RGB	Red, green, blue
RoI	Region of interest
SDA	Stepwise discriminant analysis
SPCA	Shift-invariant principal component analysis
SVM	Support Vector Machines
WSEAS	World Scientific and Engineering USA Academy and Society

CHAPTER ONE
INTRODUCTION

1.1 Background

Artificial Intelligence (AI) is transitioning from being our daily helper to something much more powerful – and disruptive – as the new machines are rapidly outperforming the most talented of us in many endeavours (Frank, Roehrig, & Pring, 2017). The branch of AI concerned with the study and design of computer programs that automatically improve with experience is called machine learning or ML. The applications of machine continue to explode in the recent years in the field of agriculture, ecommerce, health, banking, news, robotics, transportation, weather forecasts, software, industries and many more. YouTube uses ML algorithms to suggest videos, Facebook fills newsfeeds, Netflix recommends films and Google search auto-completes texts. We probably interact with machines frequently. They are used in many of the software programs that we use, such as Microsoft's infamous (and long abandoned) paperclip in Office (maybe not the most positive example), spam filters, voice recognition software, and lots of computer games (Marsland, 2015). "Your cell phone is chock full of learning algorithms. They are hard at work correcting your typos, understanding your spoken commands, reducing transmission errors, recognizing bar codes and much else. Your phone can even anticipate what you are going to do next, and advise you accordingly. For example, as you are finishing lunch it discreetly alerts you that your afternoon meeting with an out-of-town visitor will have to start late, because her flight has been delayed".

The concept of ML draws its concepts and results from many facets of knowledge, or schools of thought, namely *Symbolist*, with its origin in logic and philosophy, *Connectionist* with its origin in neuroscience, *evolutionist* with its origin in evolutionary biology, *Bayesian* with its origin in statistics and *analogizers* with their origin in psychology (Marsland, 2015). The fundamental goal

of ML is to learn to generalize (Hackeling, 2014), or to do other kinds of decision making under uncertainty. There are two main types of ML, supervised learning and unsupervised learning (Murphy, 2012). The other type of ML is called reinforcement learning.

1.1.1 Classifications

"Classification is the problem of identifying to which of a set of categories (subpopulations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known" (Tang, Alelyani, & Liu, 2014). Success in today's technologies has resulted in exponential growth in the harvested data that makes manual processing of data impractically challenging. There is no option other than to automatically recognize useful patterns from these data essentially to predict the outcome in new situations.

Spam filter for instant is an example of *classification* problems which categorize messages received as either legitimate or spam using ML. Other examples include Optical Character Recognition, text classification, documents classification, medical images classification, crop classifications and many more application areas. This is due to the increasing accuracy of image classification.

1.1.2 Deep Learning and Neural Networks

Neural networks have been used in pattern recognition and classification for a long time. A standard neural network (NN) consists of what is called a network of neurons connected as processors. Because this network is inherently inspired by neuroscience, it is called neural. Each produces a sequence of real-valued activations. "Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons. Some neurons may influence the environment by triggering actions" (Schmidhuber, 2014). Some neurons are layered input, some are layered out while others are layered in-between at multiple levels called hidden layers. The more hidden layers there are in an

NN the deeper the architecture is. The hidden layers are so called because their values are not given in the presented data; instead the model must determine which concepts are useful for explaining the relationships in the observed data (Goodfellow , Bengio, & Courville, 2016). Deep Learning is about learning features at multiple levels of representation and abstraction that help make sense of the form of data such as images, sound and text. Learning is about finding weights that make the NN exhibit desired behaviour, such as such classifying digits. ”Most deep learning algorithms are based on an optimization algorithm called stochastic gradient descent” (Goodfellow , Bengio, & Courville, 2016).

1.1.3 Mocha Package

Mocha is a deep learning framework for Julia. Close-to-C performance in native Julia code, typically there is no need to explicitly vectorize your code (as is the case for in Matlab). Mocha is inspired by the C++ framework Caffe. Efficient implementations of general stochastic gradient solvers and common layers in Mocha can be used to train deep / shallow (convolutional) NNs. Some of the advantages of Mocha are given below:

- **Modular Architecture**: Mocha has a clean architecture with isolated components like network layers, activation functions, solvers, regularizers and initializers. Mocha is rich in built-in components sufficient for typical deep (convolutional) NN applications and more are being added in each release. All of these can be easily extended by adding custom sub-types.
- **High-level Interface**: Mocha is written in Julia, a high-level dynamic programming language designed for technical computing. Combining with the expressive power of Julia and its package ecosystem, playing with deep neural networks in Mocha is easy and intuitive.

- **Portability and Speed**: Mocha comes with multiple backend choices that can be switched transparently.

    - The *pure Julia backend* is portable -- it runs on any platform that supports Julia. This is reasonably fast on small models thanks to Julia's LLVM-based just-in-time compiler and Performance Annotations, and can be very useful for prototyping.
    - The *native extension backend* can be turned on when a C++ compiler is available. It runs two to three times faster than the pure Julia backend.
    - The *GPU backend* uses NVidia® cuDNN, cuBLAS and customized CUDA kernels to provide highly efficient computation. Twenty to 30 times or even more speedup could be observed on a modern GPU device, especially on larger models.

- **Compatibility**: Mocha uses the widely adopted HDF5 format to store both datasets and model snapshots, making it easy to interact with Matlab, Python (numpy) and other existing computational tools. Mocha also provides tools to import trained model snapshots from Caffe.
- **Correctness**: the computational components in Mocha in all backends are extensively covered by unit tests.
- **Open Source**: Mocha is licensed under the MIT "Expat" License. (pluskid, 2017).

The ecosystem is still young (653 Julia packages vs. 66,687 PyPI packages). The core language is still evolving – current v0.7.0-DEV introduced a lot of breaking changes (and also exciting new features) (Julia, 2017).

1.1.4 Julia Programming Language

Julia is a rising star in the world of technical computing, having inherited the best features of both Python, with most of its syntax and ease-of-use, and C in terms of fast execution. First released in 2012, Julia is free under the MIT licence and equipped with much built-in functionality and has entered the top 50 languages (TIOBE, 2017). As the time of documenting this thesis, the latest version is 0.6.0 and is the same version used during the implementation of this thesis. It contains a built-in package manager similar to R which allows easy installations (e.g. Pkg.add("HDF5")), removal of third-party packages (with 1,518 registered packages as at October, 2017) or even update them (Julia, 2017). Packaged with Basic Linear Algebra Subroutines, Julia is parallel when doing large-scale linear algebra tasks. In fact Julia is just as good for technical computations. Addressing challenges has not been an issue with Julia because are a number of ways to address them. “One popular approach has been to build sophisticated libraries for deep learning” (Juliacomputing, 2017). MXNet, Caffe, Torch, Theano and TensorFlow are some examples of this approach. Julia is thought to be the future of all deep learning applications with the benchmarks of Knet.jl (Yuret, 2017) against other frameworks for various models (JuliaCon, 2017).

1.1.5 Tomato Fruit

Coming from the Nahuatl word “*tomato”*, tomato literally means “the swelling fruit”. Tomato belongs to the family Solanaceae. It is one of the most important vegetables all over the world. As it is a relatively short-term crop but gives a high yield.it is economically attractive and contributes to healthy well-being. It provides minerals, vitamins, essential amino acids, sugars, dietary fibres, vitamin B and C, iron and phosphorus (Ugonna, Jolaoso, & Onwualu, 2015). Tomatoes can be processed into different products including: Ketchup, puree, powder and juice, but most of the fruit is sold fresh. The fruit is used as a fresh salad vegetable, and is also a popular ingredient in soups, stews, sauces and various other dishes as well.

1.2 Motivation and Research Objectives:

Having objects classified into categories is an important issue these days in many fields, these categorizations might have been done manually or using an automation system. Studies have shown that there are good varieties of tomatoes, but only a few are suitable for industrial processing with regard to quantity and quality. In this thesis we applied deep learning methods to extract useful patterns automatically from already labelled (classified or graded) images of tomatoes with a view of identifying unlabelled ones. Ordinarily, the grading (classification) simply consists of sorting the tomatoes into a number of uniform categories in our case nine categories according to the economically important physical characteristics and qualities. Grading of tomatoes is carried out because uniformity is one of the first attributes buyers look for. The appearance of certain category attracts customers and different qualities are sold to different customers, while the standards will create customer confidence in the product and more importantly in the producer (Ugonna, Jolaoso, & Onwualu, 2015).

Our objectives are (1) to build a deep NN DNN using a TensorFlow wrapper package with Julia programming language. We shall train the NN built using already labelled dataset of nine different classes termed as grades; and (2) to extract meaningful **patterns**, which will perform as closely as possible to the original DNN.

To realize the goal of this thesis, the following research questions is be put into query: how can a deep learning algorithm learn to generalize learnt features from the pre-labelled training data? (Mitchell, 1997) defines what it makes to make a computer learn, “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” An NN is a function that learns the expected output for a given input from training datasets by determining parameters (weights and bias) by itself.

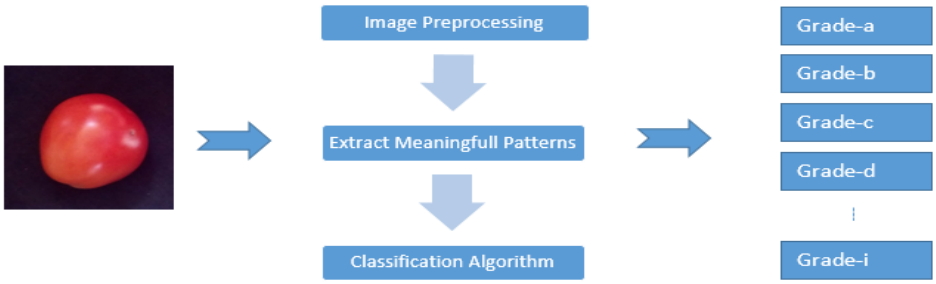


Figure 1.1: A Concise Pipeline of the thesis

Given an image of a tomato with its pixels data, the basic objective is to predict this image as being a member of a particular set of grades (grade-a, grade-b, grade-c... grade-i) based on the patterns previously learnt from the labelled data. In addition, we focus our attention in this thesis towards nine classes of grades, from grade-a down to grade-i with features such as size, colour and texture, which the algorithm learns automatically by itself.

1.3 The Scope of the Research Work

To limit the scope of the thesis, Julia programming language and Julia Mocha package was used. We do not apply localizations, detections or segmentations (label with an outline) of the objects in questions. No preprocessing algorithm was applied to the data before subjecting them to the classifier. The classifications was not based on formal method but rather farmers' and marketers' methods were used.

1.4 Importance

Processors of tomatoes require good quality ones for processing (Ugonna, Jolaoso, & Onwualu, 2015). The research will to enable them acquire modern state-of-the-art methods to automate grading for speedy production, processing and marketing of good quality tomatoes. The research will help improve well-being of citizens by encouraging consumption of healthy farm produce. It can help governments impose laws to ensurenquality standards for exports. The thesis also

opens a door for agricultural research institutions to enquire for more studies particularly on improved quality products and seeds as well as encouraging the use of modern enhancements. The research opens a door for re-activation of the companies that have closed down due to lack of raw materials and processing equipment, and opening of more processing industries in the country. It will also encourage a partial or total ban on imports of tomato products.

CHAPTER TWO
REVIEW OF RELATED LITERATURE

This chapter is dedicated to an overview of the requisite knowledge of deep learning, image classification and its applications, concepts and the mathematical techniques needed to develop the thesis and works related to this research. The text presented here also relate to the past, present and future outlook of ML as a field. There are many applications of ML. ML can be supervised learning where the instances are given with known labels (i.e. the corresponding correct label). Another method is called unsupervised (clustering) learning where instances are unlabelled. In this sense, useful classes of objects are sought to be discovered. The third class of ML is called reinforcement learning where the training information is provided to learning systems by external trainers or environments. The information is in the form of a scalar reinforcement signal which contains a measure of how well or badly the system operates. Our concentrations on this work will be based on supervised techniques on classification problems.

2.1 Deep Learning

In recent years, deep leaning has opened a new area of research for it has proved its outstanding performance in ML and pattern recognition. (Deng & D., 2013) defines deep learning as a class of ML techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification. Deep Learning was introduced with the objective of moving ML towards to one of its original goals, AI. Deep learning evolved from the acquisition of big data, the power of parallel and distributed computing, and sophisticated algorithms has in multitude ways facilitated major advances in domains such as image recognition, speech recognition, and natural language processing where the AI community struggled for many years (LeCun, Bengio, & Hinton, 2015). Deep learning is about learning multiple levels of representation and abstraction which help to make sense of

images, sound and text data. In recent years, deep artificial neural networks (including recurrent ones) have won numerous contests in pattern recognition and ML. (Min, Lee, & Yoon, 2017) Figure 2.2 briefly and diagrammatically describes deep learning:

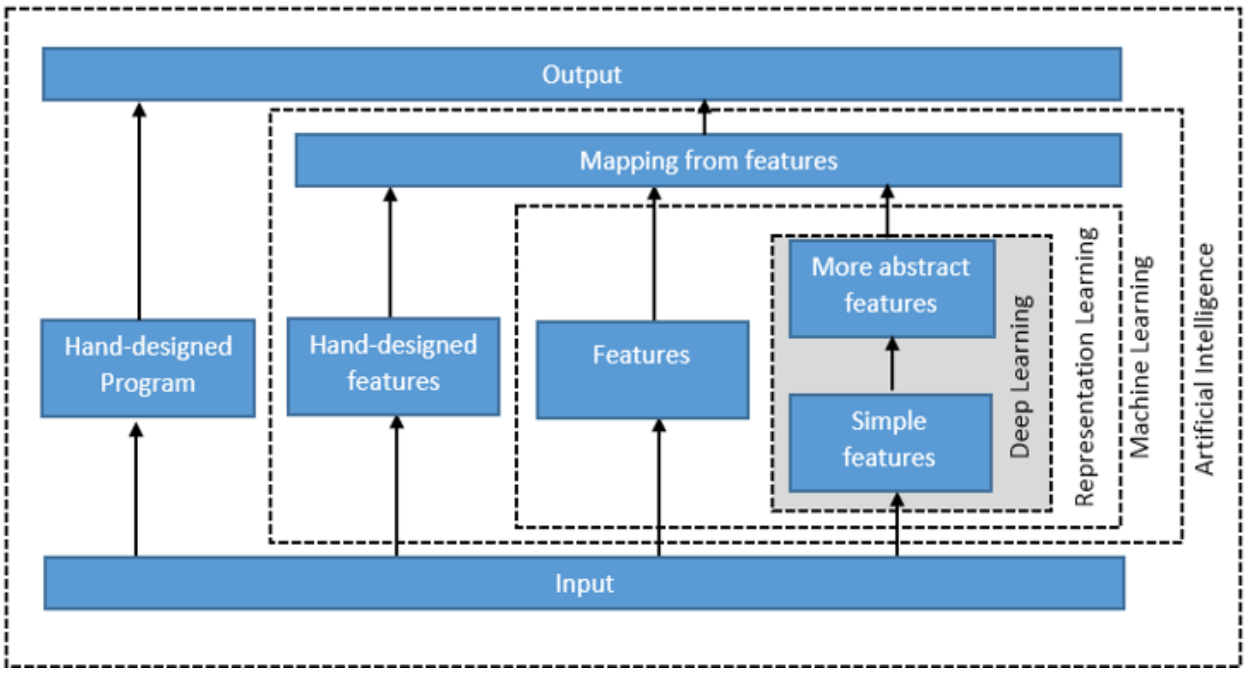


Figure 2.2: General Method of Machine Learning

Deep learning methods work towards learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of features at a lower level (Bengio, 2009). Spontaneous learning features at multiple levels of abstraction explained above allow a system to learn complex functions mapping the input to the output directly from data without depending completely on human-crafted features (Bengio, 2009).This is specifically important for higher-level abstractions, which humans often do not know how to specify explicitly in terms of raw sensory input. The depth of architecture is the number of levels of composition of non-linear operations in the function learnt. As the amount of data and range of applications to ML methods continues to grow, the ability to learn by spontaneously powerful features becomes increasingly important. There are three important reasons for the popularity of deep learning today are:

1. The drastically increased very high performance graphical processing units (GPGPUs) and tensor processing units (TPUs);
2. The significantly increased amount of data used for training; and
3. The recent advances in ML and signal/information processing research.

Deep learning is positioned in the intersections among the research areas of NNs, AI, graphical modeling, optimization, pattern recognition, and signal processing (Deng & D., 2013).

2.2 Feature Extractions

Features can be broadly categorized into continuous, categorical or binary. ML applications for computer images are commonly divided into three tasks: extraction, selection, and classification (Arif Mohamad, Nasien, Hassan, & Haron, 2015). Of all these three activities, feature extraction is most critical because the particular features made available for discrimination directly influence the efficacy of the classification or recognition task (Choras, 2007). There are many methods by which features are extracted from an input image, but the end result of the extraction task is a set of features, commonly called a feature vector, which constitutes a representation of the image. To distinguish images in the history of image analysis various colour spaces are used. And there are different type of colour spaces which include RGB, LUV, HSV and HMMD are based on extracting the mean, skewedness and standard deviation of intensity of the image pixel (Pachouri, 2015). Colour histograms, coherence vector, moments based and correlogram are used for the extraction of features in images.

The advantage of a colour histogram being successful when it comes to implementation is that it is very fast in detecting colour distribution features in any given images meeting basic requirements, but fails in matching a large set of images with inconsistency, and inaccuracy (Jinxia & Yuehong, 2011). Krishnan (2017) stated that out of these methods colour Moment is the

best in terms of simplicity, compactness and robustness in its technique of extracting features. Features are relevant information that can help identify an object or entity. Features from particular input data may be redundant, less important or too much to process at once. Then the input data is usually transformed into a reduced representation set of features from that original data called features vector. The act of transforming the input data into the set of features is called feature extraction. Feature extraction describes the relevant shape information represented in a pattern so that the task of classifying the pattern will be made easy or even easier by a formal procedure. In the literature, there are many ways features are extracted from input data, but the main goal of feature extraction is to obtain the most relevant information that is usually useful for classifications, from the original input data and then in a lower dimensionality space. In an area such as pattern recognition and image processing, feature extraction is a special form of dimensionality reduction. Good feature sets contain discriminating information i.e. the one that can distinguish one object from other ones. It must be as robust enough to prevent generating different feature codes for the objects in the same class. It is recommended that the selected set of features should be a small set, but should contain values which efficiently discriminate among patterns of different classes, but similar for patterns within a class. Features are classified into two categories:

1. Local (sometimes called low level) features, are usually small details of the image like point, curve or edges, etc. This feature can be extracted automatically from an image without knowledge of shape.
2. Global features (high-level). High-Level feature are built on top of low-level features to detect objects and larger shapes in the image (Krishnan, 2017).

If the features extracted are carefully chosen, then it is expected that the feature set will contain the relevant information from the input data in order to perform the desired task using this reduced

representation instead of the full size input; hence, computation power will be lowered. In the past many data representation learning methods have been proposed. In order to obtain low dimensional representations of data with a linear projection, the earliest data representation learning algorithms principal component analysis (PCA) and linear discriminant analysis (LDA) were both proposed. PCA is unsupervised method proposed by K. Pearson (Pearson, 1901), while LDA is a supervised method proposed by (Fisher, 1936). Variety of extensions has been proposed, such as kernel PCA (Scholkopf, A., & Muller, 1998) and (Baudat & Anouar, 2000) generalized discriminant analysis based on the PCA and the LDA (Zhong, Wang, Ling, & Dong, 2016).

Feature extraction stages in many recent object recognition systems generally composed of a filter bank, which is a non-linear transformation and some sort of feature pooling layer. Most systems use just one stage of feature extraction in which the filters are hard-wired, or in the other case two stages where the filters in one or both stages are learnt in supervised (LeCun, Bengio, & Hinton, 2015) or unsupervised mode. Jarrett, Kavukcuoglu, Ranzato, and LeCun (2099), stated that many of the recent proposals use dense features extracted on regularly-spaced patches over the input. These systems use a feature extraction process composed of a filter bank ordinarily based on oriented edge detectors, a non-linear operation (quantization, winner-take-all, sparsification, normalization, and/or point-wise saturation), and also a pooling operation that combines nearby values in real space or feature space through a max, average or histogram operator. Other models use two or more successive stages of such feature extractors, followed by a supervised classifier.

In 2000, the ML community launched the research on manifold learning, which is to discover the intrinsic structure of high dimensional data (Zhong, Wang, Ling, & Dong, 2016) with an approach that is different from previous global approaches, such as PCA and LDA, manifold learning

methods are generally locality based, such as isometric feature mapping (Isomap) and locally linear embedding. The figure illustrates the trends of data representations (Arif Mohamad, Nasien, Hassan, & Haron, 2015):

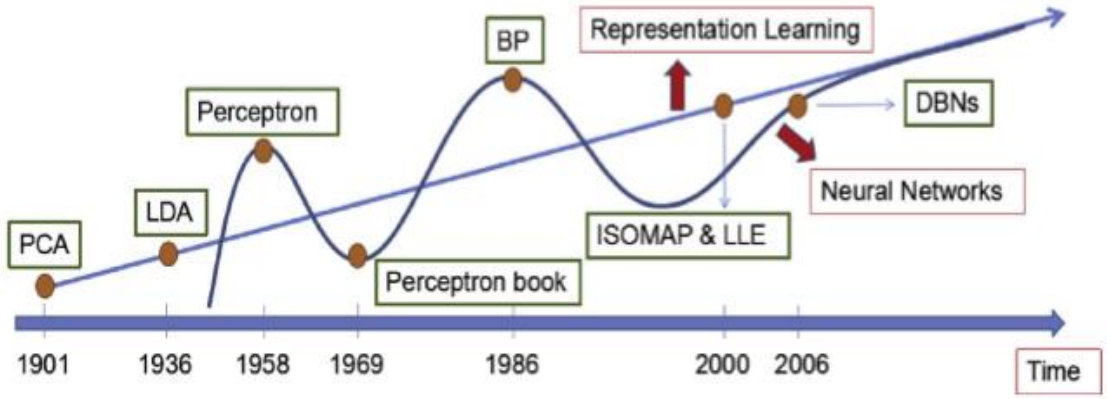


Figure 2.3: The Development of Data Representation Learning and Neural Networks

Gabor filter is widely used to extract the texture feature for image classification. A fundamental operation in image processing represents the filtering of intensity images. Thus such a filtering technique is used for computing derivative filters of first order for motion estimating and edge detection, and second or higher order for feature extraction of curvature information. Making a suitable design choice leads to the formulation of an optimization problem, yielding the appropriate filter kernels. “These filters are discretized by finite differences using convolution kernels optimized with respect to the model assumptions, scales and/or noise present in the data” (H.G. Bock et al. (eds.), 2013). Gabor filter or wavelets characterize an image by obtaining the centre frequency as well as the orientation parameter. Shape feature extraction methods can be classified into two groups as contour-based and region-based methods. Contour-based technique calculate shape feature only from the boundary and region-based method extracts feature from the entire region. Gabor filters are computationally expensive but due to a high dimension of the feature vector the results obtained from them are robust. The Gabor techniques are widely used in NN. The Gaussian filtering or function has important applications in many areas of

mathematics, as well as image filtering. In this section, we highlight the characteristics that make it useful for smoothing images or detecting edges after smoothing.

Neocognitron (Fukushima, 1980) is regarded as perhaps the first artificial neural network that deserved the attribute deep. Convolutional neural networks (or convnets) stand out for expanding the applicability of artificial neural networks (ANNs) domain. So, today's convolutional NNs took their inspiration from LeCun, Bottou, Bengio, and Haffnes' (1998) well-known research paper. LeCun, Bottou, Bengio, and Haffnes (1998), proposed an NN architecture they termed LeNet-5 which was applied to MNIST (Rodolfo, 2016) dataset (a dataset of handwritten characters) which scored 99.2% accuracy. *"Convolutional networks can be seen as synthesizing their own feature extractor.* " (LeCun, Bottou, Bengio, & Haffner, 1998). To understand how the convolution works, we first explore the origin of the convolution function, and thence we explain how the concept is applied to the information.

2.2.1 Continuous Convolution

The original use of this function comes from the eighteenth century and can be expressed, in the original application context, as an operation that blends two functions occurring on time. Mathematically, it can be defined as follows:

$$(f * g)(\tau) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau$$

Algorithmically, we can conceptualize this operation as in the following steps:

Step 1: Flip the signal: This is the *(-τ)* part of the variable.

Step 2: Shift it: This is given by the *t* summing factor for *g(τ)*.

Step 3: Multiply it: This is the product of *f* and *g*.

Step 4: Integrate the resulting curve: This is the less intuitive part because each instantaneous value is the result of an integral.

2.2.2 Discrete Convolution

The convolution can be translated into a discrete domain and described in discrete terms for discrete functions:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f(m)\, g(n-m)$$

2.2.3 Convolutional Neural Network Concepts

Convolutional Neural Networks (CNNs) have an associated terminology and a set of concepts that is unique to them, and that sets them apart from other types of neural network architectures. The main ones are explained as follows:

2.2.3.1 Input/Output Volumes:

CNNs are usually applied to image data. Conventionally, any image is a matrix of pixel values (Width x Height). The range of values that can be encoded in each pixel depends upon its bit size (8 bit or 1 byte-sized are commonly used) pixels with a range [0, 255] for each pixel value. The coloured images, however, particularly RGB (i.e. red, green and blue) have additional attributes 'depth' of separate colour channels, making the input three-dimensional. An RGB image of say, 32 x 32 (width x height) pixels would have three matrices associated with each image, one for each of the colour channels. Thus the image in its entirety, constitutes a three-dimensional

structure called the input volume (32 x 32 x 3). An RGB image with the dimensions four-units width and four-units height can be represented pictorially as:

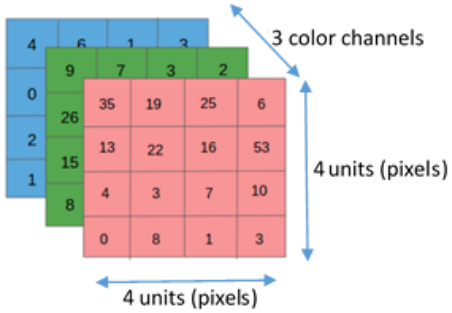


Figure 2.4: An RGB image with the dimensions 4-units width and 4-units height

2.2.3.2 Filters (Convolution Kernels)

A filter (or kernel) is an integral component of the layered architecture. It refers to an operator applied to the entirety of the image such that the information encoded in the pixels is transformed. In a real case scenario, however, a kernel is a smaller-sized matrix (measured *n x m*-dimensional matrices usually, *m* = *n*) in comparison to the original input dimensions of the image. The kernels are convolved together with the input volume to obtain what is called 'activation maps'. Activation maps indicate 'activated' regions. The convolution operation consists of multiplication (dot products) of the corresponding pixels in the input volume with the kernel, one pixel at a time, and summing the values for the purpose of assigning that value to the central pixel.

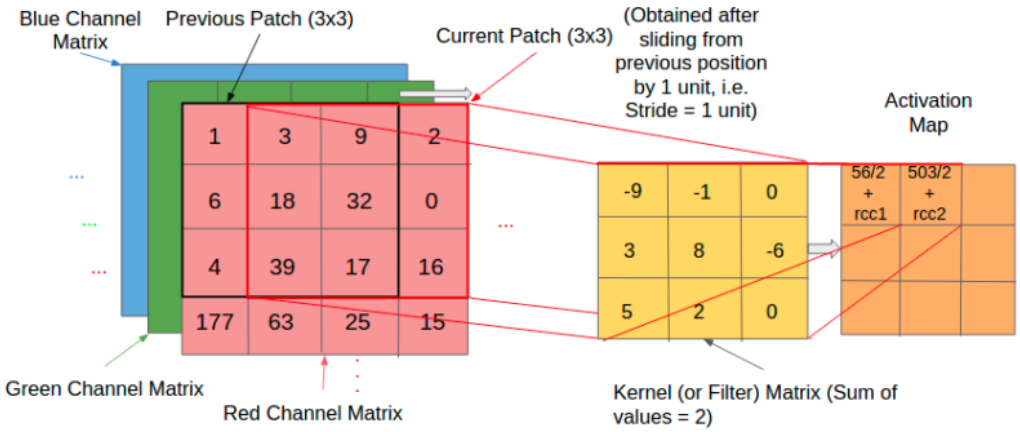


Figure 2.5: The Convolution Process with the Kernel or Filter

The process is an iterative one, and real values of the kernel matrix change with each learning iteration over the training set, indicating that the network is learning to identify which regions are of significance for extracting features from the input volume. **'**The convolution kernels highlight or hide patterns. Depending on the trained (or in the example, manually set) parameters, we can begin to discover parameters, such as orientation and edges in different dimensions' (Rodolfo, 2016, p. 208)

2.2.3.3 Receptive Field

In reality, it is impractical that the all neurons be connected with all possible regions of the input volume because there would be to too many weights to train, and would produce excessive computational complexity. Thus, instead of connecting each neuron to all possible pixels, a two-dimensional region called the 'receptive field' is specified (e.g. 5 × 5 units) extending to the entire depth of the input (5 x 5 x RGB). 'It's over these small regions that the network layer cross-sections (each consisting of several neurons (called 'depth columns')) operate and produce the activation map' (Abhineet, 2016).

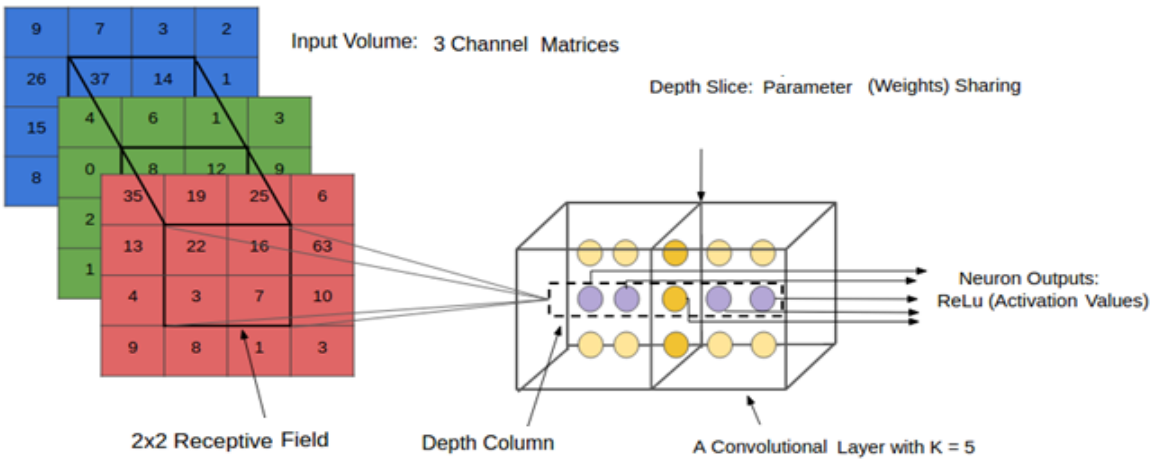


Figure 2.5: Conceptual Representation of Receptive Field

2.2.3.4 Zero Padding

Zero padding is associated with CNNs which refers to the process of symmetrically adding zeroes to the input matrix to maintain what is referred to as 'SAME', i.e. making adjustment or modification that allows the size of the input to be adjusted to a specific size. It is usually employed when the dimensions of the input volume need to be preserved in the output volume. Below is a physical representation of zero padding:

Figure 2.6: Representation of Zero Padding

2.2.3.5 Max-Pooling

An important concept of CNNs, called *max-pooling,* is a form of non-linear down-sampling. Max-pooling partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value. The reasons for using this technique are:

1. By eliminating non-maximal values, we then reduce computation for upper layers.
2. It provides a form of translation invariance. Imagine cascading a max-pooling layer with a convolutional layer (Mohamed, Yarub, & Mohamed, 2013).

The diagram shown below, is the LeNet-5 architecture, which was the first well-known convolutional architecture ever published regarding that problem. Here, can be seen the dimensions of the layers and the last result representation:

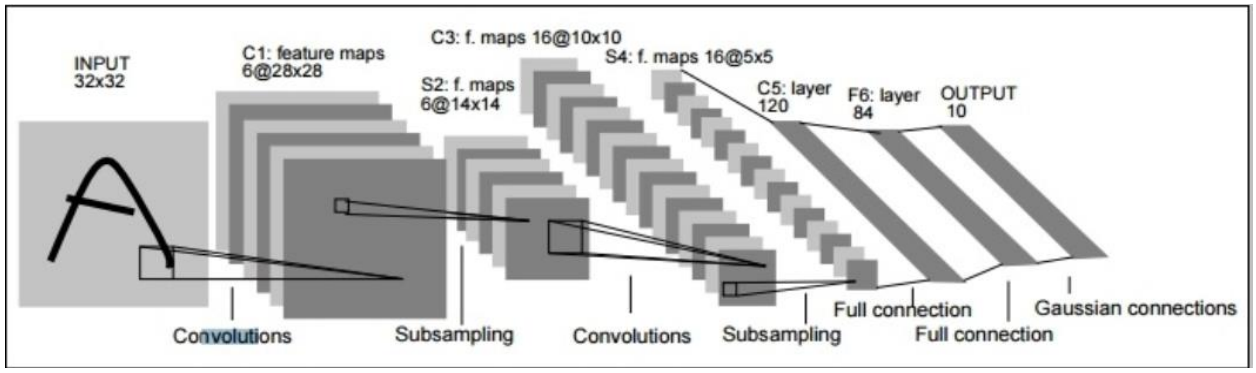


Figure 2.7: The LeNet-5 architecture

CHAPTER THREE
RELATED WORKS ON APPLICATION OF CLASSIFICATIONS

Classification is one the most commonly applied technique in ML or even computer vison. It is a technique that employs a set of pre-labelled examples used to develop a model that can predict the population of records of unseen data. Data classification processes involve learning and classifications. During learning, the training data is in some cases preprocessed and analyzed by a classification algorithm. On the other hand in a prediction test, data is used to estimate the accuracy of the learnt rules. There exist different types of classification models which are applicable to the classifications problems. These include Bayesian Classification, NNs, Support Vector Machines (SVM). In this chapter we explore different applications of these. The general concept is seen in Figure 3.1:

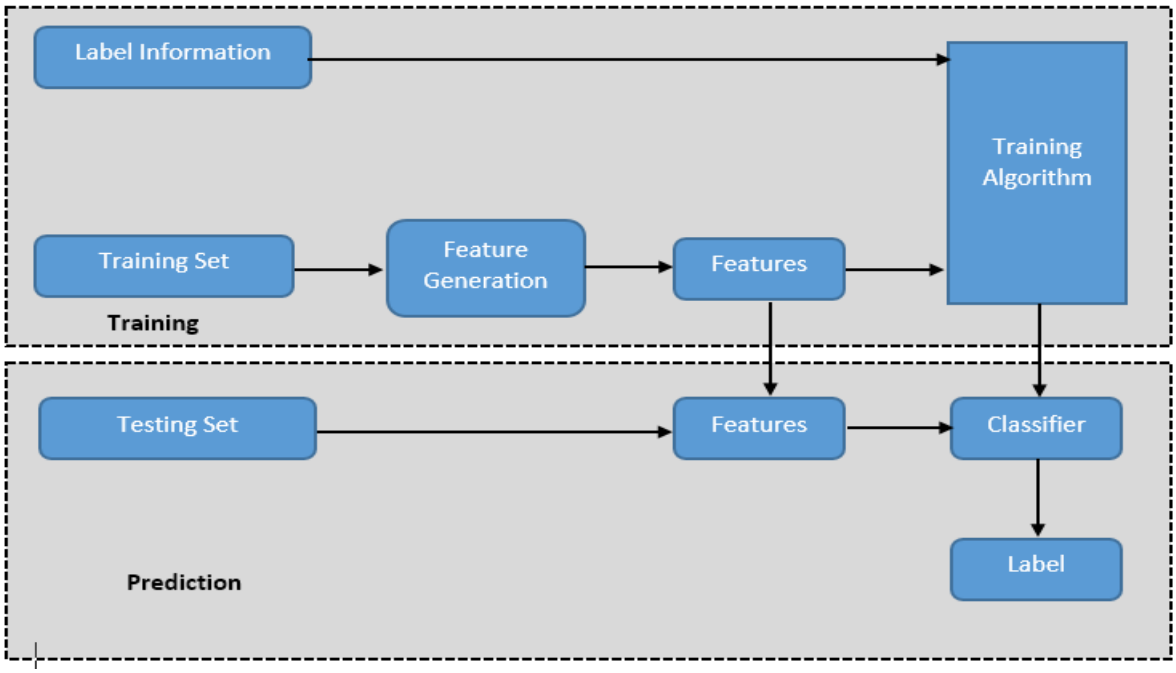


Figure 3.1: A General Process of Data Classification

A general process of data classification is shown in Figure 3.1, which in most cases consists of two phases – the training and the prediction (or inference) phases. Within the training phase, data

is used and analyzed into a set of features based on the feature generation models such as the vector space model for textual data and fine-tuned to create a predictive model, while during the prediction set untrained data is channelled to the model for prediction. There are many applications of ML classifications in our daily lives, including agriculture (Karandeep, 2016). Because humans are prone to making errors when manually grading, identifying or classifying objects which will uncontrollably lead to misclassification, (Opeña & Yusiong, 2017) introduced an automated tomato classification system that uses ANN classifier trained using the Artificial Bee Colony (ABC) algorithm. Opeña & Yusiong's (2017) results have shown that the ABC-trained ANN classifiers performed well in tomato classification and achieved a high accuracy rate which can replace manual classification methods, minimizing the chances of misclassification. However, this method is dependent on colour factor to classify tomatoes based on maturity. This is due to fact that some researchers give emphasis to classifying tomatoes on the basis of maturity, considering colour as the only crucial factor in estimating the ripeness. Opeña & Yusiong (2017), stated that the colour model is selected from several colour models such as RGB, HSI, HSV, CIE L*a*b*, and YCbCr to generate a feature vector for each tomato image. On the other hand, some researchers suggested the use of multi-valued feature representations with several features such as colour, texture, size and shape of the image. El-Bendary (2014),chose the HSV colour model for the feature extraction step proposing PCA in addition to SVMs and LDA algorithms for feature extraction and classification. Zhang & McCarthy (2012), applied magnetic resonance imaging to evaluate tomatoes for the proposed approach. MR images were captured from harvested tomatoes at different maturity stages. Then, for each of the MR images, the mean and histogram features of the voxel intensities in the region of interest were calculated. At the final stage, the partial least square discriminant analysis (PLS-DA) algorithm was applied to deduce a maturity classification model. Baltazar, Aranda and González-Aguilar (2008), used 128 tomato samples that were harvested and sorted with a colorimeter, choosing only those with roughly breaker colour. This was used to represent the ripeness stage where there was a definite break in colour

from green to tannish-yellow. They applied data fusion to non-destructive image of fresh intact tomatoes by assessing both colorimeter and nondestructive firmness measurements for the samples at the same particular testing days using two sensors placed at different points. Then the measurement data was normalized. A three-class Bayesian classifier was applied. The results showed that multi-sensorial data fusion is better than single sensor data and considerably reduces the classification error. Zhang, Lee, Tippetts and Lillywhite (2014), classified harvested dates according to the colour of the fruits. After capturing of images, a threshold segmentation procedure was applied to remove fruit from its background. From RGB colour space, only R–G plane was used, as blue-colour channel does not give reliable information for date grading. The training phase was used to generate a 2D histogram, one for each maturity class. The co-occurrence of every colour in the R and G channel in each class was then counted. After 2D histogram generation, it was then normalized. There followed back projection matrix generation. For the grading phase, after removal of the background and extraction of R-G values, a back projection step followed by colour index analysis was performed; then colour grading was computed using statistics. The aforementioned approach achieved good results without requiring a complicated training process and ML algorithms. May and Amaran (2011) proposed an approach using automated ripeness assessment using RGB and fuzzy logic feature extraction and classification model to assess the ripeness of palm oil. Palm ripeness stages were classified into three classes, under-ripe, ripe and overripe depending on different colour intensity. It depended on colour intensity and achieved an accuracy of up to 88.74%. Polder, Heijden and Young (2002), implemented the LDA algorithm to discriminate tomatoes depending based on individual pixels and included gray reference in each image for obtaining automatic compensation of different light sources. The approach is based on spectral images analysis to measure the ripeness of tomatoes for automatic discrimination. The approach proved that spectral image representations are better than standard RGB image representations for measuring ripeness stages of tomatoes by offering more classification power. Guerrero and Benavides (2014),

proposed a classification system using Fisher's LDA and K-means algorithms. RGB colour space was used and applied filter technique to minimize noises in the image. Fisher's LDA algorithm was used to separate avocado fruit from background. Based on based on pixels percentage, K-means grouping technique was applied to make classification, purposely discrimination highly matured from matured ones with the accuracy of 87.85% was attained. Fadilah and Mohamad-Saleh (2014), applied the ANN technique to classify oil palm fresh bunch ripeness into four categories based on colour features, thus ripe, overripe un-ripe and under-ripe. Fadilah and Mohamad-Saleh (2014), used the colour of oil palm fresh bunches as a ripeness indicator. Applying image segmentation technique using K-means clustering algorithm to discriminate fruits pixels from spikes ones. A hue histogram of bins was then extracted for each image as a feature vector applying two different techniques, namely PCA and stepwise discriminant analysis (SDA) for colour feature reduction purposes. The results showed that improved performance by more than 10% is achievable by reducing the colour features using SDA. While most of the previously mentioned used small datasets to their respective classifiers, Mohammed, Kamel, and Abdelouahab (2017), applied the CNN as a learning algorithm to a comparably larger dataset containing 14,828 images of tomato leaves infected with nine different diseases to train their model. Mohammed, Kamel and Abdelouahab (2017), took the advantage of CNN to automatically extract features from raw images. The trained model achieved an encouraging result of 99.18% accuracy.

In summary, we have presented different approaches centred on deep learning, digesting different methods of extracting features such as PCA (El-Bendary, 2014), as well as classification and applications. The following studies revealed that having a good feature representations led to a higher performance: Baltazar, Aranda, an González-Aguila (2008); Fadilah and Mohamad-Saleh (2014); Fadilah and Mohamad-Saleh (2014) and Guerrero & Benavides, 2014). However, having a larger dataset (Mohammed, Kamel, & Abdelouahab, 2017) together with deep neural

net technique, specifically convolutional neural nets, is capable of extracting features with an accuracy of more than 99%.

Convolutional NNs are one the successful of many of the most advanced models currently being employed today. They are used in numerous fields, but the main application field is in the realm of image classification, text analysis and feature detection. Google uses convnets in most of their products.

CHAPTER FOUR
RESEARCH METHODOLOGY

4.1 Deep Learning Toolkits / Libraries and Architecture

4.1.1 Mocha Package

Written in Julia and for Julia: easily making use of data pre/post processing and visualization tools from Julia:

- Minimum dependency: Julia backend ready to run, easy for fast prototyping.
- Multiple back ends: easily switching to CUDA + cuDNN based backend for highly efficient deep nets training.
- Correctness: all computation layers are unit-tested.
- Modular architecture: layers, activation functions, network topology, etc. Easily extendable.

4.2 Dataset Compilation

4.2.1 Collection of Data

About one thousand images of tomatoes were taken from the major tomatoes markets; Kwanar Gafan situated in Kano State of Nigeria, Gada in Jigawa State of Nigeria, and minor markets such as Kasuwar Kul-Kul, Jakara market and Bakin Asibiti of Kano State. Each image had a dimension of 3840X2160 pixels snap-shot on black background as shown below.

Figure 4.1: Sample from Images of Tomato Taken at Kwanar Gafan

4.2.2 Data Preparation

Before the data was subjected to manual discrimination, the images were cropped in to steps:

Step 1: cropping from 3840X2160 pixels dimension to 800X800 pixels dimension.

Step 2: cropping from 800X800 pixels dimension to 32X32 pixels dimension.

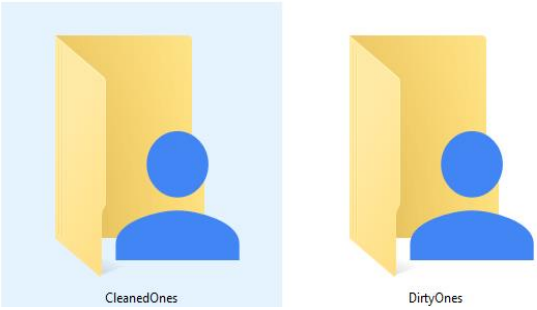


Figure 4.1: Segregations of images of tomatoes

The cropping of the images was done for the removal of unwanted and redundant background from the data.

Data preparations has three steps:

4.2.2.1 Data Deduplication

In this step, duplicates data were removed from our dataset. The following figure shows some images:

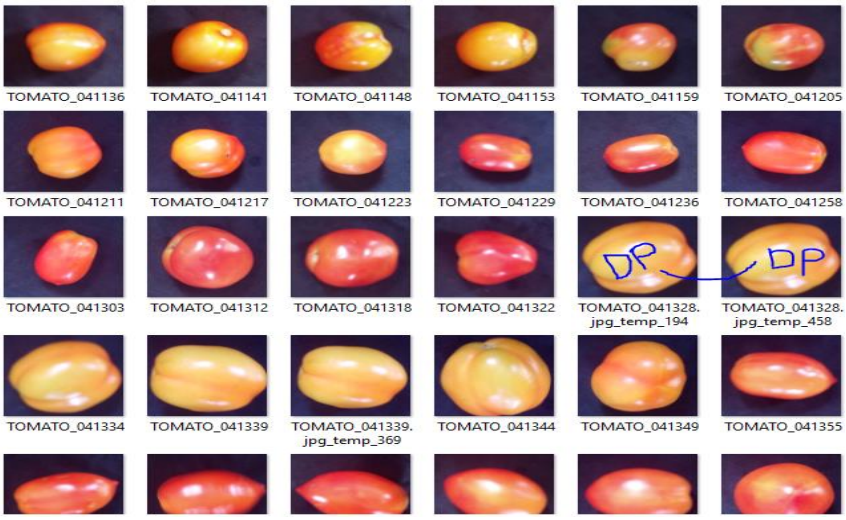


Figure 4.2: Duplicate Cropped tomato Images

4.2.2.2 Data Correction

No corrections were made to the images (darkening, whitening, etc.).

4.2.2.3 Data Discrimination (manual)

The segregations were made based on the following features for importance for the training:

4.2.2.4 Colour

The colour feature is one of the most widely used visual features in image retrieval. Images characterized by colour features have many advantages such as robustness, effectiveness, implementation simplicity, computational simplicity and low storage requirements;

- Robustness. The colour histogram is invariant to rotation of the image on the view axis, and changes in small steps when rotated otherwise or scaled. It is also insensitive to changes in image and histogram resolution and occlusion.
- Effectiveness. There is high percentage of relevance between the query image and the extracted matching images.
- Implementation simplicity: The construction of the colour histogram is a straightforward process, including scanning the image, assigning colour values to the resolution of the histogram, and building the histogram using colour, and components as indices.
- Computational simplicity: The histogram computation has O(X, Y) complexity for images of size X × Y. The complexity for a single image match is linear, O (n), where n represents the number of different colours, or resolution of the histogram.
- Low storage requirements: The colour histogram size is significantly smaller than the image itself, assuming colour quantization.

Typically, the colour of an image is represented through some colour model. There exist various colour models to describe colour information. A colour model is specified in terms of 3-D coordinate system and a subspace within that system where each colour is represented by a single point. The more commonly used colour models are *RGB* (red, green, blue), *HSV* (hue, saturation, value) and Y, Cb, Cr (luminance and chrominance). Thus the colour content is characterized by three channels from some colour model. One representation of colour content of the image is by using a colour histogram. Statistically, it denotes the joint probability of the intensities of the three colour channels.

4.2.2.5 Texture

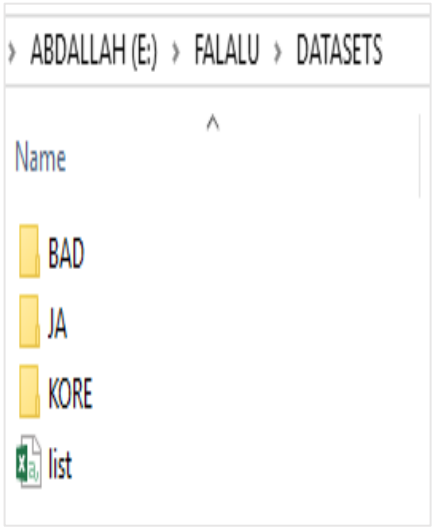
Texture is another important property of images. Texture is a powerful regional descriptor that helps in the retrieval process. Texture on its own does not have the capability of finding similar

images, but it can be used to classify textured images from non-textured ones and then be combined with another visual attributes like colour to make the retrieval more effective. Texture has been one of the most important characteristics which have been used to classify and recognize objects and have been used in finding similarities between images in multimedia databases. Basically, texture representation methods can be classified into two categories, structural and statistical. Statistical methods, including Fourier power spectra, co-occurrence matrices, shift-invariant principal component analysis, Tamura features, Wold decomposition, Markov random field, fractal model and multi-resolution filtering techniques such as Gabor and wavelet transform, characterize texture by the statistical distribution of the image intensity.

4.2.2.6 Shape/Size:

Shape based image retrieval is the measuring of similarity between shapes represented by their features. Shape is an important visual feature and it is one of the primitive features for image content description. Shape content description is difficult to define because measuring the similarity between shapes is difficult. Therefore, two steps are essential in shape-based image retrieval. They are feature extraction and similarity measurement between the extracted features. Shape descriptors can be divided into two main categories, region-based and contour-based methods. Region-based methods use the whole area of an object for shape description, while contour-based methods use only the information present in the contour of an object.

The images were first sorted into three different directories by colour. Each was then sub-sorted into by size and finally by texture.



	file_name	label	color_justification	size_justification	texture_justification
2	TOMATO_040759.jpg	grade-g	light	large	good
3	TOMATO_040857.jpg	grade-a	dark	large	good
4	TOMATO_040905.jpg	grade-b	dark	intermediate	good
5	TOMATO_040911.jpg	grade-e	intermediate	intermediate	good
6	TOMATO_040955.jpg	grade-a	dark	large	good
7	TOMATO_041015.jpg	grade-b	dark	intermediate	good
8	TOMATO_041030.jpg	grade-e	intermediate	intermediate	good
9	TOMATO_041045.jpg	grade-b	dark	intermediate	good
10	TOMATO_041050.jpg	grade-e	intermediate	intermediate	good
11	TOMATO_041056.jpg	grade-e	intermediate	intermediate	good
12	TOMATO_041108.jpg	grade-e	intermediate	intermediate	good
13	TOMATO_041124.jpg	grade-c	dark	small	good
14	TOMATO_041131.jpg	grade-e	intermediate	intermediate	good
15	TOMATO_041136.jpg	grade-h	light	intermediate	good
16	TOMATO_041141.jpg	grade-h	light	intermediate	good

Figure 4.3: Dataset Representation

4.2.3 Instruments Used

1. A mobile phone was used throughout the image capturing process. As seen previously, the camera was capable of producing an image with 3840 X 2160 pixels.
2. Mass Image Compressor V3.
3. Bulk Rename Utility version 3.0.0.1.

Working with Mocha, need input data in the form of HDF5 format. We wrote a script that converted the images into binary file format. We then used convert.jl file to convert the generated.bin file to HDF5 file format.

4.3 Defining Network Architecture

In this project, we propose deep CNNs. In the previous chapter, we explore the structures and the building blocks of convolutional neural networks.

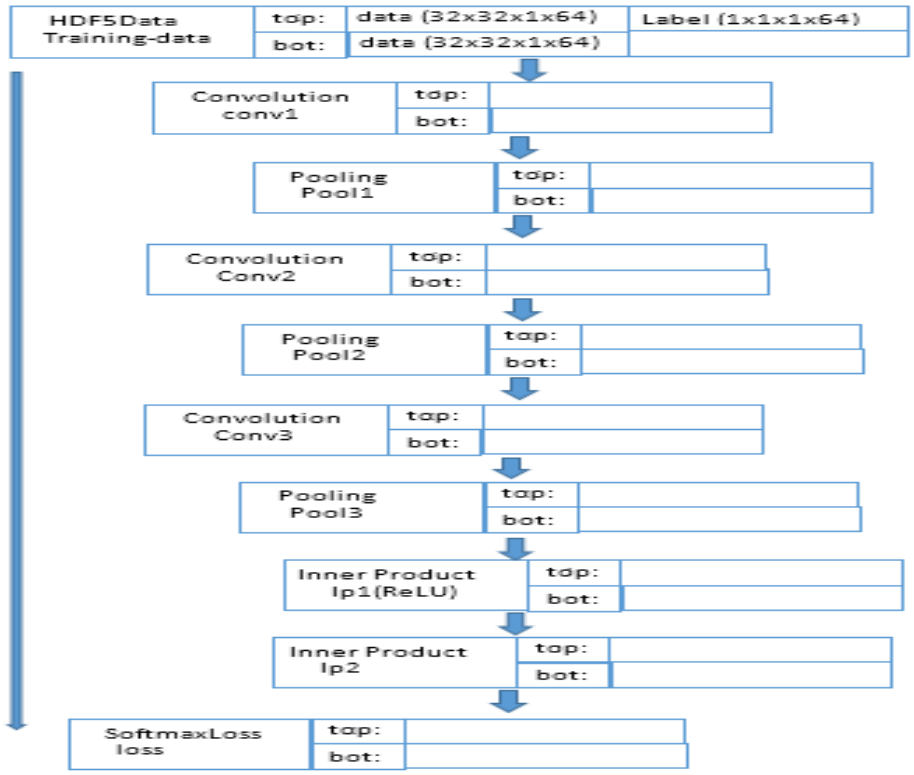
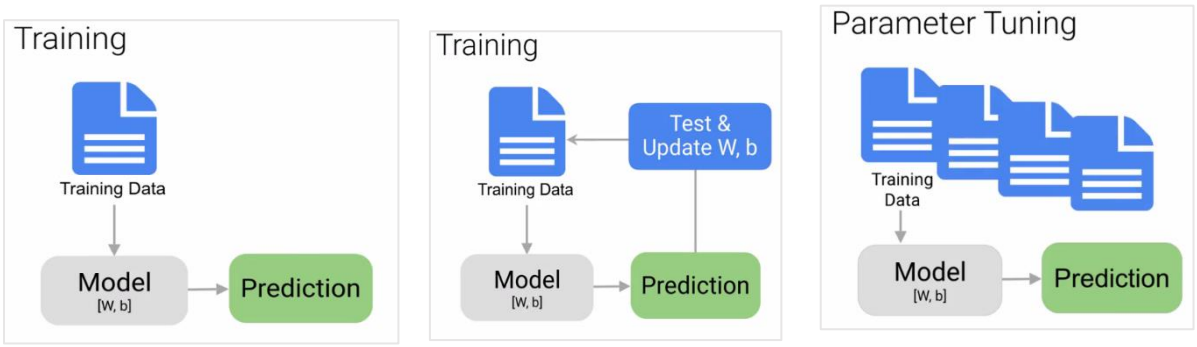


Figure 4.4: Neural Architecture Design

4.4 Training and Evaluation

First build the graph according to the specified architecture above and then secondly execute the



graph:

Figure 4.5: Training the Network

4.4.1 Evaluation

Using our trained model, we can make predictions. A way to accomplish this is to take the test data and run it through the model. This is done by setting the data of the MemoryDataLayer to the first test image and then using forward to execute the network, as the snippet below shows:

```
using HDF5
h5open("data/test.hdf5") do f
get_layer(run_net, "data").data[1][:,:,1,1] = f["data"][:,:,1,1]
println("Correct label index: ", Int64(f["label"][:,1][1]+1))
end
```

The figure below shows the evaluation processes:

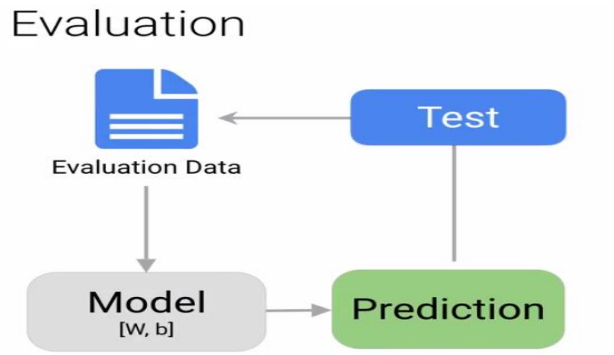
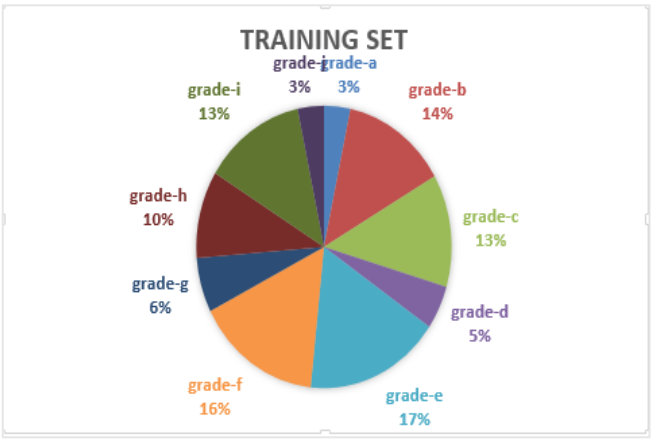


Figure 4.6: Evaluation of Learnt Parameters

CHAPTER FIVE
EXPERIMENT AND RESULTS

The dataset used in the experiments i contains exactly 300 images of tomatoes, out of which 240 were used for the training and 60 for testing or prediction – a 70:30 proportion. The effectiveness and efficiency for generalization of the model trained would be best evaluated using this dataset. The figures below illustrate the data used in details:



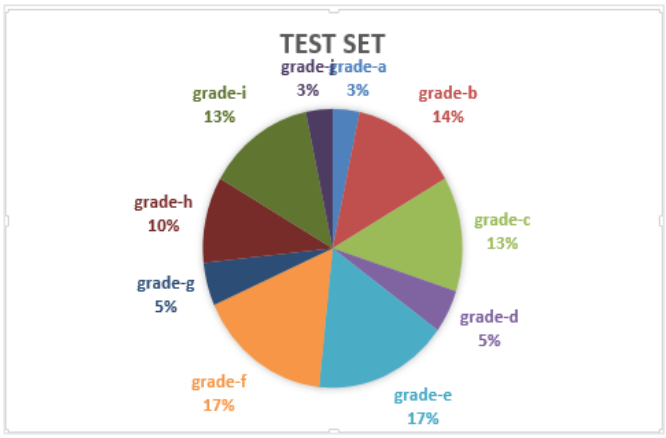


Figure 5.1: The Training Set

SN	GRADE	NUNBER OF SAMPLES	Training Set	Test Set
1	grade-a	10	8	2
2	grade-b	40	32	8
3	grade-c	39	31	8
4	grade-d	15	12	3
5	grade-e	51	41	10
6	grade-f	48	38	10
7	grade-g	18	15	3
8	grade-h	30	24	6
9	grade-i	39	31	8
10	grade-j	10	8	2
		TOTAL =	240	60

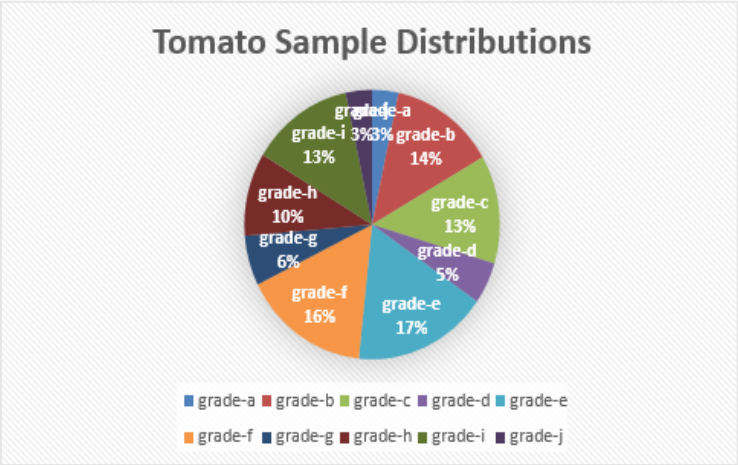


Figure 5.2: Training and Test Sets

```
[2017-11-19T18:56:39 | info | Mocha]:
[2017-11-19T18:56:39 | info | Mocha]: ## Performance on Validation Set after 0 iterations
[2017-11-19T18:56:39 | info | Mocha]: ---------------------------------------------------------
[2017-11-19T18:56:39 | info | Mocha]:   Accuracy (avg over 60) = 13.3333%
[2017-11-19T18:56:39 | info | Mocha]: ---------------------------------------------------------
[2017-11-19T18:56:39 | info | Mocha]:
[2017-11-19T18:56:54 | info | Mocha]:  TRAIN iter=000200 obj_val=2.25476789
[2017-11-19T18:57:03 | info | Mocha]:  TRAIN iter=000400 obj_val=2.30110502
[2017-11-19T18:57:15 | info | Mocha]:  TRAIN iter=000600 obj_val=2.09428883
[2017-11-19T18:57:24 | info | Mocha]:  TRAIN iter=000800 obj_val=2.64326382
[2017-11-19T18:57:33 | info | Mocha]:  TRAIN iter=001000 obj_val=2.20221877
[2017-11-19T18:57:36 | info | Mocha]:
[2017-11-19T18:57:36 | info | Mocha]: ## Performance on Validation Set after 1000 iterations
[2017-11-19T18:57:36 | info | Mocha]: ---------------------------------------------------------
[2017-11-19T18:57:36 | info | Mocha]:   Accuracy (avg over 60) = 35.0000%
[2017-11-19T18:57:36 | info | Mocha]: ---------------------------------------------------------
[2017-11-19T18:57:36 | info | Mocha]:
[2017-11-19T18:57:45 | info | Mocha]:  TRAIN iter=001200 obj_val=2.65905404
[2017-11-19T18:57:54 | info | Mocha]:  TRAIN iter=001400 obj_val=2.55462646
[2017-11-19T18:58:02 | info | Mocha]:  TRAIN iter=001600 obj_val=1.65938163
[2017-11-19T18:58:11 | info | Mocha]:  TRAIN iter=001800 obj_val=1.96202993
[2017-11-19T18:58:20 | info | Mocha]:  TRAIN iter=002000 obj_val=1.37201655
[2017-11-19T18:58:22 | info | Mocha]:
[2017-11-19T18:58:22 | info | Mocha]: ## Performance on Validation Set after 2000 iterations
[2017-11-19T18:58:22 | info | Mocha]: ---------------------------------------------------------
[2017-11-19T18:58:22 | info | Mocha]:   Accuracy (avg over 60) = 43.3333%
[2017-11-19T18:58:22 | info | Mocha]: ---------------------------------------------------------
```

Figure 5.3: A Screenshot of The Training Session of the First 2000 Iterations

Accuracy Graph shows how the network learning parameters learning progress as iterations increases.

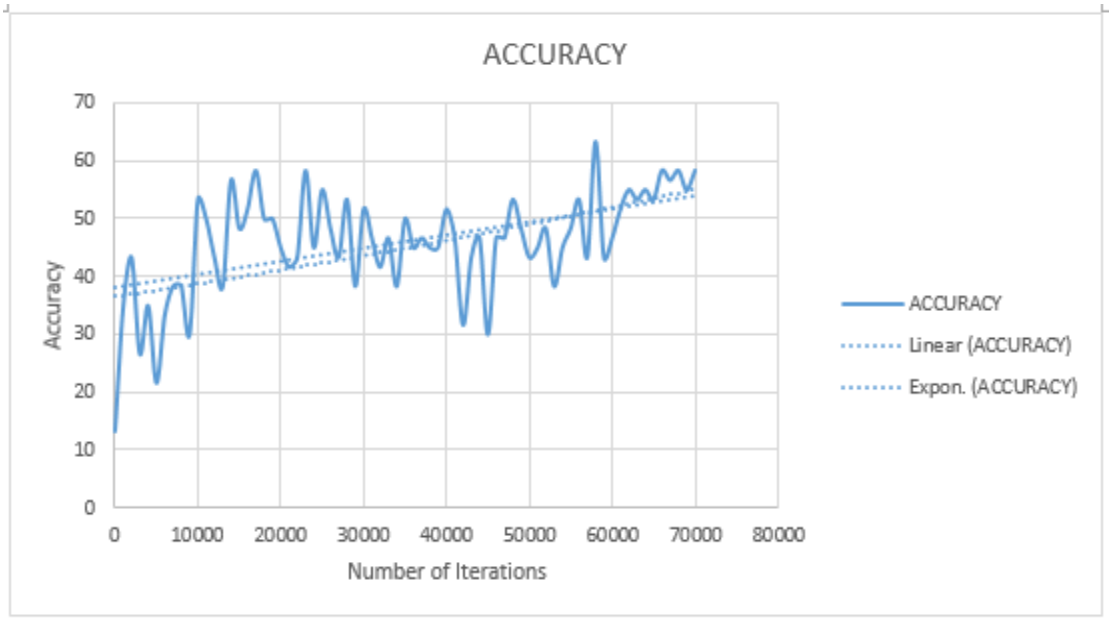


Figure 5.4: Accuracy Graph

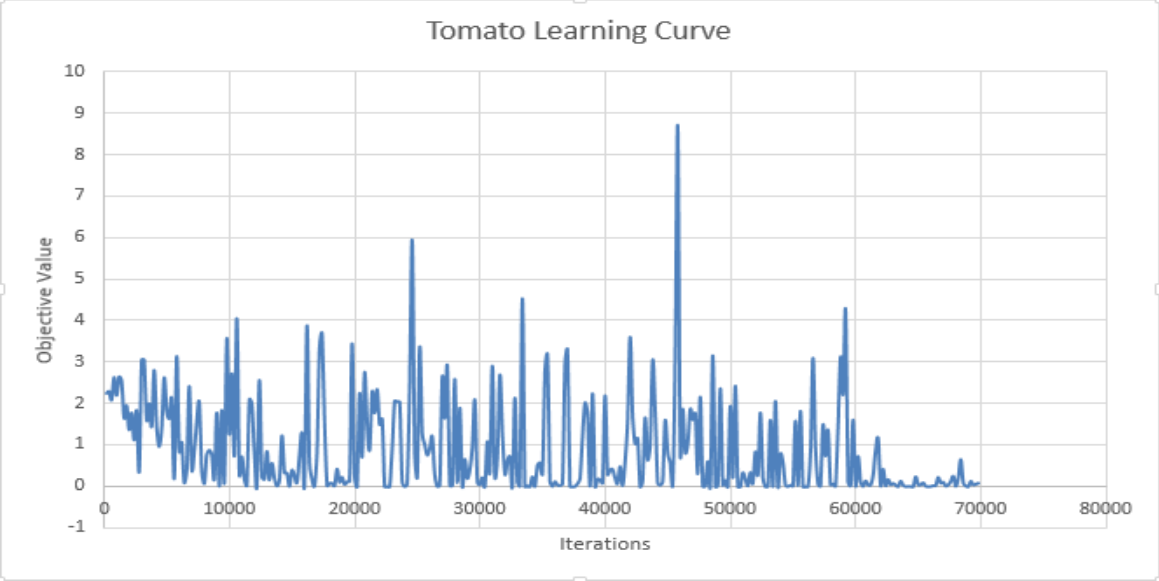


Figure 5.5: Learning Curve

While the network is training, we should verify that the optimization of the weights and biases is converging to a solution (i.e. learning takes palace). One of the best ways to do this is to plot the learning curve as the solver progresses through its iterations (0-70,000) in our case. A neural network's learning curve is a plot of iterations along the axis and the value of the objective function along the axis. Since the solver is trying to minimize the objective function so the value plotted along the axis should decrease over time.

Looking at the learning curve on the figure above after the first 1,000 iterations clearly shows that the algorithm is working and that letting it continue to train for the entire 70,000 iterations probably produces a good result. The data to plot the learning curve is conveniently saved as the solver progresses. We set up the coffee lounge and a TrainingSummary() coffee break every 200 iterations in the tomato.jl file.

We have found that the network learns to generalize as the number iterations increases and so also the accuracy of predictions.

CHAPTER SIX
DISCUSSION AND RECOMMENDATIONS

Deep learning methods for filter extraction is a promising technique with a very good performance, but needs large amount of data and computing resources. Data is oxygen for ML. To train a network such as this, there need for large amount of dataset for each grade. Cifar10 and MNIST for example, are some of the popular dataset commonly use in ML. The former consists of 60000 32 x 32 colour images in 10 classes, with 6,000 images per class. MNIST contains has a training set of 60,000 examples, and a test set of 10,000 examples.

Due irregular shapes, sizes and varieties of tomatoes there needs to be a large amount of data examples from all the varieties for better performance. In addition to that, we have several preprocessing methods such as background removal. Other techniques exist, such as augmenting the training set with artificially distorted versions of the original training samples to improve the overall performance.

The work we presented in this thesis can be improved or expanded in different directions. The future work will allow for experimentation on large amount data verified by expert on the related field. The future work may also implement the trained network on to a device (Mobile phone, Raspberry PI, personal computer or even cluster).

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J.,... Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)* (p. 265). Savannah, GA, USA: PeerJ.

Abhineet, S. (2016, June 29). *XRDS : The ACM Magazine for Students website*. (A. Saxena, Editor, & A. Saxena, Producer) Retrieved from XRDS : http://xrds.acm.org/blog/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/

Arif Mohamad, M., Nasien, D., Hassan, H., & Haron, H. (2015). A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition. *(IJACSA) International Journal of Advanced Computer Science and Applications, 6*(2), 204-212.

Baltazar, A., Aranda, J. I., & González-Aguilar, G. (2008). Bayesian classification of ripening stages of tomato fruit using acoustic impact and colorimeter sensor data. *Computers and Electronics in Agriculture, 60*(2), 113–121.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computing*(12), 2385-2404.

Bengio, Y. (2009, November 15). Machine Learning Deep Architectures for AI. *Foundations & Trends in Machine Learning, 2*(1), 1-127. doi:10.1561/2200000006

Choras, S. R. (2007). Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems. *INTERNATIONAL JOURNAL OF BIOLOGY AND BIOMEDICAL ENGINEERING, 1*(1), 1-11.

Deng, & D., Y. (2013). Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing, 7*(3–4), 197–387.

El-Bendary, N. e. (2014, October 5). Using machine learning techniques for evaluating

tomato ripeness. *Expert Systems with Applications*, 1-14. doi:10.1016/j.eswa.2014.09.057

Fadilah, N., & Mohamad-Saleh, J. (2014). Color feature extraction of oil palm fresh fruit bunch image for ripeness classification. *13th International Conference on Applied Computer and Applied Computational Science (ACACOS'14)* (pp. 51–55). Applied Computational Science.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Ann Eugen*, 179-188.

Frank, M., Roehrig, P., & Pring, B. (2017). *What to Do When Machines Do Everything: How to Get Ahead in a World of AI, Algorithms, Bots, and Big.* New Jersey: John Wiley & Sons, Inc., Hoboken.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36*(4), 193–202.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning Book.* MIT Press.

Google.ai. (2017, August 1). *tools: Google Inc.* Retrieved from Google Inc. Web site: https://ai.google/tools/tensorflow/

Guerrero, E. R., & Benavides, G. M. (2014). Automated system for classifying Hass avocados based on image processing techniques. *IEEE Colombian conference on communications and computing (COLCOM) 2014* (pp. 1-6). Columbia: IEEE.

H.G. Bock et al. (eds.). (2013). Model Based Parameter Estimation. *Contributions in Mathematical and Computational Sciences*.

Hackeling, G. (2014). *Mastering Machine Learning with scikit-learn.* Birmingham B3 2PB, UK: Published by Packt Publishing Ltd.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2099). What is the Best Multi-Stage Architecture for Object Recognition? *IEEE*, 00-00.

Jean-Philippe, V., Tomoko, M., Shin'ichi, S., & Yuji, U. (2007). *High-Level Feature Extraction Using SVM With WalkBased Graph Kernel.*

Jinxia, L., & Yuehong, Q. (2011). Application of SIFT feature extraction. *Tenth international conference on electronic measurement & instruments IEEE.* Chengdu, China: IEEE.

Julia. (2017, October 26). *Julia Packages*. Retrieved from A Julia Langauge Website: http://pkg.julialang.org

Juliacomputing. (2017, August 12). *Machine Learning And AI*. Retrieved from A Julia Computing: https://juliacomputing.com/domains/ml-and-ai.html

JuliaCon. (2017, September 10). *JuliaCon 2017*. Retrieved from Julia Conference Website: juliacon.org

Karandeep, K. (2016, April). Machine Learning: Applications in Indian Agriculture. *International Journal of Advanced Research in Computer and Communication Engineering, 5* (4), 342-344.

Krishnan, K. S. (2017). A Survey on Image Segmentation and Feature Extraction Methods for Acute Myelogenous Leukemia Detection in Blood Microscopic Images. *International Journal of Advanced Research in Computer and Communication Engineering*, 153-155.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 436-44.

LeCun, Y., Bengio, Y., & Hinton, G. (2015, MAY 28). Deep Learning : A Review. *NATURE, 521*, 436.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE. 86*, pp. 2278-2324. NJ 08855 USA: IEEE. doi: 10.1109/5.726791

Marsland, S. (2015). *MACHINE LEARNING: An Algorithmic Perspective, Second Edition.* Boca Raton: CRC Press, Taylor & Francis Group, LLC.

May, Z., & Amaran, M. (2011). Automated ripeness assessment of oil palm fruit using RGB and fuzzy logic technique. *The 13th WSEAS international conference on mathematical and computational methods in science and engineering (MACMESE2011).* (pp. 52–59). Stevens Point, Wisconsin: World Scientific and

Engineering USA Academy and Society (WSEAS).

Min, S., Lee, B., & Yoon, S. (2017, September 1). Deep Learning in Bioinformatics. *Briefings in Bioinformatics, 18*(5), 851–869. Retrieved from https://doi.org/10.1093/bib/bbw068

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill.

Mohamed, A. E.-S., Yarub, A. E., & Mohamed, A. K. (2013). Automated Edge Detection Using Convolutional Neural Network. *(IJACSA) International Journal of Advanced Computer Science and Applications, 4*(10), 12-17.

Mohammed, B., Kamel, B., & Abdelouahab, M. (2017, May 16). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligenge*, 2-17. doi: 10.1080/08839514.2017.1315516

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective.* London, England: The MIT Press Cambridge, Massachusetts.

Opeña1, H. J., & Yusiong, J. P. (2017, February). AUTOMATED TOMATO MATURITY GRADING USING ABC-TRAINED ARTIFICIAL NEURAL NETWORK. *Malaysian Journal of Computer Science, 30*(1), 12-26.

Pachouri, K. K. (2015). A Comparative Analysis & Survey of various Feature Extraction Techniques. *International Journal of Computer Science and Information Technologies (IJCSIT), VI*(1), 377–379.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philos Mag.*(2), 559-572.

pluskid. (2017, November 17). *Mocha : a pluskid repo*. Retrieved from A Github Inc. Website: https://github.com/pluskid/Mocha.jl

Polder, G. G., Heijden, V. d., & Young, I. (2002). Spectral image analysis for measuring ripeness of tomatoes. *Transactions-American Society of Agricultural Engineers, 45*(4), 1155–1162.

Rodolfo, B. (November 2016). MNIST digit classification. In B. Rodolfo, *Building Machine Learning Projects with TensorFlow* (pp. 281-219). Birmingham: Packt

Publishing Ltd.

Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *Elsevier Ltd.*(61), 85-117. Retrieved from www.elsevier.com/locate/neunet

Scholkopf, B., A., S., & Muller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computing, 1*(10), 1299-1319.

Tang, J., Alelyani, S., & Liu, H. (2014, n.d.). *Publication: Feature Selection for Classification: A Review.* Retrieved May 25, 2017, from Michigan State University Web site: http://www.cse.msu.edu/~tangjili/

Tensorflow. (2017, Ocotober 26). *About TensorFlow*. Retrieved from A. TensorFlow Website: https://www.tensorflow.org/

TIOBE. (2017, September). *TIOBE Index for October 2017*. Retrieved from TIOBE The Software Quality Comapany: http://www.tiobe.com/tiobe-index/

Ugonna, C. U., Jolaoso, A. M., & Onwualu, P. A. (2015). Tomato Value Chain in Nigeria: Issues,Challenges and Strategies. *Journal of Scientific Research & Reports*, 501-515.

Yuret, D. (2017, September 21). *Knet*. Retrieved from Github: https://github.com/denizyuret/Knet.jl

Zhang, D., Lee, D.-J., Tippetts, B. J., & Lillywhite, K. D. (2014). Date maturity and quality evaluation using color distribution analysis and back projection. *Journal of Food Engineering*, 161–169.

Zhang, L., & McCarthy, M. J. (2012). Measurement and evaluation of tomato maturity using magnetic resonance imaging. *Journal of Postharvest Biology and, 67*, 37–43.

Zhong, G., Wang, L.-N., Ling, X., & Dong, J. (2016). An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*(2), 265-278.