Fruit Freshness Grading Using Deep Learning

Yuhang Fu

A thesis submitted to the Auckland University of Technology in partial fulfillment of the requirements for the degree of Master of Computer and Information Sciences (MCIS)

2020

School of Engineering, Computer and Mathematical Sciences

Abstract

This thesis presents a comprehensive analysis of a variety of fruit images for freshness grading using deep learning. A number of algorithms have been reviewed in this project, including YOLO for detecting region of interest with considerations of digital images, ResNet, VGG, Google Net, and AlexNet as the base networks for freshness grading feature extraction. Fruit decaying occurs in a gradual manner, this characteristic is included for freshness grading by interpreting chronologically-related fruit decaying information.

The contribution of this thesis is to propose a novel neural network structure, i.e., YOLO + Regression CNNs for fruit object locating, classification, and freshness grading. Fruits as an object, its images are fed into YOLO for segmentation and regression, then for freshness grading. The results reveal that our approach outperforms linear predictive model and demonstrate its special merit.

Keywords: CNN, YOLO, Deep Learning, Fruit Freshness, Regression, Image Recognition

Table of Contents

Table of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), n or material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: Yuhang Fu Date: 1 March 2020

Acknowledgment

This research work was completed as the part of the Master of Computer and Information Sciences (MCIS) course at the School of Computer and Mathematical Sciences (SCMS) in the Faculty of Design and Creative Technologies (DCT) at the Auckland University of Technology (AUT) in New Zealand. I would like to deeply thank my parents for the financial support they provided during my entire time of academic study in Auckland.

My deepest thanks are to my primary supervisor Dr. Wei Qi Yan who has provided me with much appreciated technological guidance and support. I believe that I could not achieve my Master degree without his invaluable help and supervision. In addition, I would like to appreciate my secondary supervisor Dr. Minh Nguyen and our school administrators for their support and guidance through the MCIS study in the past years.

Auckland, New Zealand

March 2020

Chapter 1 Introduction

In this chapter, we will provide an overview on fruit freshness grading, including but not limited to identification of the research problem, answer the questions of motivations, review the existing research trends.

1.1 Background

Fruit spoilage has significant ramifications on economic activities, it is estimated that nearly a third of fruit costs go to decaying matters. Besides, the sale of fruits will be impacted as it is in consumers' perception that spoiled fruits are detrimental to health (Péneau, Linke, Escher, & Nuessli, 2009) as decreased concentrations of amino acids, vitamins, sugar/glucose along with other nutrients inevitably riase public concerns on edibility issues, which all together prompt discussions on this subject to prevent or slow down the decaying process.

Given the significance of food status in people's lives and contribution to the economy, fruit freshness grading becomes important but the manual operation is time-consuming. Grading automation by using computerised approaches is believed as the solution to this problem.

Fruit spoilage refers to human perceptions on fruit quality regarding desirability and acceptance to consumption of the portion being edible and averseness to unfavourable sensory characteristics (Akinmusire, 2011). Research discovers that there exists a strong relationship between bacteria and fruit spoilage, including aerobic psychrotrophic gram-negative bacteria (with secretion of extracellular hydrolytic enzymes that corrupt plant's cell walls), heterofermentative lactobacilli, spore-forming bacteria, yeasts and molds. Bacteria-related fruit degeneration is a consequence of pectin degradation (a structural acidic heteropolysaccharide grown in terrestrial plant's cell walls, mainly consisted of galacturonic acid). Starch/amylum and sugar (or polymetric carbohydrates of same purposes) are then metabolized with produced lactic (an acid that is a metabolic intermediate as the end product of glycolysis releasing energy anaerobically) and ethanol (Rawat, 2015). Colonizing and induced lesions as a result of microbe dissimination are frequently observed, and infestation is a common cause of spoilage for postharvest fruits (Tournas & Katsoudas, 2005). Besides, lack of nutrients can have complications that result in growth of dark spots, e.g., insufficient calcium can cause apples developed cork spots (Sindhi, Pandya, & Vegad, 2016). The exposure to oxygen is another factor as an enzyme known as polyphenol oxidase (PPO) triggers a chain of biochemical reactions involving proteins, pigments, fatty acids and lipids, that lead to fading of the fruit colours as well as degrading to having undesirable taste and smell (Shukla, 2017).

Established research evidence shows that when fruit deterioration occurs, fruit goes through a series of biochemical transformation that leads to changes in its physical conditions, e.g., visual features including colour and shape. Most of these features can be captured. It is expected that computer vision-based approach is the most economical solution. Given the advancement of deep learning technology, grading algorithms should produce satisfactory accuracies (Bhargava & Bansal, 2018) (Rashmi, Sapan, & Roma, 2013).

The state-of-the-art technology in computer vision sees the categories in fruit/vegetable automatic grading (Cunha, 2003) (Pandey, Naik, & Marfatia, 2013): Detections of fruit/vegetable diseases and defects caused by foreign biological invasion (Mahaman, et al., 2004), fruit/vagetable classification for assorted horticultural products (Brosnan & Sun, 2002), estimation of fruit/vegetable nitrogen contents (Tewari, Arudra, Kumar, Pandey, & Chande, 2013), fruit/vegetable object realtime tracking (Ozyildiz, Krahnst-over, & Sharma, 2002), etc.

1.2 Fruit Spoilage Visual Characteristics

Academics in this area have identified the visual characteristics of fruit decay (Barrett, Beaulieu, & Shewfelt, 2010). Fruit colour is derived from natural pigments when ripening, enzymatic and non-enzymatic browning reactions lead to the formation of water-soluble dark colours. Visual characteristics, e.g., shape, wholeness, spots, bruises, and blemishes, can reflect the speed of fruit deterioration (Mitcham, Cantwell, & Kader, 1996). It is observable that fruits with physical defects are vulnerable to diseases and prone to fast decaying. The consistency of physical shape may indicate the thickness of fruits that may have implications of its capability to defend against diseases.

Given the statistics that bacteria-caused fruit spoilage is salient among others, it is reasonable to assume, and is in fact often observed, that bacteria invasioins often start at particular spots then dissimitate that eventually grow into noticebale dents and rots. Rerdsearch identifies a number of types of spoilage with each one assigned distinguishable visual features (Sindhi, Pandya, & Vegad, 2016) (Hartman, 2010), some of the prominent ones are scabs characterised by brown cork spots, rots featured in sunken circular brown spots and a crimson halo in the middle and blotches distinguished in irregularities of spot lobed edges.

Geometric changes are a frequently observed result of fruit degradation. Mostly found chemical compounds for a cell structure arepolysaccharides cellulose, hemicellulose and pectin, and the primary storage is polymer is starch. Microbe invasion occurs via releasing cellular lytic enzymes that corrupt these polymers to extract nutrients (water and other intracellular constituents) for growth. Fruits possess protective epidermis barriers to repel invasion, typically covered by a waxy cuticle layer that gives fruit natural glisters (Lequeu, Fauconnier, Chammai, Bronner, & Blee, 2003). Spoilage renders fruits shrivelled due to loss of cell fluid, served as a strong degradation indicator.

Texture is another important measurement of the level how a fruit has decayed, illustrating general characteristics of fruit surfaces. The texture of a healthy fruit is drived from turgor pressure and plant cell lamella that binds individual cells together (Barret, Bealiue, & Shewflet, 2010). Spoilage can cause deformation and disintegration of cells that result in overall texture morphological transition into wizened surfaces.

The liquids, in combination with semi-permeable membranes and cell walls, give unique appearances and sensual tastes of fruit (Hargava, 2018), and loss of cell liquid (quite often where natural pigments exist) from corruption has significant implications on fruit hue histograms.

1.3 Fruit Freshness Grading

Fruit freshness grading via computer vision technology exploits on the fruit texture, colour and shape for visual feature evaluation. A fruit during a decay process appears in gradual changes, e.g., the growth of dark spots from oxygenizing and shrinkage due to the loss of contained water. In this thesis, efforts have been mainly emphasized on work for the algorithm development of fruit freshness grading.

A literature review (Hargava, 2018) examined fruit spoilage visual features and concluded that most experiments considered that the presence of fruit lesion indicates the start of fruit spoilage, however, did not find a progressive definition of the ongoing decaying fruit that to which degree of spoilage a fruit should be defined according to its biological ageing stage.

Fruit texture, colour and shape are three important visual features for fruit quality grading (Moallem, Serajoddin, & Pourghassem, 2017). The research work (Moallem, Serajoddin, & Pourghassem, 2017) focuses on golden delicious apple and uses SVM + KNN for grading. However, this research project has two categories only: healthy and defect, only takes account of one type of fruit. The limitation to greater fruit quality grading matters is obvious.

Another research work on the quality of tomato grading (Arakeri & Lakshmana, 2016) treated fruit texture, colour, and shape as important features and developed a computer vision solution based on statistics of these features. The problem is thought as a binary classification matter that fruits are either recognized as defected or healthy.

Deep learning is extensively used in visual object recognition. The work (Bresilla, et al., 2019) adopted YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016) for fruit and vegetable recognition. YOLO is fast compared to other approaches, which achieved 20 *fps* image processing speed that is applicable for real-time usage. However, the fruits in the project are constraint to the conditions when the fruits remain connected to the biological hosts.

Another research work (Zeng, 2017) uses a deep neural network VGG for fruit recognition, which (Zeng, 2017) proves that convolutional neural network when going deep, can achieve high accuracy.

In contrast to the previous one, a shallow neural network is adopted (Mureșan, 2018) consisting of four convolutional and pooling layers only for feature extraction, followed by two fully connected layers. However, the source images in this experiment are simple. The images are devoid of background noises. All fruit objects are placed in a pure white background and fixed at a static position.

There are research experiments conducted for fruit freshness issues specifically. An automatic freshness grading system (Nashat & Hassan, 2018) was developed for olive fruit batches by using discrete wavelet transform and textural features. Another work (Prakash, 2018) addressed raspberry spoilage recognition by using deep learning (a 9-layer neural network consisted of 3 convolutional and pooling layers, one input and one output layer).

Mandarin decay process is impacted by a disease called penicillium digitatum, there is research work (Gomez-Sanchis, et al., 2008) dedicated to early detection of this disease by examining decay visual features. The visual features are captured and processed by a combination of decision trees. However, these experiments only focus on one type of fruits, assuming non-background noises. Another problem is that the grading mechanism is a classification model which takes fruit into account as being either healthy or rotten/defect, but the decay process occurs in a gradual fashion, the final predictive layer should regress the output rather than perform a classification task.

1.4 Motivations

We consider fruit freshness grading is one step of fruit post-harvest assessments. A literature review (Mditshwa, Magwaza, Tesfay, & Mbili, 2017) poroposed a detailed summary of post-harvest fruit quality grading. Another research work (Ntsoane, Zude-Sasse, Mahajan, & Mahajan, 2019) evaluated ambient conditions on post-harvest fruits, including temperature, humidity, and the impacts on fruit decay rate.

A review of existing fruit-freshness study inspired us to conduct this experiment as there lacks such research work. Most approaches for fruit grading are based on classification, the fruits are classified either as defect or healthy. For fruit quality grading, academics did not focus on freshness aspect, they only consider overall visual changes, most of them only take account of diseases.

Another motivation is that despite the recent rise of popularity of deep learning, more than half of the academics in their survey (Tripathi & Maktedar, 2019) remained conservative and did not use deep learning methods. Although many non-deep-learning models have achieved high accuracies, utilizing deep learning approach for fruit freshness matters based on digital images is still absent. In addition, this proposal treats the fruit freshness grading as a regression problem, which is the first of this kind of research work to our knowledge. We summarize our motivations:

- Most existed research for fruit freshness matters or related issues are conduced based on classifications different from our approach employing regression for freshness grading.
- Academics tend to simplify the problem evident in their assumptions such as unvarying white background, in contrast to our comprehensive considerations inclusive of complex backgrounds.

- To the best of our knowledge going through literature reviews, there is no existing research work based on deep learning for a systematic approach (a combination of different neural networks) for fruit freshness grading.

The major novelty of this proposal is the development of a systematic solution that assumes complexity (multiple object placements of an assortment of fruit species with noisy backgrounds) in the first place and addresses it by image segmentation for region of interest extraction, the retained information is processed by using isolated deep learning models responsible for individual fruit categories.

1.5 Thesis Structure

This thesis attempts to develop a comprehensive analysis of how these visual features applied to human perceptions that can assist to identify at which degree the fruits have decayed. In the second part of this thesis, a technical overview is provided that covers the-state-of-the-art technology in computer vision and deep learning. For a better understanding of the data representation of fruit freshness, the technical overview illustrates how visual information is captured in artificial neural networks.

The third chapter is an illustration of how visual data will be collected and preprocessed, including visual feature extraction. Data preprocessing is a concept of how source image data can be manipulated to be fed into deep learning models. It is expected that the source data, with added disturbances and improved image quality how visible spectrums are distributed in reality, can enhance model's predictive capability as the model is adaptive to noises as well as how visual characteristics are presented in histograms in real world. There are four types of preprocessing methods introduced, on purposes of having the trained model which is more resistant to noises and accurate on prediction.

The fourth chapter of this thesis is a discussion of algorithmic design in relation to fruit freshness issues. We first approached this problem via intuition plus our biochemistry study which has suggested that abnormalities in various physical properties are the primary indicator for spoilage, by which a linear regression model was built. To compare with what deep learning models have achieved, as evidence in its superb results in computer vision contests and implementations in commercial projects, the construction of a hierarchical deep learning model was introduced, is capable of object localization and classification, as well as regression for fruit freshness degree regression.

The fifth chapter is deployment, entailing how the algorithms were realized given a programming environment and toolboxes. This chapter illustrates training specifications under what circumstances models are derived, as provided in pseudo code.

The sixth chapter is a summary of empirical results, comprised of performance metrics and semantic analyses. We first review the linear predictive model and its production, mainly including explanatioins of the underlying factors that implies failures of the model. Upon the revelation, this chapter narrates through a number of performance metrics along with our comprehension of the issues and why the problems can be addressed in this proposal.

The last chapter concludes major contributions of this thesis and sheds light on futher research interest. Assorted fruit freshness grading is inheritently complex as it is inclusive of a large amount of resembled visual features, many of which are indiscernible from each other that render intractability of grading.

Chapter 2 Literature Review

This chapter provides an overview of the foundation technology as well as the trends. Fruit freshness grading by using computer vision (CV) is an unchartered field but shares the same characteristics with any CV problems. We examine the state-of-the-art neural networks for computer vision tasks, which shed lights on finding possible solutions to address the fruit freshness grading problem, it indicates how the wisdom is conducieve to make accurate assessments on fruit freshness grading.

2.1Machine Learning

Machine learning is a subject (Samuel, 1959) describing how a computer program is capable of learning from human experience for a target, this ability could be measured. Machine learning is classified into three major categories: Supervised learning and unsupervised learning. Data samples are categorized into four types for classification: false negatives (FN), true negatives (TN), true positives (TP), and false positives (FP).

Classification accuracy (CA) is the most frequently used performance evaluation metrics for classification problems. This metric describes overall how accurate a model is when categorizing a data samples into the right class.

$$CA = \frac{TN + TP}{TN + TP + FN + FP} \tag{2.1}$$

Recall is a metric that evaluates a model how good it is to predict relevant data points given the labels.

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

Precision describes how precise a model is to select positive/relevant data points from a set of data points with predicted positive labels.

$$Precesion = \frac{TP}{TP + FP} \tag{2.3}$$

There are several types of loss functions to evaluate the error gap between predictive output and the ground truth:

- Mean square error (MSE)

$$Loss_{L2}(t) = \left(\hat{y}(t) - y(t)\right)^2 \tag{2.4}$$

- Mean absolute error (MAE)

$$Loss_{L1}(t) = |\hat{y}(t) - y(t)| \tag{2.5}$$

- Huber loss (Huber, 1964)

$$Loss_{Huber}(t) = \begin{cases} \frac{1}{2}\left(\hat{y}(t) - y(t)\right)^2, & |\hat{y}(t) - y(t)| < \delta \\ \delta\left(\hat{y}(t) - y(t)\right) - \frac{1}{2}\delta, & otherwise \end{cases} \qquad (2.6)$$

- Cross entropy (given n classes for each probability output $\hat{y}_{o,c}(t)$ on observation o of class c at the time t corresponding to the ground truth $y_{o,c}(t)$)

$$Loss_{CrossEntropy}(t) = -\sum_{c=1}^{n} y_{o,c}(t)\, log\left(\hat{y}_{o,c}(t)\right). \qquad (2.7)$$

In this thesis, we mainly focus on utilizing neural networks for image information processing: localization, classification and regression.

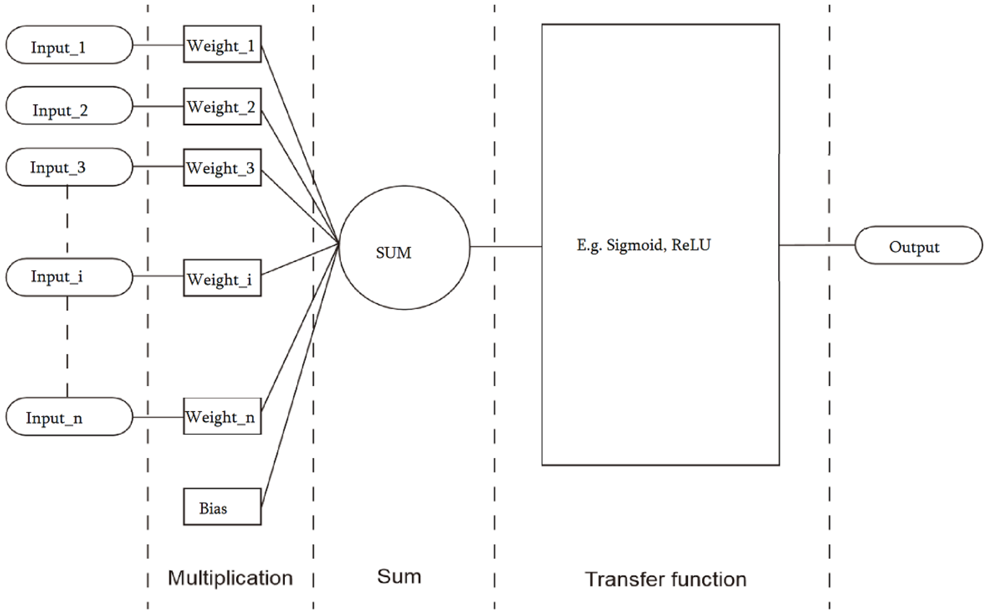


Figure 2.1:A typical neuron of a neural network

2.2 A Literature Review of Artificial Neural Networks

2.2.1 Artificial Neural Networks (ANNs)

Inspired by the working mechanism of biological brains, artificial neural networks are proposed with the philosophy that the algorithms should be capable of “learning” from given events/samples (Kleene, 1956) (McCulloch & Pitts, 1943). The basic block of neural network is neuron, which is a mathematical unit that takes m inputs with corresponding weights w_i and bias b. The weighted inputs are summed and sent to an activation/transfer function.

A neuron at the time (training epoch) t can be summarised as

$$\hat{y}(t) = F_{activate}\left(\sum_{i=1}^{m} w_i(t)x_i(t) + b\right), \tag{2.8}$$

where

- $x_i(t)$ is input at discrete time t with integer $i \in [1, m]$ for m inputs in total.
- $w_i(t)$ is input weight at discrete time t with integer $i \in [1, m]$ for m input weights in total.
- b is bias to the summed weighted inputs.
- $F_{activate}$ is activation/transfer function, e.g. step, sigmoid (Hahnloser & Seung, 2006) hyperbolic tangent and ReLU (Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000) (Hahnloser. & Seung, 2002) as,

$$step(x) = \begin{cases} 1, & x > threshold \\ 0, & otherwise \end{cases}, \tag{2.9}$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}}, \tag{2.10}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{2.11}$$

$$ReLU(x) = \max(0, x), \tag{2.12}$$

- $\hat{y}(t)$ is the output of the neuron at the time t.

$ReLU(x)$ is currently the most useful activation function in deep learning as it requires little computation resources but presents comparable results to that of $sigmoid(x)$. This represents a forward calculation of a neuron. For each forward, there is an output $y(t)$ corresponding to a ground truth y_{truth}. The gap between $\hat{y}(t)$ and $y(t)$ is the error (loss value) expressed as

$$Loss(t) = \hat{y}(t) - y(t). \tag{2.13}$$

The loss functions have different levels of sensitivities to sample outliers and to a target of regression or classification problem. The hyperparameters of the network are adapted, given the back-propagated errors. Assume the changes to weight $w_{i,j}$ of a node that connects the i-th layer and the j-th layer is

$$\Delta w_{i,j} = -\frac{\alpha\ (\partial\ Loss)}{\partial\ w_{i,j}} \tag{ 2.14}$$

where α is the learning rate that controls how fast the weight should be updated. There are numerous schemes for learning rate changes. The most selective one is a constant and a decreasing scheduler with an initially defined learning rate α_{start} associated with training time epoch t. Thus:

- Linear decreasing scheduler (with a decrease constant $c \in (0,1)$ and a minimum learning rate ϵ)

$$\alpha = \begin{cases} (1-ct)\alpha_{start}, & \alpha \geq \epsilon \\ \epsilon, & otherwise. \end{cases} \tag{2.15}$$

- Exponential decreasing scheduler (with a decrease constant $c \in (0,1)$)

$$\alpha = \alpha_{start}^{(1-ct)}. \tag{2.16}$$

With $\Delta w_{i,j}$, the weight $w_{i,j}$ can be updated by using stochastic gradient descent (SGD) (Bottou & Bousquet, 2012) at each epoch t of parameter update

$$\mathrm{w_{i,j}(t)} = \mathrm{w_{i,j}(t-1)} + \Delta w_{i,j} + \xi, \tag{2.17}$$

where ξ is a stochastic term.

Adaptive Moment Estimation (Adam) is an extension to SGD. Given four arguments β_1 for decaying the average gradient (0.9) and β_2 for average squared gradient0.999, α is the learning rate and $\epsilon = 10^{-8}$ to prevent zero division error. The process can be illustrated as

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1-\beta_1)\Delta_w L^t$$

$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1-\beta_2)\left(\Delta_w L^{(t)}\right)^2$$

$$\widehat{m}_w = \frac{m_w^{(t+1)}}{1-\beta_1^{t+1}}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-\beta_2^{t+1}}$$

$$w^{t+1} \leftarrow w^{t} - \alpha \frac{\widehat{m}_w}{\sqrt{\widehat{v}_w} + \epsilon} \tag{2.18}$$

where '$\leftarrow$ ' is denoted as an assignment operator, $\Delta_w L^t$ is the propagated error, $m_w^{(t+1)}$ and $v_w^{(t+1)}$ represent the first-order and second-order error with forgetting factor β_1 and β_2. The ratio $\frac{m_w^{(t+1)}}{\sqrt{v_w^{(t+1)}}}$ with an added ϵ to avoid zero division is the update item.

One advantage of a neuron is the functioning of leveraging the power of linearity and non-linearity to interpret input information, which most traditional machine learning approaches lack. Neurons collectively form a neural network.

2.2.2 Convolutional Neural Networks (CNNs)

A convolutional neural network (CNN) (Wu, 2017) is comprised of one or more convolutional layers with associated subsampling step, whose outputs are extracted as features and flattened and fed into a series of fully connected layers. Variants of CNN have different structures but the basic remains unaltered.

Source data samples, usually with the size of $w \times h \times c$ for an image having width w, hight h and three color channels in RGB, i.e., $c = 3$ and $c = 1$ for a greyscale image, are fed into the first convolutional layer. The first convolutional layer has k filters (kernels) of $n \times n \times q$ size (a kernel should be smaller than the input in terms of sizes) that convolves with the source data and allows features to be passed by the kernels' configuration. The kernels are initialized randomly and adapt the sample data with the help of backpropagation. The extracted features are subsampled (pooling), and the process repeats until the visual features are ready to be fed into fully connected layers for classification or regression.

In CNNs, a filter shares the same concept in a neural network as a neuron, except that the neurons in CNNs are 2D or 3D given the input data usually is 2D or 3D. The fully connected layers in CNN are same as a typical layer of a neural network. CNN has a series of particular layers named pooling layers. These are used for fast subsampling because image inputs are often significant in size and the information is often redundant (e.g., an apple contained in an image of a size of $448 \times 448 \times 3$ is still likely visible when the image is subsampled to half to $224 \times 224 \times 3$). There are two types of pooling:

- Max pooling

  Assume $R_{x,y,w,h}$ is a small 2D region of an image I with width w and height h at a relative position (x, y), here we denote the pixel value of this position (x, y) as $p_{x,y}$,

$$R_{x,y,w,h} = \begin{bmatrix} p_{x+h,y} & \cdots\ p_{x+h,y+i} \cdots & p_{x+h,y+w} \\ \vdots & \ddots & \vdots \\ p_{x+i,y} & \cdots\ p_{x+i,y+i} \cdots & p_{x+i,y+w} \\ \vdots & \ddots & \vdots \\ p_{x,y} & \cdots\ p_{x,y+i} \cdots & p_{x,y+w} \end{bmatrix}. \qquad (2.19)$$

  Max pooling is the selection of the max value from $R_{x,y,w,h}$ so that

$$MaxPooling(R_{x,y,w,h}) = \max(R_{x,y,w,h}). \qquad (2.20)$$

- Average pooling

  Similar to that in max pooling, for an image region $R_{x,y,w,h}$, the process can be expressed as

$$AvePooling(R_{x,y,w,h}) = \frac{1}{w \times h} sum(R_{x,y,w,h}). \qquad (2.21)$$

2.2.3 R-CNN

R-CNN (Regional Convolutional Neural Network) (Girshick, Donahue, Darrell, & Malik, 2013) is a novel CNN structure for the contribution of semantic image regions to the targets which has been proven high accuracy in the PASCAL VOC dataset competition.

The advantage of this proposal is based on CNN features, not being randomly selected but initially generated with semantic segmentations so that the CNN features can better reflect the image content. Region proposal for R-CNN is implemented with selective search (Uijlings, Sande, Gevers, & Smeulders, 2012). Selective search is an object detection algorithm that uses a variety of selection strategies and merges the results. There are four primary conditions taken into account: Texture, colour, size and overlapping (Uijlings, Sande, Gevers, & Smeulders, 2012). The final location hypotheses are proposed after a trade-off between quality and quantity. Location hypotheses are ranked, and low-ranking proposals are removed to ensure that the selected location bounding boxes are of high confidence with regard to contain an object.

The location hypotheses are fed into deep learning networks. In the proposal (Uijlings, Sande, Gevers, & Smeulders, 2012), a large CNN was considered with the capability of extracting 4096 features and SVM for the final classification problem.

2.2.4 Fast R-CNN

Fast R-CNN (Girshick, Fast R-CNN, 2015) is a step forward of R-CNN with faster detection speed via computation simplification. This network employs VGG16 network which is 9 times faster than the original R-CNN proposal.

Instead of feeding region proposals, this algorithm applies feature maps to classifications. Region of Interest (RoI) is pooled and converted into a fixed-size feature region. Softmax with a fully connected layer is used for classification instead of heavy SVM

2.2.5 Faster R-CNN

Faster R-CNN (Ren, He, Girshick, & Sun, 2017) is an advancement of Fast R-CNN that is built for high accuracy and fast computation speed. Both R-CNN and Fast R-CNN use selective search which is time consuming. This algorithm suggests us a convolutional neural network to learn from source images and produce feature maps. The generated features are fed into the same CNN as Fast R-CNN.

The R-CNN family is classic but remains slow in contrast to other approaches despite novel mechanisms to accelerate computation speed. It requires region extractions which are computation intensive. Although Faster R-CNN addresses this matter by introducing Region Proposal Network (RPN), it adds the complexity of model construction.

2.2.6 SPPnet

Spatial pyramid pooling network (SPPnet) (He K. , Zhang, Ren, & Sun., 2014) accelerates computation speed and improves accuracy with shared computation. The features selected with the pooling technique are processed with various filter sizes and are then concatenated as the input to a fully connected network.

This structure reflects global and local visual features. It is faster than R-CNN with comparable performance metrics, but not fast enough in comparison to other approaches.

2.2.7 Bilinear CNN

B-CNN (Lin, RoyChowdhury, & Maji, 2017) splits input matrices into two streams which are multiplicated. The products of multiplication are transformed into linear form and the model then continues computation the same as in typical fully connected layers. The resultant matrix X from two streams $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$ before linearization is shown as eq. (2.22)

$$X = \frac{1}{n}\left(\sum_{i=1}^{n} a_i b_i^T \right) + \varepsilon. \qquad (2.22)$$

However, the two-stream mechanism requires two times computation resources. A basic illustration of the structure is shown in Fig. 2.7.

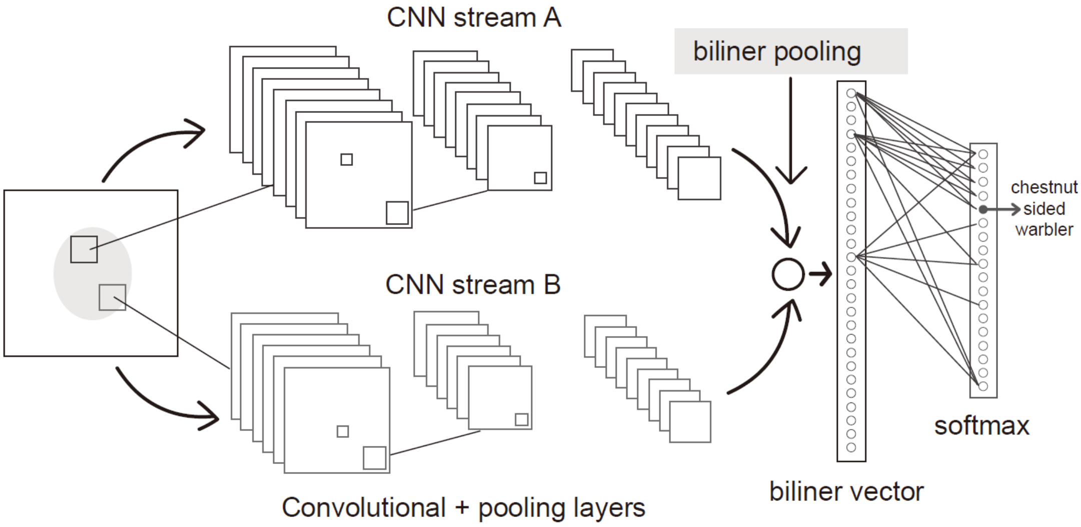


Figure 2.2: B-CNN structure

2.2.8 AlexNet

AlexNet is the name of a particular structure of CNN, awarded for its performance over several image recognition competitions (Krizhevsky, Sutskever, & Hinton, 2017). This work presents the importance of neural network depth that has tremendous impacts on computation efficiency when training (via GPU).

AlexNet has five convolutional layers followed by three fully connected layers. The network is separated by using half with each one being trained on an isolated GPU until the final output layer for prediction.

2.2.9 GoogleNet and Inception

GoogLeNet (Szegedy, et al., 2014) is a novel CNN structure that adopts an inception module. This research work (Szegedy, et al., 2014) examined the performance of the neural network that includes the impact on computation speed due to enlarged network size and uniformity of kernel sizes that is prone to inefficiency when dealing with features with various shapes.

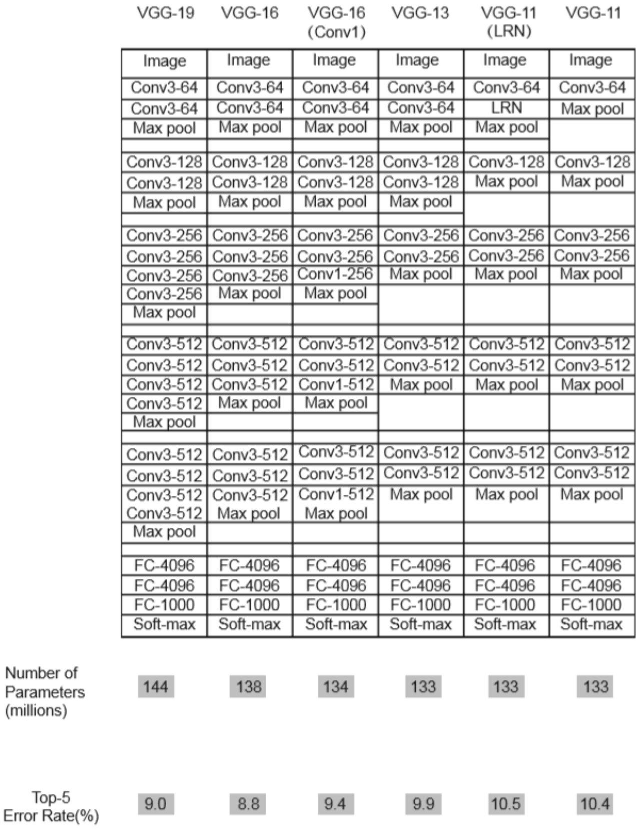
GoogLeNet proposes a novel neural network architecture that exploited the advantage of sparsity, the existing work proposed that there exists a great likelihood of performance enhancement given clustering sparse matrices. GoogLeNet introduces inception modules that leverage local sparse structure of a convolutional vision network. In intuition into this concept is that the filters should capture data of a large scale as well as keep retain fine-resolution information.

2.2.10 VGGNet

VGGNet (Simonyan & Zisserman, 2015) has uniformed convolution and fully connected layers with high accuracies in numerous competitions. VGGNet refers to the family of convolutional neural networks where kernels and convolutional layers are careful designed. Fig. 2.10 shows the comparisons of various VGGNet.

2.2.11 ResNet

ResNet (He K. , Zhang, Ren, & Sun, 2015) is the name of a particular structure of CNN with cross-layer information that enables information to "skip" the activation gates at the next layer and sent to the following one. From a mathematical viewpoint, this model deals with vanishing gradient problems, where when a network goes deep, the propagated information takes a long time to converge, the derivatives of the propagated errors might be vanished.



	VGG-19	VGG-16	VGG-16 (Conv1)	VGG-13	VGG-11 (LRN)	VGG-11
	Image	Image	Image	Image	Image	Image
	Conv3-64	Conv3-64	Conv3-64	Conv3-64	Conv3-64	Conv3-64
	Conv3-64	Conv3-64	Conv3-64	Conv3-64	LRN	Max pool
	Max pool	Max pool	Max pool	Max pool	Max pool	
	Conv3-128	Conv3-128	Conv3-128	Conv3-128	Conv3-128	Conv3-128
	Conv3-128	Conv3-128	Conv3-128	Conv3-128	Max pool	Max pool
	Max pool	Max pool	Max pool	Max pool		
	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256
	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Conv3-256
	Conv3-256	Conv3-256	Conv1-256	Max pool	Max pool	Max pool
	Conv3-256	Max pool	Max pool			
	Max pool					
	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
	Conv3-512	Conv3-512	Conv1-512	Max pool	Max pool	Max pool
	Conv3-512	Max pool	Max pool			
	Max pool					
	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512	Conv3-512
	Conv3-512	Conv3-512	Conv1-512	Max pool	Max pool	Max pool
	Conv3-512	Max pool	Max pool			
	Max pool					
	FC-4096	FC-4096	FC-4096	FC-4096	FC-4096	FC-4096
	FC-4096	FC-4096	FC-4096	FC-4096	FC-4096	FC-4096
	FC-1000	FC-1000	FC-1000	FC-1000	FC-1000	FC-1000
	Soft-max	Soft-max	Soft-max	Soft-max	Soft-max	Soft-max
Number of Parameters (millions)	144	138	134	133	133	133
Top-5 Error Rate(%)	9.0	8.8	9.4	9.9	10.5	10.4

Figure 2.3: VGGNet family: The structure and error rate

Given a weight matrix $W^{l-1,l}$ for connection between layer $l-1$ and l, another weight matrix $W^{l-2,l}$ for connection between layer $l-2$ and l, for a forward propagation, we have this layer l output

$$h^{l} = \sigma(W^{l-1,l} \cdot h^{l-1} + W^{l-2,l} \cdot h^{l-2} + b^{l}). \tag{2.23}$$

For a backpropagation, weights are updated with regard to the two preceding layers with this layer error E^{l},

$$\Delta w^{l-2,l} = -\alpha \frac{\partial\, E^{l}}{\partial\, w^{l-2,l}}, \tag{2.24}$$

$$\Delta w^{l-1,l} = -\alpha \frac{\partial E^{l}}{\partial w^{l-1,l}}, \tag{2.25}$$

where α is learning rate.

Figure 2.3 shows a ResNet module, where input x is fed into two weight layers and x duplicate is summed with the two weight outputs, then go through another activation function (e.g., ReLU or sigmoid).

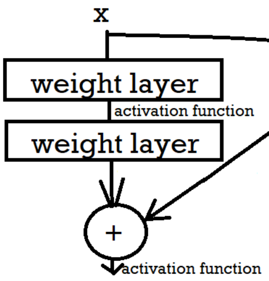
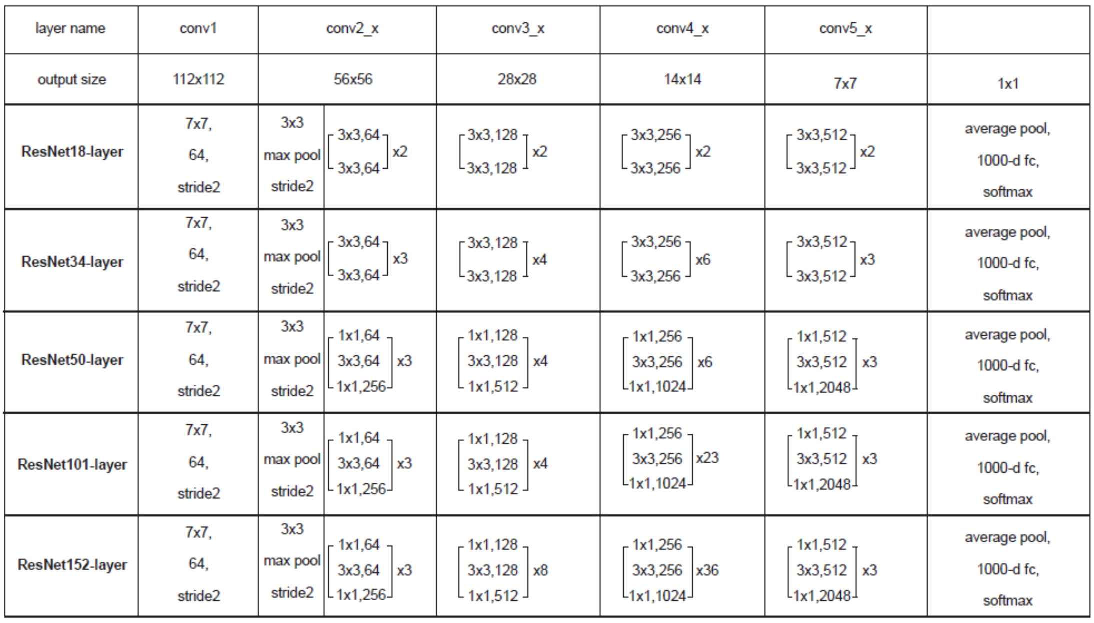


Figure 2.4: A ResNet module

ResNet has many variants, of which the best performer is ResNet152 (He K. , Zhang, Ren, & Sun, 2015) consisted of 152 layers. This study shows that the growth of network depth can improve accuracy, in our experiments, we selected the deepest one for fruit freshness grading.

ResNet variants are similar in construction, e.g., adoption of pooling methods and utilization of filters, but different from the number of layers. Table 2.1 illustrates these variant structures.



layer name	conv1	conv2_x		conv3_x	conv4_x	conv5_x	
output size	112x112	56x56		28x28	14x14	7x7	1x1
ResNet18-layer	7x7, 64, stride2	3x3 max pool stride2	[3x3,64 3x3,64] x2	[3x3,128 3x3,128] x2	[3x3,256 3x3,256] x2	[3x3,512 3x3,512] x2	average pool, 1000-d fc, softmax
ResNet34-layer	7x7, 64, stride2	3x3 max pool stride2	[3x3,64 3x3,64] x3	[3x3,128 3x3,128] x4	[3x3,256 3x3,256] x6	[3x3,512 3x3,512] x3	average pool, 1000-d fc, softmax
ResNet50-layer	7x7, 64, stride2	3x3 max pool stride2	[1x1,64 3x3,64 1x1,256] x3	[1x1,128 3x3,128 1x1,512] x4	[1x1,256 3x3,256 1x1,1024] x6	[1x1,512 3x3,512 1x1,2048] x3	average pool, 1000-d fc, softmax
ResNet101-layer	7x7, 64, stride2	3x3 max pool stride2	[1x1,64 3x3,64 1x1,256] x3	[1x1,128 3x3,128 1x1,512] x4	[1x1,256 3x3,256 1x1,1024] x23	[1x1,512 3x3,512 1x1,2048] x3	average pool, 1000-d fc, softmax
ResNet152-layer	7x7, 64, stride2	3x3 max pool stride2	[1x1,64 3x3,64 1x1,256] x3	[1x1,128 3x3,128 1x1,512] x8	[1x1,256 3x3,256 1x1,1024] x36	[1x1,512 3x3,512 1x1,2048] x3	average pool, 1000-d fc, softmax

Figure 2.5: ResNet family structure

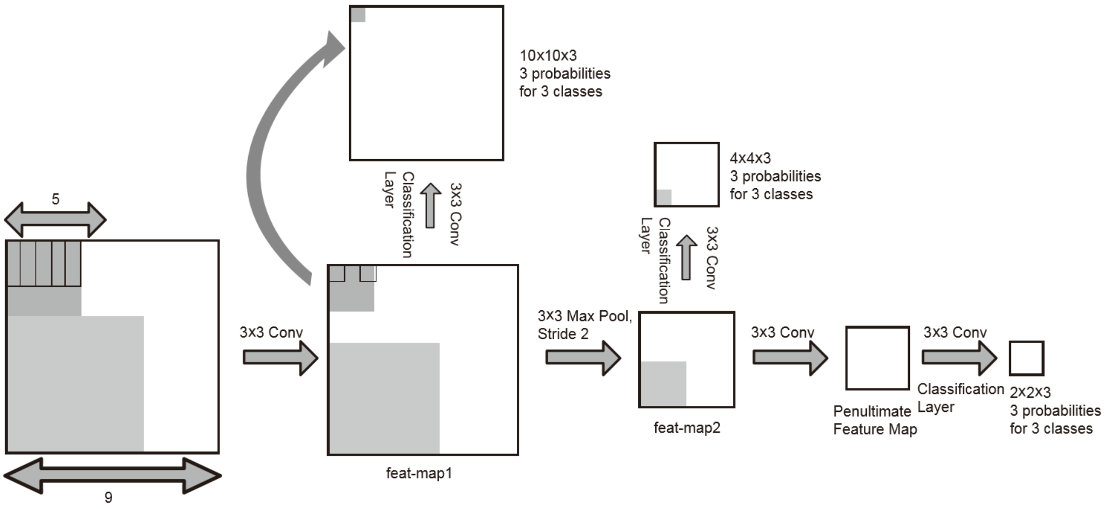


Figure 2.6: SSD convolutional layers on which predictions are made

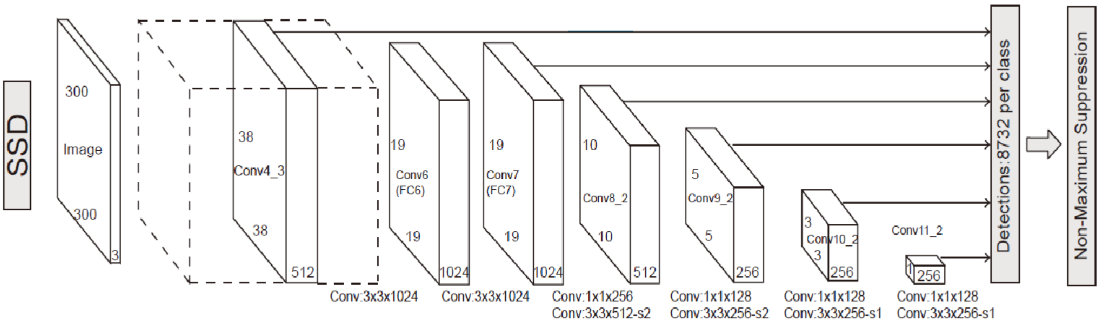


Figure 2.7: A typical SSD structure

2.2.12 SSD

Single Shot Detector (SSD) (Liu, et al., 2015) is the CNN class that permits object detection using only one deep neural network, rather than the traditional approaches that one forward only produces one detection of a possible label. This model regards multiscale feature layers added to the end of a base network. These layers progressively decrease in shape and permit detection predictions at multiple scales. The convolutional predictors for object detection are of various sizes with smallest one to $3 \times 3 \times C$ (C for channel number). In Fig.2.6, that feature maps are downsized progressively; on each layer, there is a corresponding 3×3 filter convolving through the map.

Let's denote $x_{i,j}^{p} = \{1,0\}$ for matching the i-th default box to the j-th ground truth box of category p. Given this matching strategy, there should be $\sum_i x_{i,j}^{p} \geq 1$. The loss function can be expressed as

$$L(x,c,l,g) = \frac{1}{N}\Big(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\Big). \tag{2.26}$$

where N is the number of default boxes that match ground truth boxes. For $N = 0$, here we define loss $L = 0$, α is the weight term set to 1.0 for cross validation. $L_{conf}(x,c)$ refers to confidence loss which is the softmax loss over class confidence c. The equation states that for each positive prediction (object detected), there apply penalties to wrong class estimation. There is no penalty applied to non-object existence boxes

$$L_{conf}(x,c) = -\sum_{i\in Positive}^{N} x_{i,j}^{p} \log(\hat{c}^{p}) - \sum_{i\in Negative} \log(\hat{c}^{0}) \tag{2.27}$$

where

$$\hat{c}^{p} = \frac{\exp\left(c_i^{p}\right)}{\sum_p \exp\left(c_i^{p}\right)}. \tag{2.28}$$

$L_{loc}(x,l,g)$ is localization loss,

$$L_{loc}(x,l,g) == -\sum_{i\in Positive}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{i,j}^{p}\, smooth_{L1}\left(l_i^{m} - \hat{g}_j^{m}\right), \tag{2.29}$$

in which ground truth box g is obtained and predicted box l with (cx, cy) as the centre of the default bounding box d with respect to its width w and height h

$$\hat{g}_j^{cx} = \frac{g_j^{cx} - d_i^{cx}}{d_i^{w}}, \tag{2.30}$$

$$\hat{g}_j^{cy} = \frac{g_j^{cy} - d_i^{cy}}{d_i^{h}}, \tag{2.31}$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^{w}}{d_i^{w}}\right), \tag{2.32}$$

$$\hat{g}_j^{h} = \log\left(\frac{g_j^{h}}{d_i^{h}}\right). \tag{2.33}$$

Smooth L_1 loss is defined as

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & |\&x| < 1 \\ |x| - 0.5, & otherwise. \end{cases} \tag{2.34}$$

Improvements are observed in SSD in combination with other networks or with adjustments for particular contexts. Deconvolutional SSD (Fu, Liu, Ranga, Tyagi, & Berg, 2017) saw increased mAP over PASCAL VOC and COCO dataset with added deconvolutional layers.

RefineDet (Zhang, Wen, Bian, Lei, & Li, 2017) inherits merits of SSD and improves the prediction capability through the adjustments of anchors. An attention mechanism is introduced dedicated to text region image detection (He, et al., 2017). A feature-focused network with a built-in bi-directional network circulating semantic features saw improvements in accuracy (Wang, et al., 2019). For face detection, a context-assisted SSD is developed with novel contextual anchors introduced (Tang, Du, He, & Liu, 2018).

2.2.13 YOLO

YOLO stands for You Only Look Once (Redmon, Divvala, Girshick, & Farhadi, 2016). YOLO takes the object anchoring process as a regression problem that the anchor coordinate, width w and height h should be defined for object localization. One advantage of YOLO over other CNN approaches is that this network takes account of the global input rather than locals.

YOLO divides the input images into a grid consisted of a $S \times S$ grid of cells. If a cell contains part of an object, the cell is responsible for this object detection. Each cell produces B bounding boxes and confidence scores accordingly. Confidence scores describe the confident level of the model regarding the bounding box containing the target object. The confidence can be defined as in the eq. (2.35)

$$Prob(Object) \times IOU_{predict}^{truth} \tag{2.35}$$

where IOU (Intersection Over Union) is a process for calculating the overlapping area of two unions. In this case, the IOU should be the intersection between the ground truth and the predict. The resultant bounding box should be the shared area of the two unions. $Prob(Object)$ refers to whether the grid cell contains an object or not.

Consider that each object should have a label, the confidence can be expressed as eq. (2.36) for the prediction of a bounding box encapsulating an object of a class

$$\begin{aligned} Prob(class_i) \times IOU_{predict}^{truth} &= \\ Prob(class_i \mid Object) \times Prob(Object) \times IOU_{predict}^{truth}&. \end{aligned} \tag{2.36}$$

As a result, each cell should predict a total of five parameters. The four parameters that define a bounding box are location and size (x, y, w, h). The probability of each class is associated with the detected object.

In real implementation, a particular YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016) proposed by the author has 24 convolutional layers followed by two dense layers. YOLO employs 1×1 reduction layers with 3×3 convolutional layers following behind. This structure is shown in Figure 2.8.

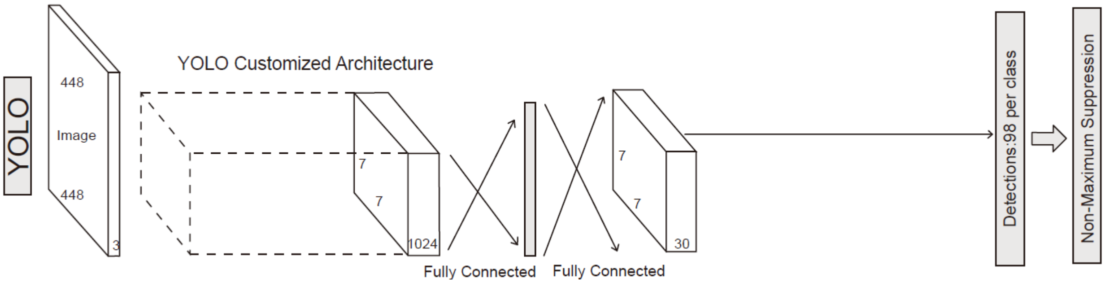


Figure 2.8: YOLO architecture

The loss function for YOLO is given as

$$\begin{gathered}
Loss = \\
\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\
\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\widehat{w}_i} \right)^2 + (\sqrt{h_i} - \sqrt{\widehat{h}_i})^2 \right] + \\
\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} \left[\left(C_i - \hat{C}_i \right)^2 \right] + \\
\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{noobj} \left[\left(C_i - \hat{C}_i \right)^2 \right] + \\
\lambda_{noobj} \sum_{i=0}^{S^2} \mathrm{I}_{i,j}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2 . \qquad (2.37)
\end{gathered}$$

YOLO loss function consists of five parts regarding penalties to the bounding box parameters x, y, w, h and class prediction of C for an image, divided into $S \times S$ cells with each cell predicting B bounding boxes. λ_{coord} and λ_{noobj} are scalers which are set at 5.0 and 0.5 respectively to control the penalties of bounding box coordinates and classification. Equally treating bounding box coordinates and object classification errors might lead to model instability as cells without containing any object tend to go zero in localization confidence scores (Redmon, Divvala, Girshick, & Farhadi, 2016).

$\mathrm{I}_{i,j}^{obj}$ is a binary operator that denotes the presence of an object for the i-th cell and the j-th proposal. It is expected that the width and height of the bounding box should be tight to the contour of the object so that square root is applied to w and h.

YOLO is different from SSD. The most distinct one is the employment of multiscale convolutional layers by using SSD. The convolutional layers in SSD are progressively downsized along with the bounding boxes for prediction. YOLO is simple in structural construction.

YOLO has developed multiple variant structures, which is similar to the prediction mechanism but different in specifications with improvements, e.g., YOLO9000 (Joseph & Ali, 2016) is capable of detecting 9000 object categories with improvements to the prior work. YOLOv3 is the state-of-the-art network. This model improved prediction accuracy via added residual layers on top of YOLOv2 and Darknet-19.

2.3 Justifications of Network Selections

In this thesis, YOLOv3 is adopted as the network for object localization and classification, VGG, googleNet, AlexNet and ResNet are employed for regression. The primary reason for the selections is computation speed given the hierarchical structure. Networks with shallow layers are thought to perform worse than the deep ones but the computation cost is much lower. Hence, the decision on selections comes into the balance between the depth of a network and how much processing power our hardware expects to consume. Since the number of fruit types is small, it is plausible to conclude that the network for object classification and localization does not need to go too deep, YOLOv3 satisfies our needs.

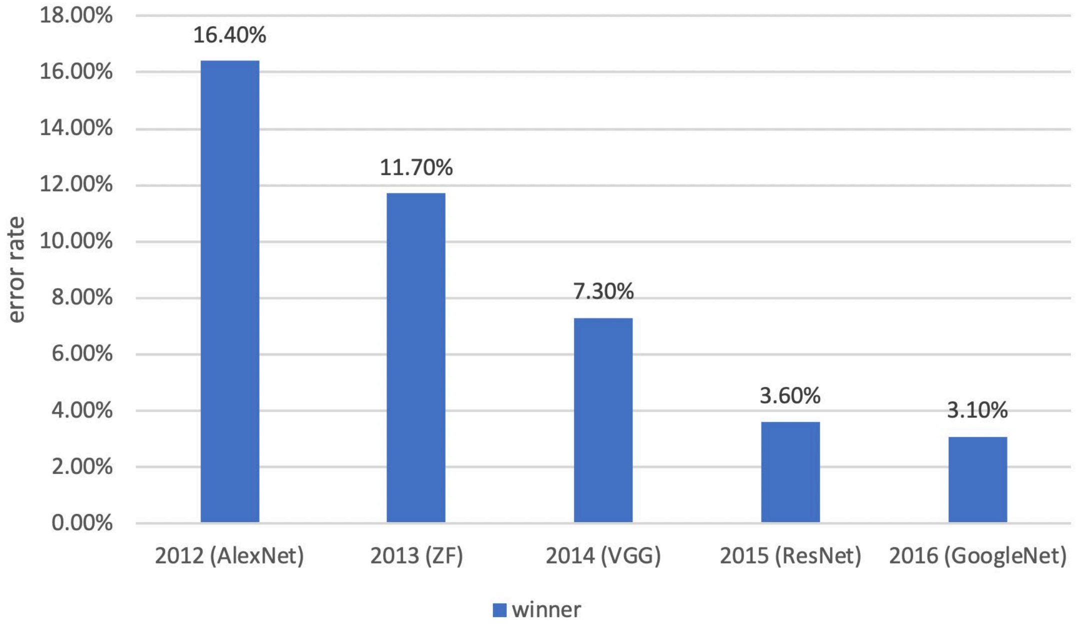


Figure 2.9: Error rates for top perframance networks in ImageNet image recognition competition between 2012 - 2016

ImageNet is a computer vision competition project that provides large databases consisting of more than 14 million annotated images. Since 2010, ImageNet has hosted a number of image recognition contest annually, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). However, the contest results can justify the performances on the given dataset only, there is not conclusive evaluation study claiming absolute superiority of any particular networks. In addition, we regard the freshness grading matter as a regression issue, which does not totally conform to what ILSVRC was designed for. For this reason, four base networks (VGG, ResNet, GoogleNet, and AlexNet) are selected. The top performce models are listed in Figure 2.9.

Chapter 3 Data Preparation

In this chapter, we provide a detailed description of how source data was collected and augmentation/enhancement are implemented before executing the algorithm. Given the novelty of our research project, fruit data is not available publicly. Our illustration is made clearly in motivations with regard to data preparation, we provide empirical evidence why the collection is an accurate representation of the data for fruit freshness grading.

3.1 Data Collection

Since there is no existing fruit freshness dataset available, this project encompasses the work for data collection. The collected dataset consists of six types of fruits: Apple, banana, dragon fruit, orange, pear and kiwi, derived from a variety of locations with different ambient noises, irrelevant adjacent objects, and light conditions. In total, there are (approximate) 4,000 images collected with each type of fruit about 700. The dataset was split into training and validation sets at the ratio of 1: 9 (90% for training and 10% for validation).

The freshness grading is scaled from 0.0 to 10.0 with 0.0 indicating total corruption and 10.0 for total freshness. In this project, we define the fruits being harvested as absolute freshness with a numerical level description of 10.0. However, based on extensive research on the definition of absolute degradation, there lacks a conclusion of definitive judgement on this matter. As suggested in existing research (Akinmusire, 2011), fruits not being edible or not being recognized are labelled as the highest level of decay, and labelled such fruits as near-to-zero level of freshness. The labelling process is subjective.

The fruit images were sourced from frames extratced from manually recorded video footages.

It is believed that the decay process is nonlinear. For example, consider time spans, that an apple degrades from the moment when it was harvested and sliced till the moment when it grows brown spots and regarded in common sense not edible, it continues decaying to the level that it is highly corrupted. The two decay processes may take totally different amounts of time. For this reason, labelling the fruits according to the elapsed time from the moment the fruits were harvested and cut into pieces in a linear manner is not accurate hence not implemented, e.g., if an apple degradation process takes about five days, the labelling plan should not linearly assign the apple with the level 8 after one day, level 6 after two days, etc. The labelling task is very subjective, so that in order to address this issue, 10 participants were invited to engage in the labelling work.

After completion of image labelling, a few images were sampled (about three images for each type of fruits at different decay levels), the participants were consulted to give their suggestions (freshness degrees in numerical description), then the mean and standard deviation of these proposed freshness levels were recorded.

For fruit images with significant grade gaps between participants (with standard deviation greater than or equal to 3.0), the independent raters were contacted for grading for the second time in attempt to narrow the disagreement. Fruit labels are kept unaltered if the freshness degrees proposed by the participants are close to what our team first proposed, and modified if the initially proposed freshness level is far from the suggestions derived from the 3rd party participants in consensus. Images obatined from a same video are assigned close freshness grading. Figure 3.1 shows how the worlflow of the labelling task.

The ratio of chlorophylls, carotenoids, anthocynanis as well as other compounds determines the colour of apple peel, with various degrees of impurities and distributions of these chemicals. Fresh apple peel is low in chlorophyll and carotenoid concentrations (Knee, 1972) and spoilage leads to gradual degradation of the constituent pigments, that reflect different wavelengths in spectrophotometry. The apple with a freshness score of 8.20 in table of graded fruit shows the colour of fresh apple peel. This apple is rich in fructose, sucrose and glucose as displayed on the sliced surface. Microbe colonization was observed in brown spots and the grade goes down to 5.30. Fungi invasion is evident on the apple image labelled at a freshness level of 1.45.

A banana when ripe having bright yellow colour is likely a result of carotenoid accumulation (Davey, et al., 2007). Contomitant to this prominent pigment, flavonoids and betalains are found in banana peel that adjust the appearance in mixture of colours from orange/red to violet/blue with the yellow being the dominant (Pandey, et al., 2016). Similar to the pattern of apple degradation, banana decays alone with fading brightness and growth of brown spots highly likely caused by microbe invasion. The banana, exhibited in the table of fruit freshness levels, firstly reflects yellow/green colour, then went to corruption with dark blotches nearly having its peel covered.

The main compositions of orange peels and flesh are pectin, cellulose, and hemicellulose if excluding water that represents 60% - 90% of weight (Bampidis & Robinson, 2006) (Zheng, et al., 2011), pigments are mostly carotenoids and flavonoids that give orange the red apearance. Orange degrades with continuing loss of water and growth of microbes on the surface, and this phenomenon is clearly displayed in the transition between the orange with a freshness level of 8.45 and one of 3.45.

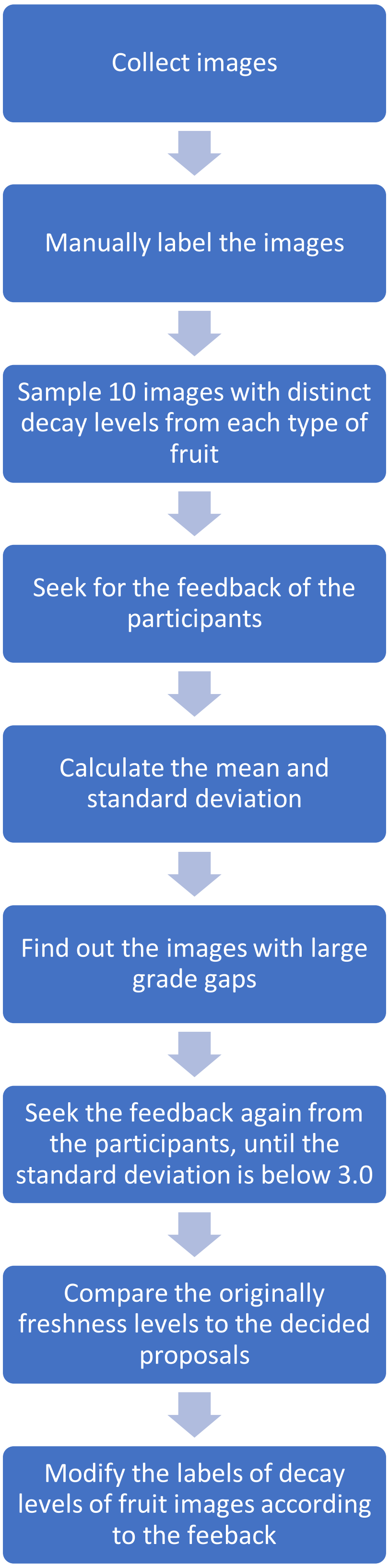
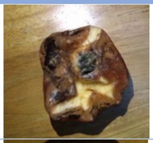
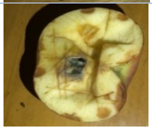
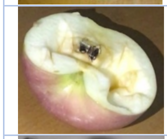
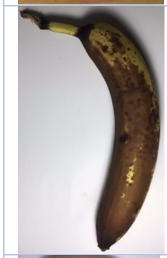
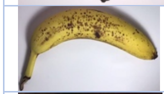
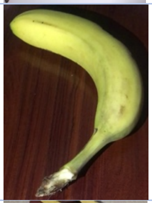
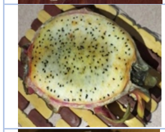
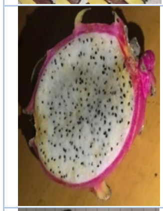
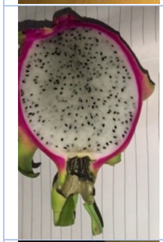
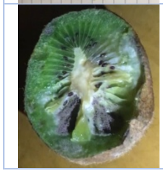
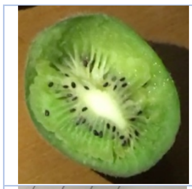
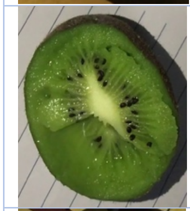
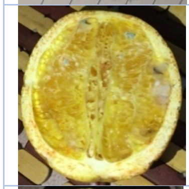
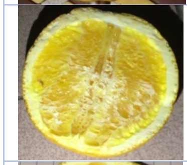
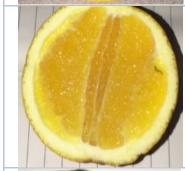
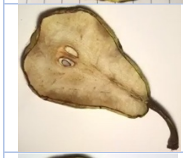
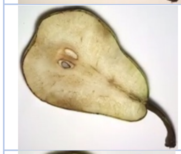
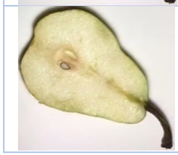

Collect images
Manually label the images
Sample 10 images with distinct decay levels from each type of fruit
Seek for the feedback of the participants
Calculate the mean and standard deviation
Find out the images with large grade gaps
Seek the feedback again from the participants, until the standard deviation is below 3.0
Compare the originally freshness levels to the decided proposals
Modify the labels of decay levels of fruit images according to the feeback

Figure 3.1: The labelling process

Table 3.1 shows 18 distinct fruit images, three fruit types are at various decay levels.

Table 3.1: The means and standard deviations of the fuit freshness levels

Fruit Images	Fruit Names	The Means	The Standard Deviations
	Apple	1.45	0.35
	Apple	5.30	0.75
	Apple	8.20	0.84
	Banana	2.75	0.96
	Banana	6.00	1.05
	Banana	8.15	0.74
	Dragon fruit	3.40	0.86
	Dragon fruit	5.10	0.89
	Dragon fruit	7.8	0.84
	Kiwi fruit	2.50	1.10

	Kiwi fruit	7.25	0.51
	Kiwi fruit	7.70	1.08
	Orange	3.45	1.12
	Orange	5.30	0.95
	Orange	8.45	0.27
	Pear	2.90	0.83
	Pear	5.35	0.53
	Pear	8.45	0.52

Dragon fruit has distinct colours and shape from others with dominant red and yellow/green appearance. The exotic aesthetic exterior look is given by belatains comprised of red-violet betacyanins and yellow betaxanthins (Herbach, Stintzing, & Carle, 2006). As shown in Table 3.1, dragon fruit peels are resistant to microbe colonization during the process of degeneration whereas the flesh was invaded with growing yellow-brown spots.

Kiwi fruits have rich green colour which is a visual manifestation of chlorophylls when degrading gives rise to the formation not only pheophytins but also pyropheophytins that renders olive-brown colour to the fruit (Schwartz & Von Elbe, 2006). Our experiment sees degradation alone with dehydration and growth of fungus.

Pears are similar with apples on physical exterior as well as having a characteristic compartmented core. The green/yellow peel is a result of congregated chlorophylls and once degradation occurs, chlorophylls degenerates, blue-black pheophytins and pyropheophytins are produced (Schwartz & Von Elbe, 2006). Microbe colonization can appear in brown spots as well. In Table 3.1, fruit freshness levels indicate that dehydration happens alone with decolorization of pears.

3.2 Image Quality Enhancement

Many of the source images are of low quality, e.g., blurred and low exposure to light. Several image enhancement approaches were taken to ensure the quality of the images. The contrast enhancement allows the revelation of latent information for too much or too little ambient light exposure. Some spots of interest in the contrast applied images are more evident than in the derived initially ones.

Given a three-dimensional image $I(x, y, z)$ and each pixel value $v(x, y, z)$, there exists a contrast factor $f_{contrast}$ which renders a pixel value as same as the average pixel value of the whole image when $f_{contrast} = 0$, keeps the pixel value unchanged when $f_{contrast} = 1$. Pixel value variation increases if $f_{contrast}$ increases. The relationship between $f_{contrast}$ and input/output pixel values is described as

$$v_{x_{new}, y_{new}, z_{new}} = f_{contrast} v_{x,y,z}. \tag{3.1}$$

Denote $v_{\min i}$ as the minimum pixel value and $v_{\max i}$ as the maximum pixel value in the input image, $v_{\min o}$ and $v_{\max o}$ as the minimum and maximum pixel value in the output image respectively, here:

$$v_{x_{new}, y_{new}, z_{new}} = \left(v_{x,y,z} - v_{\min i}\right) \times \left(\frac{v_{\max o} - v_{\min o}}{v_{\max i} - v_{\min i}} + v_{\min o}\right). \tag{3.1}$$

$f_{contrast}$ = 1.2 was chosen. This is determined as the result of human perceptions to the degree that the contrast-processed images are inclusive of necessary visual features, being enhanced enough to render granularities that may be easy for neural network training.

The third party raters were invited to evaluate the quality of the contrast-processed images how much they are confident about or feel comfortable with the images highlighting the fruit object visual features, $f_{contrast} = 1.2$ is the best choice.

In this experiment, some images are blurred due to vibration when shooting the videos. This was addressed by introducing sharpeness. It was observed that granular details are more evident than in the image before applying sharpening. Fruit edges are precise in contrast to the original. It is believed that if corruption concerns granularity, sharpened images can render better results. Interpolation and extrapolation can be used in image sharpening (Haeberli & Voorhies., 1994). We define a 2D filter for smoothing

$$kernel_{smooth} = \frac{1}{13}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \qquad (3.2)$$

We consider that fruit spoilage features appear in granularities that a kernel of a size of 3×3 should be suficient in covering and highlighting granular visual features. For any source image I_{source}, the convolution result I_{smooth} is expressed as

$$I_{smooth} = I_{source} \cdot kernel_{smooth}, \qquad (3.3)$$

where $\cdot$ denotes a convolution multiplication operator.

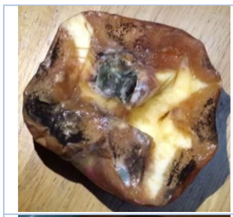
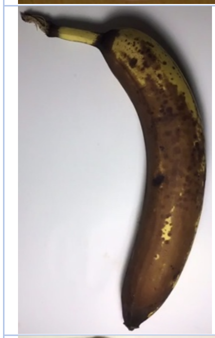
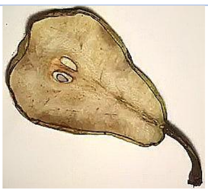
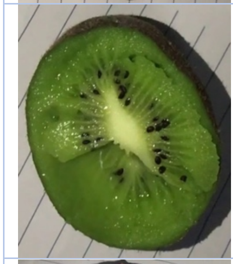
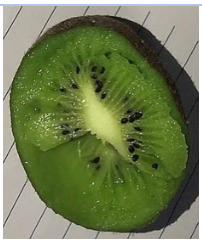
Similar to that of the contrast process, we define a sharpness factor $f_{sharpen}$, the derived image I_{blend} is obtained (Haeberli & Voorhies., 1994)

$$I_{blend} = (1 - f_{sharpen})I_{smooth} + f_{sharpen}I_{source}. \qquad (3.4)$$

The interpolation result for $f_{sharpen} \in (0, 1)$ has the effects of partially blurring the image I_{source} and extrapolation where $f_{sharpen} \in (1, +\infty)$ inverses smoothing transformation to sharpening. Provided that decrement of $f_{sharpen} \in (0, 1)$ renders increasingly blurring effects, of a result of linear extrapolation, $f_{sharpen} \in (-\infty, 0)$ blurs multifolds of what single $kernel_{smooth}$ renders.

Table 3.2: The data augmentation with contrast and sharpening

Source Images	After contrast applied	After sharpening applied

3.3 Image Augmentation

Image augmentation is the methodology to transform source images into images with added information, including scaling, rotation, crop and added random noises. A number of augmentations were experimented; based on the observations, rotation and random noises are included out of robustbess concerns. All images were rotated with an angle of $120°$ by using the equation (3.1). Denote an image as I of a two-dimensional matrix with corresponding coordinates (x, y) for pixel value v,

$$I(x,y) = v_{x,y}. \tag{3.6}$$

Denote a rotation matrix as R,

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}. \tag{3.7}$$

For any θ degree rotation,

$$[x_{new}, y_{new}] = [x, y] \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}, \tag{3.8}$$

The new image I_{new} can be represented as

$$I_{new}(x,y) = I\left([x, y] \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}\right). \tag{3.9}$$

The source images are three-dimensional with z-axis indicating the channel. For an RGB encoded image I_{rgb} with $z = 3$, the rotation matrix is applied to all three dimensions.

All images are added with random noises consisting of random changes of brightness, contrast, saturation and erasion of 10 image regions. The added random noises follow the sequential order: Random brightness adjustment, random contrast, and random erasion of 10 image regions.

- Random brightness adjustment

Given a three-dimensional image $I(x, y, z)$ and each pixel value $v(x, y, z)$,with a brightness factor $f_{brightness}(v_{add})$, where

$$f_{brightness}(v_{add}) = \frac{v(x,y,z) + v_{add}}{v(x,y,z)}, \tag{3.10}$$

indicating the level of brightness adjustment in proportion to the pixel value, here sets $f(v_{add}) \in [0.9, 1.5]$. This can be intuitively interpreted as that the pixel value might be as low/dark as 90% of its original and as high/bright as 150% of its original.

- Random contrast

$$v_{x_{new}, y_{new}, z_{new}} = f_{contrast} v_{x,y,z}, \tag{3.11}$$

where $f_{contrast}$ denotes the level of contrast of an image. In this thesis, here chooses $f_{contrast} \in [0.9, 1.5]$ randomly.

- Random erasion of image regions

Random removal of image regions (Zhong, Zheng, Kang, Li, & Yang, 2017) is an image augmentation technique that addresses generalization issues. This technique removes parts of input image that is expected to enhance the robustness of a neural network in the absence of part of the input image.

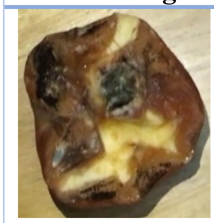
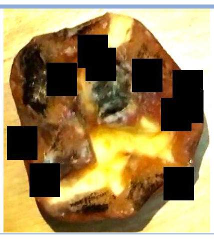
Assume an input image I with width w_I and height H_I, two integers $x_{start} \in [0, w_I]$ and $y_{start} \in [0, h_I]$ are set as the start coordinates (x_{start}, y_{start}). Then, we define the width and height of a region with a proportion r_b to the width and height w_I and h_I of the image, e.g., $r_b = 0.15$. The two coordinates (bottom left (x_{start}, y_{start}) and top right (x_{end}, y_{end})) of the removed region are defined as:

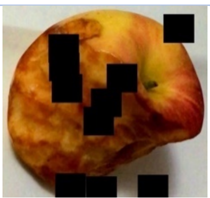
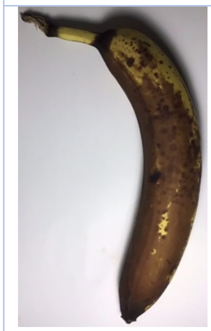
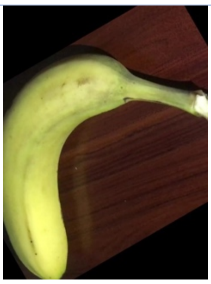
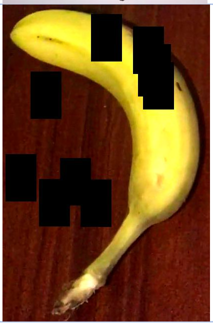
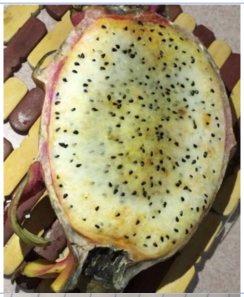
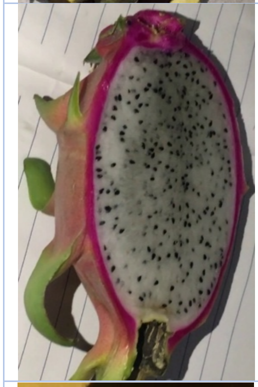
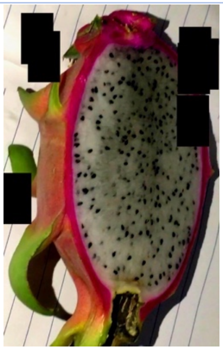
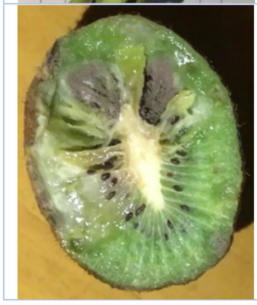
$$(x_{start}, y_{start}) = \big(random(0, w_I), random(0, h_I)\big) \tag{3.12}$$

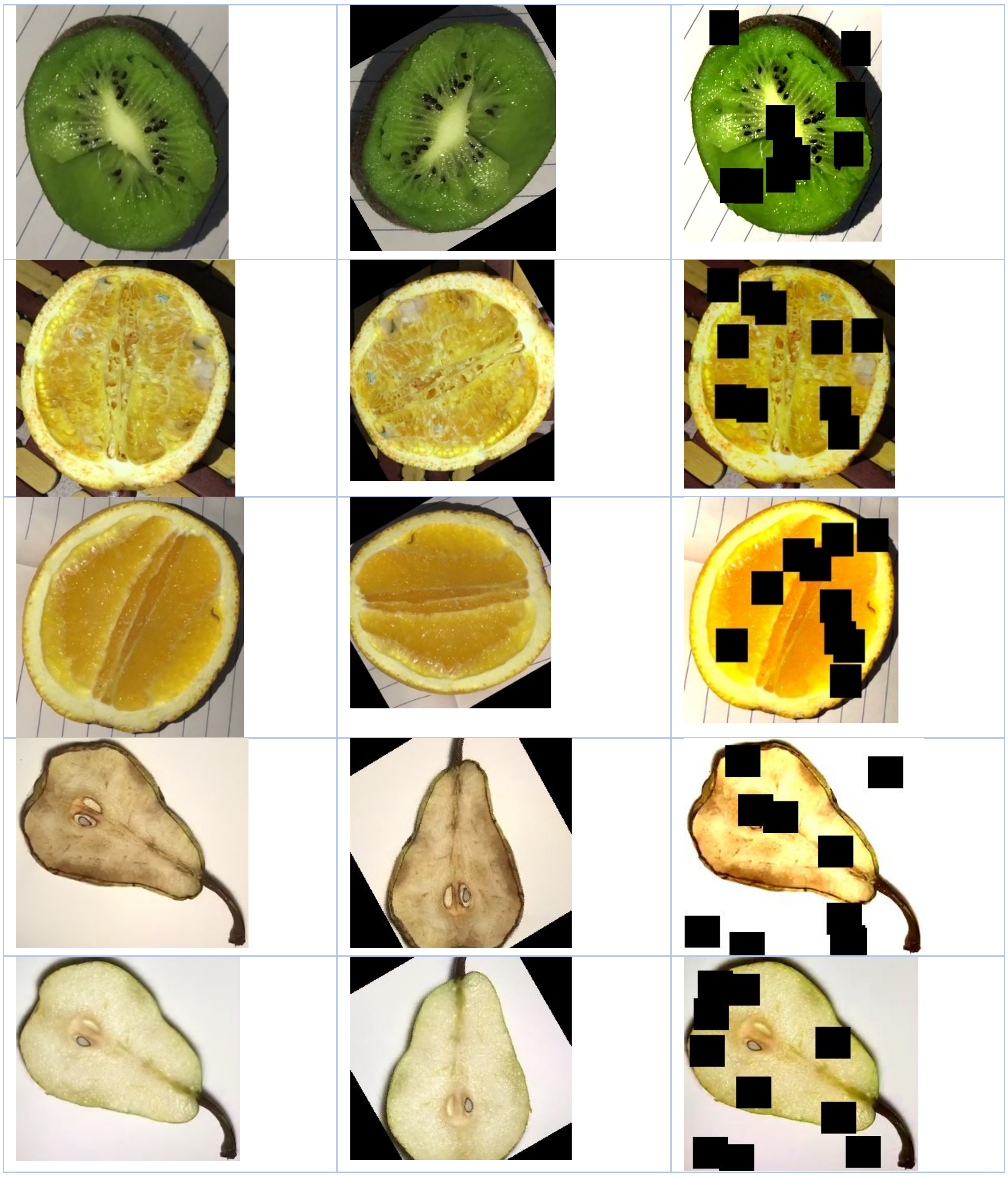
$$(x_{end}, y_{end}) = (x_{start} + r_b\, w_I, y_{start} + r_b\, h_I) \tag{3.13}$$

The random selection process repeats 10 times. The results are shown as Table 3.3.

Table 3.3: The examples of image augmentation

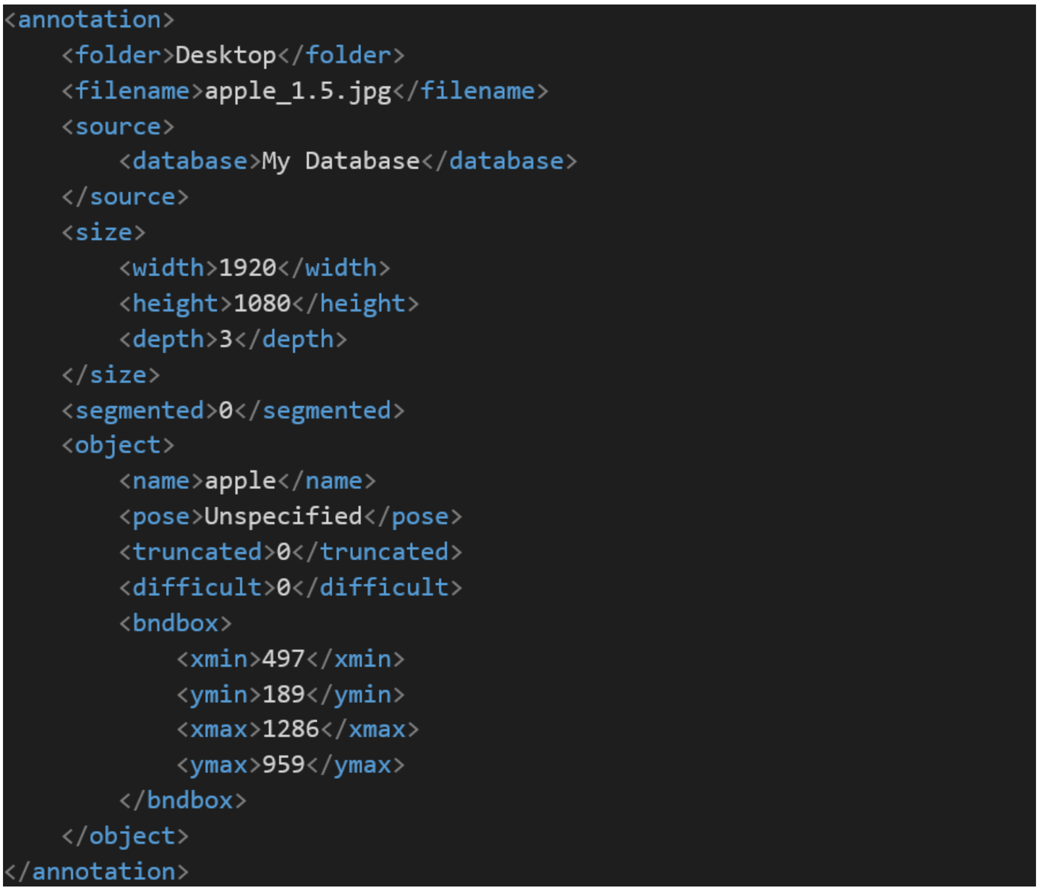
Source Image	**With Rotation**	**With Random Noises**
		



3.4 The bounding box

All images are labelled in VOC annotation format (Everingham, Gool, Williams, Winn, & Zisserman., 2010) that describes the location. Given an image (taken by iPhone X) with the size $1920X1080$, the object location is described in bottom left and top right coordinates as in a bounding box.



```
<annotation>
    <folder>Desktop</folder>
    <filename>apple_1.5.jpg</filename>
    <source>
        <database>My Database</database>
    </source>
    <size>
        <width>1920</width>
        <height>1080</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>apple</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>497</xmin>
            <ymin>189</ymin>
            <xmax>1286</xmax>
            <ymax>959</ymax>
        </bndbox>
    </object>
</annotation>
```

Figure 3.2: An example of VOC labelling

The fruit images are cropped so that the resultant images retain the regions of interest. The region of interest is located via the height and width of the object in the image. In contrast to the rectangular shape of the bounding box, most object shapes (the ground truth of region of interest) are polygons or with rounded angles or corners. As a result, most cropped images contain various levels of noises.

The ground truth of a bounding box should keep most of the object information with minimal background noises. Unfortunately, if it includes the maximal object information, there exist large noises. Out of optimization concerns to permit inclusion of most object information whereas reducing irrelevant image areas, here sets a minimal information retention $area_{obj}$ percentage 65% out of the total region of interest $area_{totalObj}$

$$\frac{area_{obj}}{area_{bndBox}} \geq \approx 65\%. \tag{3.15}$$

Under the satisfied retention percentage condition, the cropped image region (the rectangular bounding box) $area_{bndBox}$ is determined that maximizes $area_{obj}$ under the satisfied condition $\frac{area_{obj}}{area_{bndBox}} \geq \approx 65\%$.

Data pre-processing is an important step ahead of model training and testing. Standard image sizes are 256×256, 128×128, 96×96 and 60×60 (Tripathi & Maktedar, 2019). The input images are resized into 416×416 for fruit freshness features often appear in granularity, e.g., small dark spots scattered across the skin of fruit. In contrast to fruit diseases' features, fruit ageing features are subtle and hard to capture. The relatively large size (high resolution) of input images can deliver rich information for granular details. In the end, we collected nearly 4,000 images in total with each type of fruit having about 700 images with various ambient noises.

In conclusion, the data preprocessing work includes six classes of fruits with various decay stages. Data augmentation is extensively considered in this thesis. For each image, there are four variants: sharpened with contrast, rotated with random noises. There are two types of labels for fruit objects: Freshness grades and VOC annotated locations.

Chapter 4 Algorithm Design

In this thesis, a hierarchical deep learning model is constructed and illustrated in details, YOLO is proposed for fruit classification and localization, whose results are fed into a second one (regression CNN) for freshness grading. In comparison to the deep learning method, a linear model focusing on texture and colour of images was proposed, the relevant analysis paves the way for explaining the reason of adopting a deep learning approach.

4.1 A Linear Proposal

Simple ambient noises refer to the image background with little distractions, usually plain black or white colour. In an environment, fruit localization and freshness grading become easy, as simple pixel-value manipulation can render satisfacgory results. The primary advantage of this project is fast computation for fruits grading.

This thesis proposes a simple solution regarding how to localize a fruit and how to grade its freshness. Because fruits have distinct appearances when the background is a plain or pure colour, a simple pixel-value threshold can be applied to segment a fruit object from an image. Image regions within the pixel thresholds will be selected while others are masked. The contour of the selected image regions will be depicted to determine the bounding boxes for the object detection.

Denote an image as I consisting of pixel $v_{x,y,z}$ where $x \in [1, width]$, $y \in [1, height]$ and z is the channel, for example, an RGB image has $z \in [1,256]$. a binary mask is obtained

$$mask(x, y, z) = \begin{cases} 1, & v_{x,y,z} \in threshold \\ 0, & otherwise. \end{cases} . \quad (4.1)$$

where the $threshold$ is the pixel value of a particular fruit. For apples, the most observed colours are beige and crimson with RGB colors $(166, 123, 91)$ and $(220, 20, 60)$, respectively. Thus, the colour thresholds can be defined as

$$threshold_r = [166 \pm 20, 220 \pm 20] \quad (4.1)$$

$$threshold_g = [123 \pm 20, 220 \pm 20] \quad (4.2)$$

$$threshold_b = [9q \pm 20, 60 \pm 20] \quad (4.3)$$

Another issue worth contemplating is the colour gradient. It is expected that near to the edges of an object, there exist large gradients. It is a task of edge detection. In this experiment, Canny edge detector was selected.

For freshness grading, we regard the brightness and the pixel values within a bounding box as the two conditions. It is believed that generally for a rotten fruit, it grows with brown/dark spots. This appearance change results in the increases of pixel values and the decreases in brightness. Entropy for any given image I with histograms h_i is

$$entropy(I) = -\sum_i (h_i \cdot \log(h_i)) . \quad (4.5)$$

A brightness value for any given image I with pixel value p_i, where $i = 0,1, \dots, n$; n represents the number of pixels that comprise the image

$$brightness(I) = \frac{1}{n}\sum_{i}(p_i). \tag{4.6}$$

The freshness level is calculated by the equation as

$$freshness = k_e\ entropy(I) + k_b\ brightness(I) + b, \tag{4.7}$$

where k_e and k_b are weight adjustment parameters, b is the bias. These parameters are determined via linear regression, assuming a regression output y_i and a data sample x_i , where x_i consists of n features/dimensions, thus,

$$\hat{y}_i = \beta_0 + \beta_1\ x_{i,1} + \beta_2\ x_{i,2} + \cdots + +\beta_n\ x_{i,n}. \tag{4.8}$$

The loss function for linear regression is

$$Loss = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2\ . \tag{4.9}$$

To minimize the loss; hence,

$$\vec{\hat{\beta}} = \arg_{\hat{\beta}} min\ \ Loss(X, \vec{\beta}). \tag{4.10}$$

Therefore, $\vec{\hat{\beta}} = \{b, -k_e, k_b\}$.

A number of fruit images were collected of various decay levels and calculated the entropy as well as brightness of the detected bounding box, meanwhile k_e and k_b are determined. For example, a fresh apple, cut only in a few second, should be shining and present high brightness and low entropy; while a rotten apple should present the opposite way. The fresh apple should have a fresh level close to 10.0, the rotten one should be near to 0. Adjustments have been given to k_e and k_b so as to make the sum of the entropy and brightness closer to the corresponding freshness level in correspondence with actual fruit entropy and brightness scores.

4.2 A Hierarchical Deep Learning Model

In this thesis, we propose a new deep learning model: YOLO + Regression CNN (YOLO-RegreCNN), which is a hierarchical neural network that employs YOLO whose predictive bounding boxes are fed to the regression CNN for freshness grading. Regression CNNs are individually trained for each type of fruit. In this project, there are six types of fruit, for which six regression CNNs are trained. YOLO first classifies the class of the visual object, i.e., fruit as well as the bounding box which localizes the object, according to the classified fruit type, the corresponding regression CNN for this type of fruit is activated to run for freshness level regression.

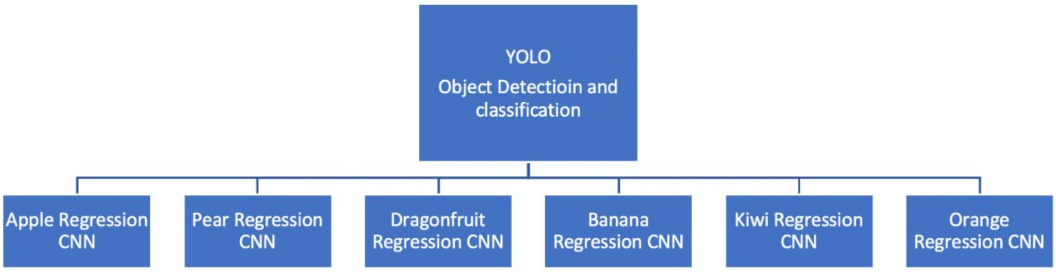


Figure 4.1: YOLO + Regression CNN process

Source images are fed into YOLO for object recognition, where the central coordinates, width and height of the bounding box are determined. YOLO is responsible for object classification. With YOLO prediction, the model maps the predicted class of the detected fruit to its corresponding regression using convolutional neural network. The detected object region in the image is cropped from its background as the input image to the regression of the fruit class.

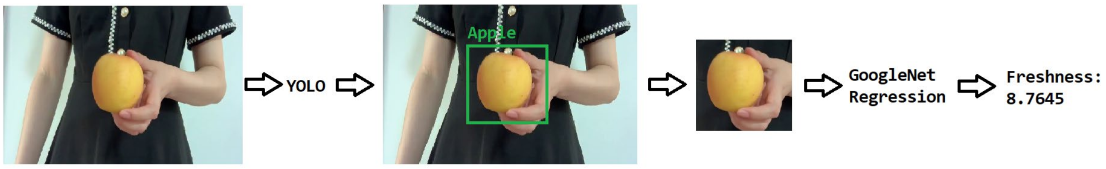


Figure 4.2: An illustration of image process in the proposal hierarchical deep learning model

Define a set of input data D, in which

$$D = \{I_1, I_2, \dots, I_n\} \tag{4.10}$$

where n is the total number of input images, I_i with $i \in [1, n]$ is the i-th image. Our input images are having RGB colors. This defines each image I_i is three dimensional.

The input images are resized to a square. The image I_i with a 2D matrix of pixel values $p_{x,y}$ at the coordinate (x, y) is defined as

$$I_i = \begin{bmatrix} p_{0,0}, p_{0,1} & \cdots & p_{0,w} \\ \vdots & \ddots & \vdots \\ p_{h,0}, p_{h,1} & \cdots & p_{h,w} \end{bmatrix}. \tag{4.11}$$

For a square image, there exists $n = m$.

To further prevent overfitting, additional random flips are applied to images, after YOLOv3 takes the source data and starts the computation. Here we define a comprehensive abstract context of image information at the time t rendered by YOLOv3 whose prediction is $\hat{Y} = \{\widehat{y_1}, \widehat{y_2}, \dots, \widehat{y_n}\}$

$$\hat{Y}(t) = P(\xi_{YOLO}(t)|D). \tag{4.12}$$

For YOLO, a bounding box is obatined, the associated object class, the estimation is

$$\widehat{y_\iota} = \{\widehat{x_\iota}, \widehat{y_\iota}, \widehat{w_\iota}, \widehat{h_\iota}, \widehat{c_\iota}\}. \tag{4.13}$$

According to the predicted class $\widehat{c_\iota}$, the anchored box position and size $\widehat{x_\iota}, \widehat{y_\iota}, \widehat{w_\iota}, \widehat{h_\iota}$, the source image I_i is cropped. The derived new image is

$$newI_i = crop\left(I_i, \widehat{x_\iota}, \widehat{y_\iota}, \widehat{w_\iota}, \widehat{h_\iota}\right) \tag{4.14}$$

$where\, \widehat{x_\iota}\, and\, \widehat{y_\iota}$ are the central position of the predicted bounding box, the $crop(I_i, \widehat{x_\iota}, \widehat{y_\iota}, \widehat{w_\iota}, \widehat{h_\iota})$ for the i-th image I_i can be expressed as

$$crop(I_i, \widehat{x_\iota}, \widehat{y_\iota}, \widehat{w_\iota}, \widehat{h_\iota}) =$$

$$\begin{bmatrix} p_{\widehat{x_\iota}-\frac{\widehat{w_\iota}}{2}, \widehat{y_\iota}-\frac{\widehat{h_\iota}}{2}} & \cdots & p_{\widehat{x_\iota}+\frac{\widehat{w_\iota}}{2}, \widehat{y_\iota}-\frac{\widehat{h_\iota}}{2}} \\ \vdots & \ddots & \vdots \\ p_{\widehat{x_\iota}-\frac{\widehat{w_\iota}}{2}, \widehat{y_\iota}+\frac{\widehat{h_\iota}}{2}} & \cdots & p_{\widehat{x_\iota}+\frac{\widehat{w_\iota}}{2}, \widehat{y_\iota}+\frac{\widehat{h_\iota}}{2}} \end{bmatrix} \tag{4.15}$$

The cropped image $newI_i$ is fed into a regression convolutional neural network. Hereinafter, we define the regression convolutional neural network $\xi_{rege}(t)$ at the training epoch t, for a cropped image dataset is $newD = \{\, newI_1, newI_2, \dots, newI_m\}$ there is

$$\hat{R} = P\left(\xi_{regr}, \widehat{c_\iota}\middle|newD\right) \tag{4.16}$$

$\hat{R}$ is the set of fruit freshness regressed values,

$$\hat{R} = \{\mathrm{r}_1, \mathrm{r}_2, \dots, \mathrm{r}_\mathrm{n}\}. \tag{4.17}$$

The hierarchical model can be expressed as

$$\hat{Y}(t) = P\big(\xi_{YOLO}(t), \xi_{regr}(t) \big| D\big) \tag{4.18}$$

For each prediction given as $\widehat{y_\imath}$

$$\widehat{y_\imath} = \{\widehat{x_\imath}, \widehat{y_\imath}, \widehat{w_\imath}, \widehat{h_\imath}, \widehat{c_\imath}, \widehat{r_\imath}\}. \tag{4.19}$$

YOLO is divided into an $S \times S$ grid of cells who propose the anchoring information of object bounding boxes and the object class. It is expected that each cell can interpret the information within the boundary of cells only, and may have a limited understanding of what other cells may have. The bounding box anchoring information is a collective result over a number of cell information. For this reason, the cell sizes may have impacts on the performance of YOLO.

In this project, we experimented on a number of base networks, including AlexNet, VGG, ResNet, GoogleNet for regression on six types of fruits. It is likely that each type of fruit has unique features distinct from others, the extracted features should be processed by dedicated regression convolutional neural network.

The base networks (AlexNet, VGG, ResNet, GoogleNet) are believed to be conducieve to feature extraction for fruit freshness, in the final fully-connected layers, modifications have been made to the number of neurons to fit our fruit freshness regression problem better. An additional four-layer fully connected network cascaded to the base networks are built. The figure below illustrates this process.

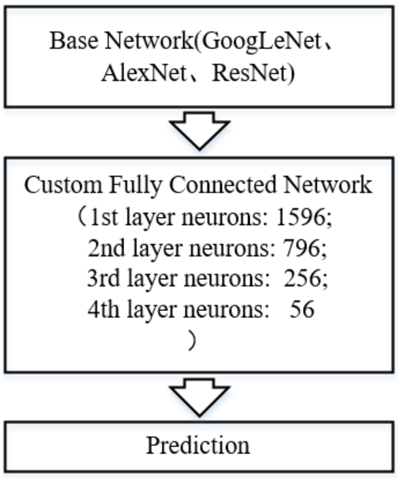

Base Network(GoogLeNet、AlexNet、ResNet)
Custom Fully Connected Network
（1st layer neurons: 1596;
2nd layer neurons: 796;
3rd layer neurons: 256;
4th layer neurons: 56
）
Prediction

Figure 4.3: custom regression model (AlexNet, GoogleNet, ResNet)

VGG is known for its profound depth and has rich features in comparison to others. In order to capture the extracted visual features, here include five fully connected layers, one more than the previous custom dense network.

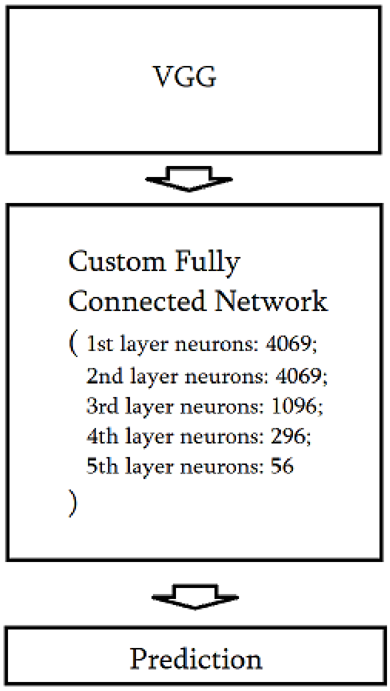


Figure 4.4: custom regression model (VGG)

Dropout (Warde-Farley, Goodfellow, Courville, & Bengio, 2013) is a regularization technique that prevents overfitting. The motivation is that neuron co-adaptations often lead to overfitting, that after a number of epochs of training, neurons have learned the features with associated activation energy. However, there exists a problem regarding extendibility of the captured features against the generalization of such features. To address this issue, the dropout mechanism proposed a stochastic neuron energy removal system to enhance robustness of a network when facing input of various yet similar features. In the custom fully connected network, the dropout mechanism has been implemented to prevent overfitting.

In addition, we conclude the levels of significant impacts on results when utilizing different activation functions, and record the performances of different activation functions.

Chapter 5 Deployment

In this chapter, the details of how our approach is used to resolve this problem are provided, by depicting the program running environment and data training as well as validating. The deployment issues are summarized in three aspects: Python environment, OpenCV, and PyTorch. The pseudo-code assists us in delineating the algorithms.

5.1 Execution Environment

Python (Rossum, 1995) is a general-purpose programming language. Python emphasizes code readability with a strong resemblance to natural English language. In addition to that, Python is interpreted and dynamic that allows for fast project development. In the recent release of python, new features are added to contain a limitation introduced by dynamic typing, that python permits annotation which is a data type declaration realized in compiling.

Established for data analytics, anaconda is a Python and R distribution aiming to simplify package management and deployment. Anaconda has many default applications dedicated to scientific computing tasks, e.g. Spyder and Jupyter Notebook.

OpenCV is a popular computer vision framework, initially developed by Intel, when facing computation optimization problems for CPU-intensive work (Kaehler & Bradski, 2016). OpenCV improves alongside with advancement of computer vision research development, it has traditional computer vision algorithm such as Canny detector in the recent release, it has built-in deep learning networks. The supported programming language is C++ with bindings of Python, Java and MATLAB. To leverage the computation power of GPU (SIMD, Simple Instruction Multiple Data), the CUDA-based and OpenCL-based GPU interfaces are used in the progress of the model development.

PyTorch is a deep learning framework with embedded datatype tensor (a multi-dimensional matrix datatype with built-in support for computation-intensive tasks) and auto-differential mechanism for deep neural networks (Ketkar, 2017). PyTorch can operate CUDA-based Nvidia GPU with mass parallelism. The PyTorch neural network module simplifies building computational graphs and gradient calculations.

5.2 Linear Predictor Constructor

To construct the linear predictor

$$\hat{y} = k_1 J + k_2 B + b, \tag{5.1}$$

where J and B are image entropy and brightness, the following configurations are specified for regression:

- Batch size: 1

- Iteration limit: 1000
- Initial k_1, k_2 and b: $[0.0, 0.0, 0.0]$
- Termination MSE: 0.0001
- Training / Validation split: 90% / 10%

The pseudo code goes as

```
define source_images $D$; // D is of the size of [[numSamples, width, height, channel], label];
                          // label for each source image contains information of an object's location, class and
                          // ground-truth human-rated freshness level
$J$ = calculate_entropy($D$);
$B$ = calculate_brightness($D$);
$[k_1, k_2, b]$ = training_linear_regression($k_1 J + k_2 B + b$, $D.label$); // to train $a_1 J + a_2 B + b$ against
                                                                           // ground truth $D.label$
```

5.3 YOLO+Regression CNN Training

The model framework is PyTorch, which manages forward and backpropagation automatically. For this purpose, one loop of information flow can be quickly processed within each batch data.

The pseudocode, shown in Algo. 5.1, is the training process for how the prepared source data are fed into YOLO for classification and object localization into respective regression CNNs for freshness estimation. The intermediate results, e.g., loss and output, are recorded and plotted to show how the models have converged during the training period. The training configuration is listed as:

- Iteration limit: 50
- Batch size: 4
- Learning rate: 0.05
- Learning rate scheduler: 85% learning rate retention at every iteration and constant after 30 iterations
- Momentum: 0.9
- Optimization: SGD
- Input image resize: $224 \times 224 \times 3$

- Training / Validation split: 90% / 10%

The pseudocode is provided with comments on the declared variables explaining the semantic contribution to, and how they have participated in the training period.

Algorithm 5.1

```
define source_images D; // D is of the size of [[numSamples, width, height, channel], label];
                        // label for each source image contains information of an object's location, class and
                        // ground-truth human-rated freshness level
define YOLO_pre_trained modelYOLO; // The YOLO is pre-trained on over 1 million common object images
                                   //  by the YOLO designer himself. The designer claimed in his research
                                   // that the pre-trained YOLO had learned most common object features.
                                   // We examined the pre-trained object features and found that fruit
                                   // objects were included.
define learning_strategy lrnStratgyYOLO; // Given the pre-trained YOLO, the learning task becomes a transfer
                                         // learning problem. We applied a gradual decreasing learning rate
                                         // scheme.
training_D, validation_D = split(D, 0.9); // 0.9 is the splitting ratio
for epoch in range(50): // range(50) is a vector [1, 2, 3, …, 50]; indicated for the number of total epochs
  for batch in training_D: // here each batch contains 4 samples
    training_output= modelYOLO (batch, lrnStratgyYOLO);
    training_loss=evaluate(training _output);
    backpropagation(training_loss, lrnStratgy);
  end_for
  // for validation dataset, we only need to evaluate their performance given the trained model
  for batch in validation_D:
    validation_output= modelYOLO (batch);
    validation_loss=evaluate(validation _output);
  end_for
end_for
// regressionCNN is a dictionary in which each fruit type associates with a regression CNN
define regressionCNN regressionCNN = {`apple`: regressionCNN,
                                      `banana`: regressionCNN,
                                   `kiwi`: regressionCNN,
                                   `orange`: regressionCNN,
                                   `pear`: regressionCNN,
                                   `dragonfruit`: regressionCNN};
define baseCNN baseCNN = {googleNet, resNet, alexNet, vgg11}; // the base networks are pre-trained as
                                                 // well on over 1 million images and
                                                 // fruit object features are considered.
define learning_strategy lrnStratgyRegr; // same as what have been implemenmted in the etraining of YOLO,
                                 // that a gradual decreasing learning rate scheme was adopted.
cropped_D = crop(D); // only the regions of interest are fed into regression CNN
training_cropped_D, validation_cropped_D = split(cropped _D, 0.9);
for eachBaseCNN in baseCNN:
  regressionCNN = eachBaseCNN; // here the base CNN is applied to all six fruit regression CNN
  for epoch in range(50):
    for batch in training_cropped_D: // here each batch contains 4 samples
      batch_fruitType = getFruitType(batch); // here the training of regression CNN for fruit freshness
                                       // grading targets image data of same labels
      switch batch_fruitType:
      case fruitType:
        training_output = regressionCNN[fruitType](batch, lrnStratgyRegr);
        break;
      end_switch
      training_loss=evaluate(training _output);
      backpropagation(training_loss, lrnStratgy);
     end_for
     for batch in validation_cropped_D:
      batch_fruitType = getFruitType(batch);
      switch batch_fruitType:
        case fruitType:
          validation_output = regressionCNN[fruitType](batch, lrnStratgyRegr);
          break;
```

```
            end_switch
          validation_loss=evaluate(validation_output);
      end_for
  end_for
end_for
```

Chapter 6 Results

In this chapter, we discuss the results of the proposed models. As explained, the model is constructed in a hierarchical structure, consisting of a classification-and-localization-purposed model (YOLO) and a set of regression CNNs for each class. The convergences of these base models are shown in this chapter, while the metrics are offered to demonstrate the robustness of these models. The performance based on the metrics is to show what have contributed to this convergence.

6.1 The Results of the Proposed Linear Regression Model

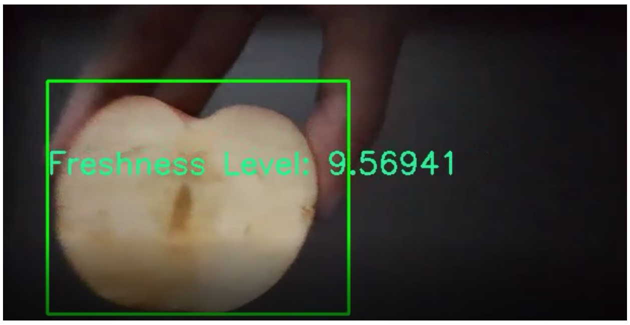


Figure 6.1: The prediction of fruit freshness levels through simple entropy/brightness-based approach

Figure 6.1 shows an example of fruit freshness grading. It is noticeable that any insignificant changes in the ambient envoirement, e.g., exposure to different ambient lighting environments, lead to significant changes in entropy and brightness, which leads to inaccurate grading for fruit freshness analysis.

The average brightness and entropy are calculated for frames in each video. Pertaining to the images with complicated background noises, the localization can hardly work, and the brightness/entropy approach does not converge as expected. The defined freshness function is shown in eq.(6.1),

$$freshness = k_e\ entropy(I) + k_b\ brightness(I) + b \tag{6.1}$$

where it seem not to have a linear relationship with entropy and brightness of the image. The configurations of a linear regressor are shown as:

- k_e: -2.7701
- k_b: 0.00367
- b: 9.0004

It is evident that there hardly exists a linear relationship as expressed in eq.(6.1). Our assumption that decay appears with overall fruit image brightness going dark and increases in entropy, works on images with simple noise background only. For example, for the first two apple images shown in Table 6.1, a raised entropy is observed as well as a decreased brightness level. The two images are of similar backgrounds (at least based on human understanding, the brightness levels are close to each other). It is observable that for the apple with a low grading of freshness, the entropy saw a 19%

increase. As reflected in the local entropy description, this apple has more detected object edges.

However, this approach is subject to background noises, even if a minor change of background might result in significant errors. During this experiment, different physical backgrounds were set up when taking pictures of the fruits, including but not limited to set up different ambient light conditions and place adjacent foreign objects.

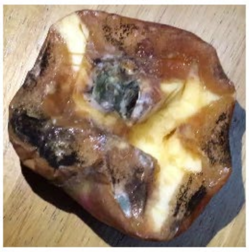
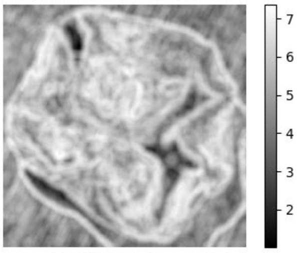
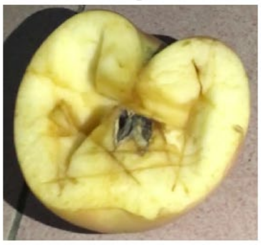
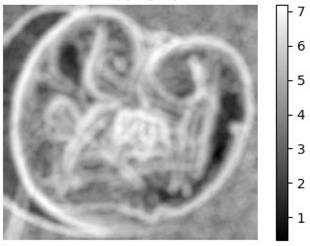
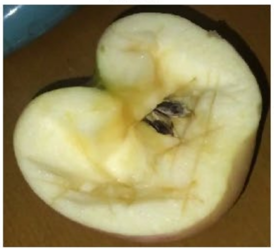
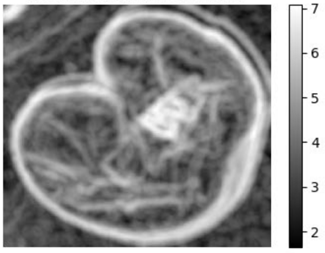
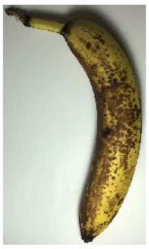
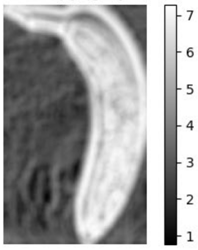
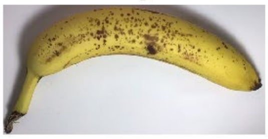
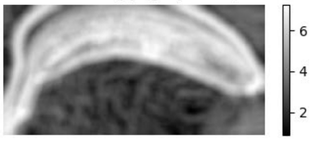
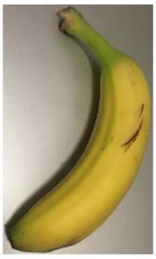
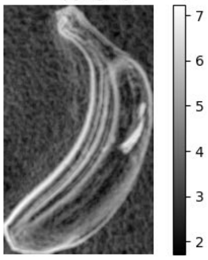
It is evident that the three bananas as shown in Table 6.1 were placed on the same physical platform (a plain white colour table) but with various levels of ambient lighting conditions, the entropy levels vary significantly. In addition to that, the brightness levels seem to be easier to be influenced by ambient lighting conditions than dark spots introduced by decaying.

Another issue to take into account is that this assumption (entropy/brightness) is only partially right. For fresh fruits such as apples and banana, observations are clear that there exist correlations between entropy/brightness levels and decay stages when the background is set static; however, for other fruits such as kiwi fruits and oranges, this assumption is hardly correct.

Kiwi fruits and oranges are by nature having rich texture that introduces high entropy value, the growth of spots may have uniform colours and textures that reduce entropy value. For dragon fruits, our experiment shows that the decay process results in changes of colour only, and the entropy/brightness approach might be incorrect in nature to this type of fruits.

This preliminary approach through entropy/brightness reveals the complexity of fruit freshness grading. Different fruits have their own processes of decaying, for each decay characteristic, there is no apparent relationship between static visual features (a set of defined rules of pixel statistics) and freshness levels. Based on these discoveries, it is only reasonable to assume each type of fruit be treated individually rather than by a comprehensive approach.

Table 6.1: Entropy/brightness approach result examples

Image	Local Entropy (grey scale)	Statistics
	 	Freshness Level: 1.5 Average Entropy: 4.584 Average Brightness: 127.5
	 	Freshness Level: 7.085 Average Entropy: 3.825 Average Brightness: 157.2
	 	Freshness Level: 8.290 Average Entropy: 3.416 Average Brightness: 123.6
	 	Freshness Level: 3.967 Average Entropy: 3.361 Average Brightness: 163.6
	 	Freshness Level: 5.978 Average Entropy: 3.420 Average Brightness: 176.3
	 	Freshness Level: 7.900 Average Entropy: 2.935 Average Brightness: 137.2

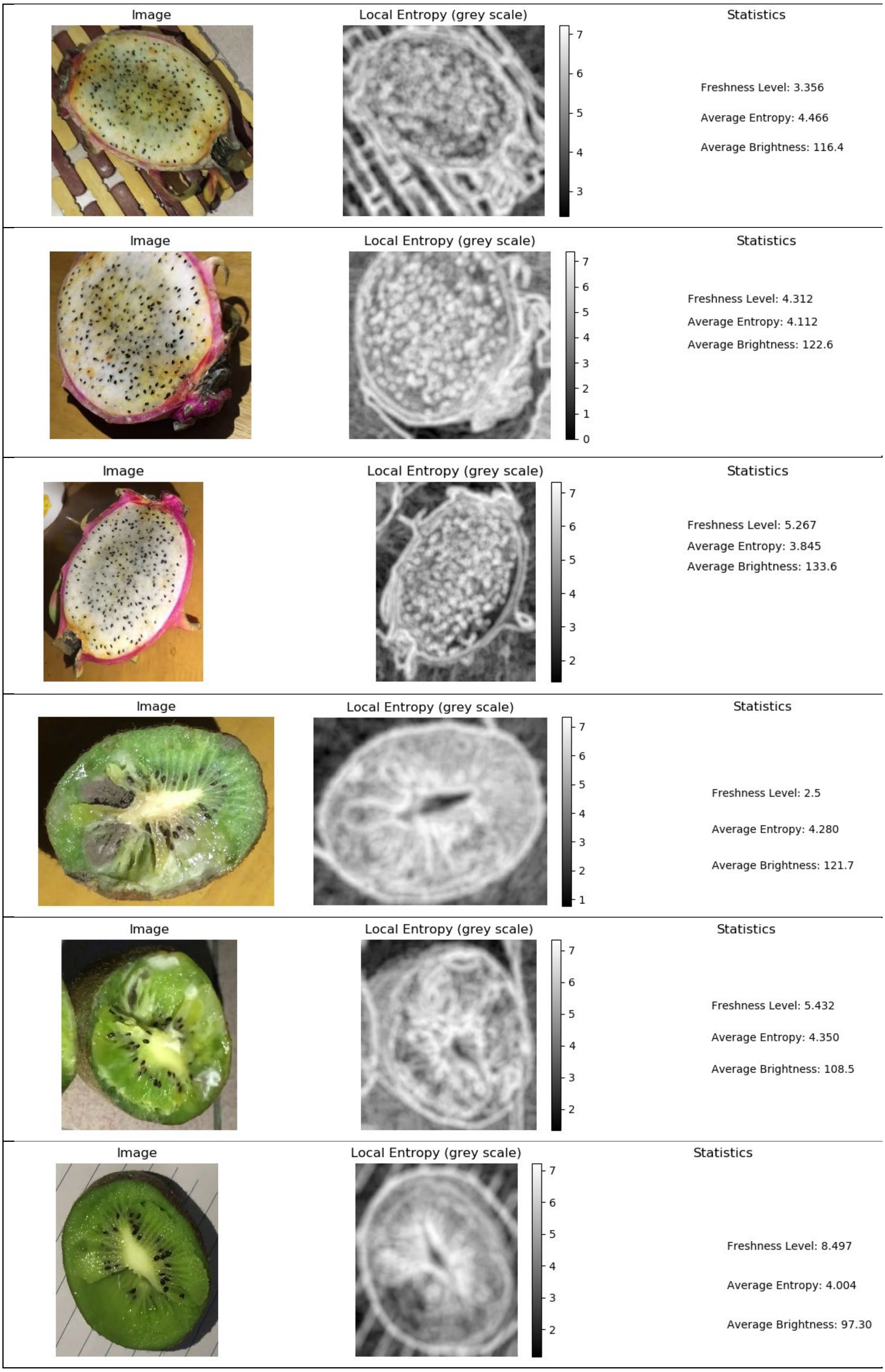
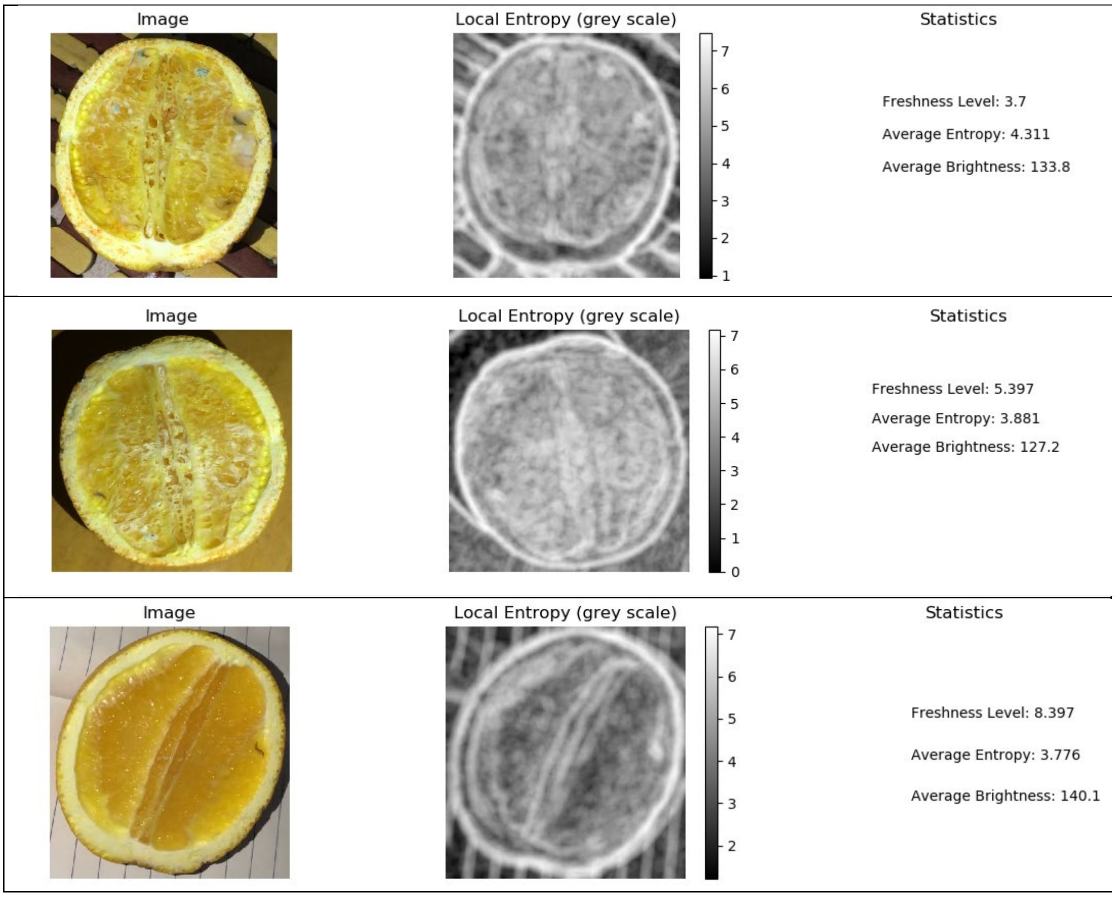


Image
Local Entropy (grey scale)
7
6
5
4
3
Statistics
Freshness Level: 3.356
Average Entropy: 4.466
Average Brightness: 116.4
Image
Local Entropy (grey scale)
7
6
5
4
3
2
1
0
Statistics
Freshness Level: 4.312
Average Entropy: 4.112
Average Brightness: 122.6
Image
Local Entropy (grey scale)
7
6
5
4
3
2
Statistics
Freshness Level: 5.267
Average Entropy: 3.845
Average Brightness: 133.6
Image
Local Entropy (grey scale)
7
6
5
4
3
2
1
Statistics
Freshness Level: 2.5
Average Entropy: 4.280
Average Brightness: 121.7
Image
Local Entropy (grey scale)
7
6
5
4
3
2
Statistics
Freshness Level: 5.432
Average Entropy: 4.350
Average Brightness: 108.5
Image
Local Entropy (grey scale)
7
6
5
4
3
2
Statistics
Freshness Level: 8.497
Average Entropy: 4.004
Average Brightness: 97.30



6.2 The Performance of YOLO Classification

The metrics for performance evaluation are accuracy, precision, and recall for YOLO classification and MSE for the bounding boxes to which degree the object is contained. YOLO might consider multiple bounding boxes for one image with only one object of interest present. The metrics for YOLO classification in fact measure the classification of the drawn bounding boxes.

Table 6.3 shows the classification results by using YOLO. All performances of fruit classification are analyzed, during the training and validation. The metrics for average performance of all six fruit species are calculated to evaluate the overall performance of YOLO for the task of classifications.

Table 6.2: The metrics for evaluating performance of YOLO-based classifications

Fruits	**Accuracy**		**Precision**		**Recall**	
	Training	Validation	Training	Validation	Training	Validation
Apple	0.9352	0.9251	0.8509	0.8504	0.9387	0.9195
Dragon fruit	0.9620	0.9572	0.9330	0.9236	0.9295	0.9283

Kiwi	0.9450	0.9432	0.8841	0.9059	0.9313	0.9003
Pear	0.8498	0.8200	0.7192	0.6656	0.6816	0.6376
Banana	0.9978	0.9972	0.9943	0.9927	0.9978	0.9975
Orange	0.8525	0.8464	0.7442	0.7057	0.7001	0.6947
Average	0.9237	0.9149	0.8542	0.8407	0.8631	0.8463

The results for the fruits freshness grading show that the banana is the fruit which is the most distinct one from others while oranges and pears are the fruits having been least recognized, where the banana has the highest accuracies, precisions, and recalls for both training and validation sets. This is the opposite for pears and oranges, that the two fruit species scored the lowest among the six types of fruit. The metrics of the apple, kiwi fruit, and dragon fruit closely follow that of banana with small drops between $3\% - 6\%$ on accuracy.

All fruits received good recognition results with the highest accuracies up to 99% for both the training and validation sets, the lowest one is at 85% and 82% for training and validation, respectively.

The average performance of the YOLO classifier is displayed in accuracy, precision, and recall metrics, the scores are above 90% for accuracy and 80% for precision and recall.

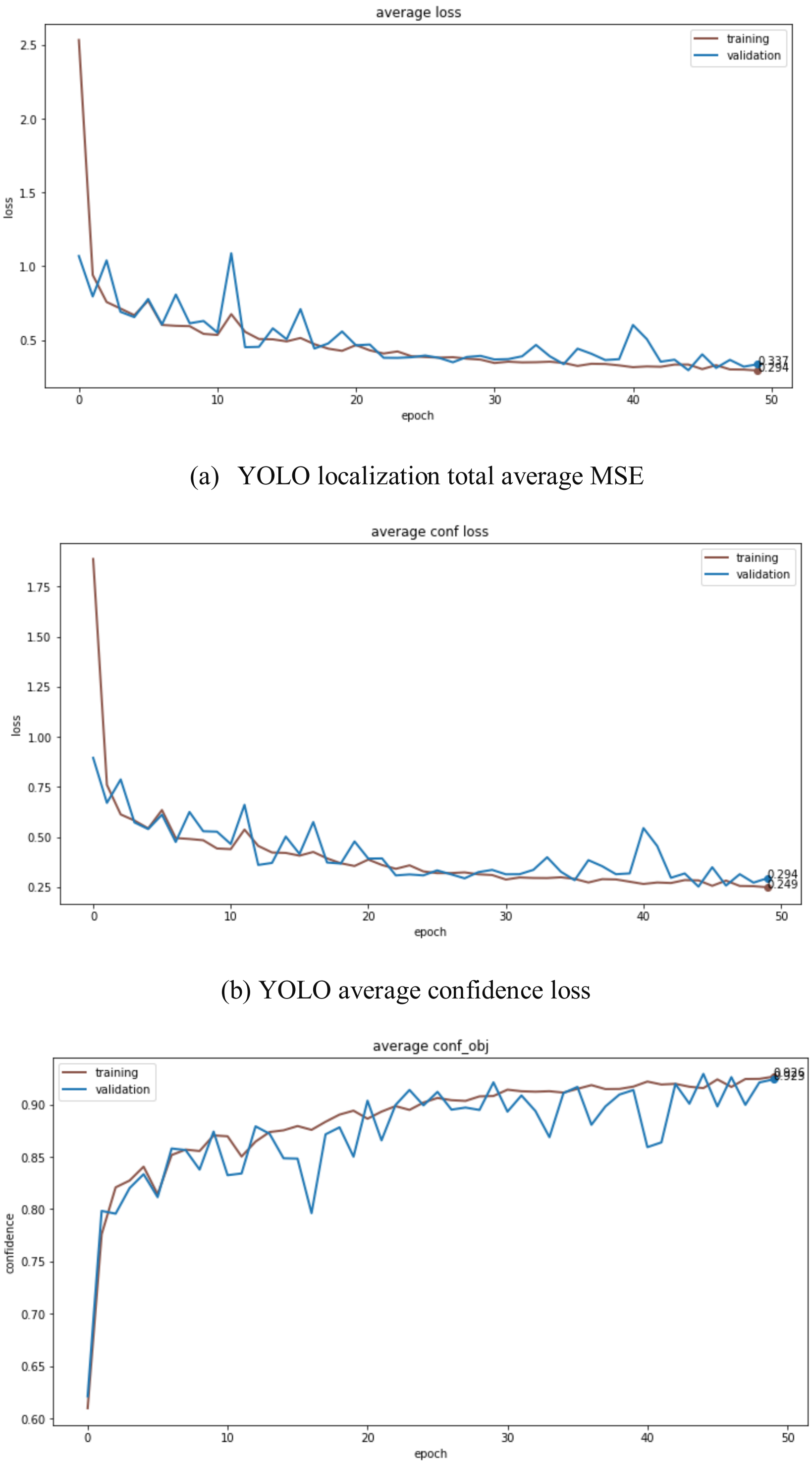
There is no significant gap between the training and validation by using performance metrics. This indicates low probabilities of YOLO being overfitted during the training session.

6.3 YOLO Localization Performance

In order to evaluate the performance YOLO localization capability, the loss MSE for the general loss of the predicted bounding boxes are computed. The loss function is given as eq.(6.2).

$$loss_{total} = loss_x + loss_y + loss_h + loss_w + loss_{cls} + loss_{conf} \tag{6.2}$$

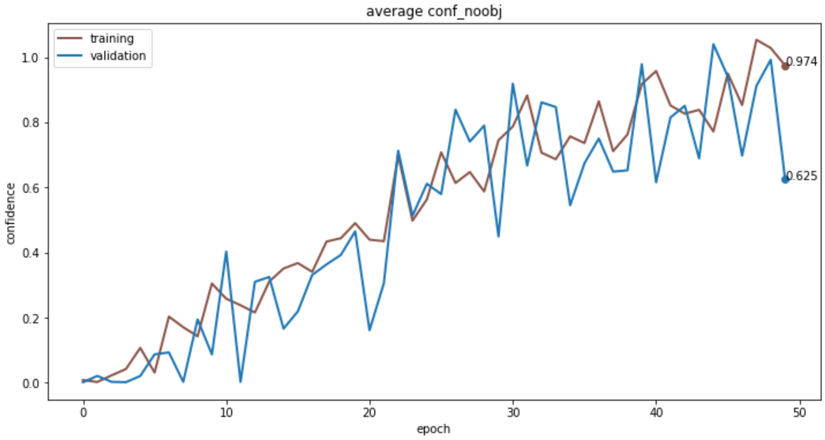
where the subscripts are from the centroid (x, y) of bounding box, hight h and width w, the object classification error cls , and the confidence $conf$. The loss convergence is shown as Fig.6.2. The final loss results are 0.294 and 0.337 for the training and validation, respectively.



(a) YOLO localization total average MSE

(b) YOLO average confidence loss

(c) Average confidence of YOLO including an object



(d)YOLO average confidence of rejecting containing an object

Figure 6.2: The losses of YOLO model

The metrics include the loss to the confidence, positive confidence pertains to an object, and the confidence of rejecting presence of an object inside the predicted bounding box. YOLO converges over the training period in terms of average confidence loss, despite small fluctuations. The final loss values for the training and validation are settled closely.

During the entire training period, YOLO grows in the confidence with a right bounding box. The training convergence trend is smooth compared to that of the validation set. Figure 6.2(c) shows how confident the YOLO is to reject the existence of an object inside the predicted bounding box. Both the training and validation processes display relatively high fluctuations in contrast to tha confidence of an object within a predicted bounding box.

6.4 The Performance of Regression CNN

Inspired by the approach that each type of fruit has its own way of decaying, there appears a lack of universal features, we developed a deep learning model to regress only one kind of fruits. For evaluating the newwork, we use Mean Squared Error (MSE) to measure the average loss of the prediction, standard deviation to measure how stable the prediction is. For classification, we calculate the average accuracy, recall, and precision by the end of training for all types of fruits. We experimented on various deep learning structures.

In addition, feature filters at the first layer were operated on the whole image. The features learned on the first layer often preserve semantic geometric properties that resemble the visual features in source images (Uchida, Tanaka, & Okutomi, 2018). It is observed that most feature filters see similarities with fruit objects.

We tested all six types of fruits on GoogLeNet. For the first batch in the first epoch of training, our records show the MSE of all six types of fruits is greater than 100 then reduced to below 5. The convergence of the training dataset and validation dataset demonstrates that the proposed prediction models are trained to have a better capability of recognizing freshness/festering features. On average, the performance of GoogLeNet-based regression CNN is at 3.625 (MSE) for training and 4.404 (MSE) in for validation. Generally, the fruit freshness grading by using GoogLeNet inherently has about two degrees of errors based on 10 degrees of fruit freshness grades. Standard deviation indicates output stability, where based on average googleNet, the deviation is 1.323 for the training and 1.500 for the validation.

In GoogLeNet, different types of fruits show their degrees of regression on grading fruit freshness. Banana is the most accurately predicted type of fruit grading while kiwi fruits are the most difficult one. Apple freshness grading appears the most unstable one in the validation, the difference is 2.722. This can be traced back to the features of spoiled apples that apples have rich features when decaying, in comparison to other fruits with relatively universal rottenness features, e.g., the skin of dragon fruit covered by yellowish dark spots.

Table 6.3: The metrics for evaluating the performance of GoogLeNet

Measurement Items	**MSE**		**Standard Deviation**	
	Training	Validation	Training	Validation
Apple	4.499	4.653	2.082	2.722
Dragon fruit	2.629	2.926	1.065	1.725
Kiwi	5.810	5.997	1.172	1.430
Pear	4.250	5.958	2.045	1.705
Banana	1.661	1.705	0.967	0.964
Orange	2.905	3.005	0.606	0.451
Average	3.625	4.404	1.323	1.500

We extract the features at the first layer from GoogLeNet (there are 64 features at the first layer, with the size $7 \times 7 \times 3$ of each 3 channels filter). Due to the existing weights

(GoogLeNet's features are pretrained), the fruit freshness features for learning show the high similarities with pretrained features.

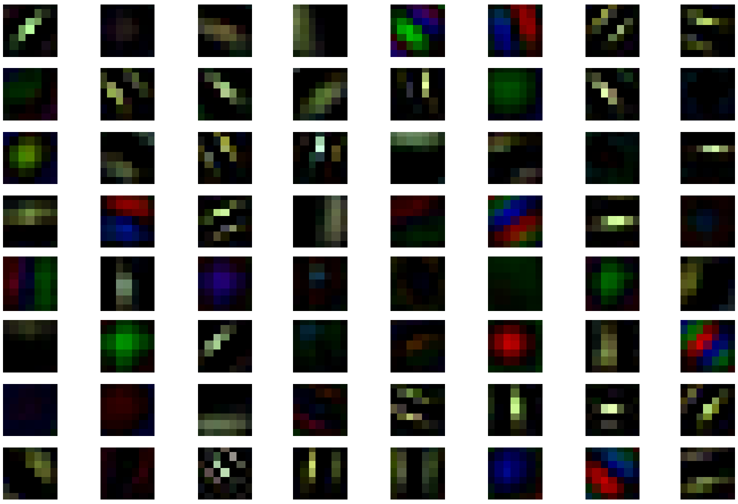


Figure 6.3: The fruit freshness features from GoogLeNet

The metrics for GoogLeNet evaluations with regard to oranges show that both the training and validation procedures are convergence, despite of minor fluctuations of the validation process. The freshness grading of Kiwi fruits by using GoogLeNet is stable in comparison to other fruits. Both the training process and validation process display convergence.

Despite great fluctuations in the early training period/epochs, the performance of regression for the validation by using GoogLeNet for bananas is convergence in the end of the training.

The metrics for evaluating the performance of googleNet based on apples dataset exhibit more fluctuations than aforementioned fruit types. However, the convergence for training and validating process is still visible.

The pear freshness grading by using GoogLeNet is not stable throughout the training period. In addition, a large gap between the training and validation process is evident.

The freshness grading for dragon fruit by using GoogLeNet shows the fluctuations during the process of validation. However, the training procedure exhibits its stability during the training.

The test on AlexNet reveals that the performance of AlexNet for the six types of fruits is similar to other base network regarding on which type of fruit the regression is prone to suffering from deviating with the ground truth. Apple, Kiwi fruits and pears are the three most challenging ones to regress while banana grading is the most accurate one. Fruits grading with relatively large errors tends to be less stable when regressing. This is evident in both training and validating procedure of classification with all types of fruits. AlexNet performed neither good nor bad in contrast to other three base networks.

The average MSE for all six types of fruits is 3.500 for training procedure and 4.099 for validating. In terms of regression stability, this base model reports 1.480 for the training and 1.248 for the validation.

Table 6.4: AlexNet performance metrics

Measurement Items	**MSE**		**Standard Deviation**	
	Training	Validation	Training	Validation
Apple	4.974	4.987	1.687	1.497
Dragon fruit	2.658	2.794	1.247	1.686
Kiwi	4.279	5.664	2.422	0.893
Pear	4.250	5.958	2.045	1.705
Banana	1.696	1.818	0.793	0.892
Orange	3.139	3.368	0.687	0.816
Average	3.500	4.099	1.480	1.248

The first layer feature filters are with the size $11 \times 11 \times 3$. The total number of the filters is 64.

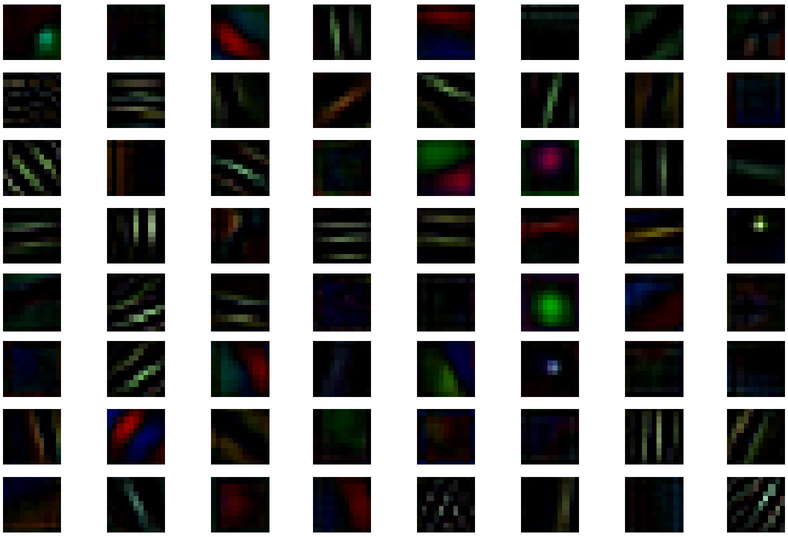


Figure 6.4: The fruit freshness features at the fist layer of AlexNets

The pear freshness grading by using AlexNet shows less-noticeable convergence in contrast to other base networks, despite the training set displays uniform convergence throughout the training period.

The performance of AlexNet for freshness grading regarding Kiwi fruits exceeds other base networks. Not only does the base network displays quick convergence for the entire training period for the training, but also it shows even better convergence results on the validation set.

Although the convergence of the training procedure is stable, the one for the validation displays saliement fluctuations. AlexNet appears already converged at the first epoch in the training period on the validation set. However, the convergence trend indicated by the training set shows this base network still converges.

Similar to the convergence by using AlexNet on dragon fruits, the trend for apples grading does not appear convergence on the validation set, but the training set shows the base network does converge.

The convergence for orange freshness grading by using AlexNet has high fluctuations while the training tends to be more stable during the later period of the training.

The performence of ResNet-152 is the top one among the ResNet family, as well as the deepest network among the ResNets. Again, ResNet fails to deliver good results based on three particular types of fruit images: Apples, Kiwi fruits, and pears. The regression error is large on the kiwi fruit images, both on the training and validation. For pears, there exists a possibility of overfitting as the validation set shows 6.057 while the training set reports 3.984. Banana freshness grading is the most accurate. In terms of regression stability, pears are the least stable while oranges are the most (judging by careful evaluations of training and validation sets).

On average, MSE values of training and validation for ResNet-152 are 3.582 and 4.058, respectively. For stability measurement, the standard deviation shows 1.329 for the training set and 1.842 for the validation set.

Table 6.5 The metrics for evaluating the performance of ResNet

Measurement Items	**MSE**		**Standard Deviation**	
	Training	Validation	Training	Validation
Apple	4.226	4.374	2.029	2.188
Dragon Fruits	2.634	2.815	0.913	0.840
Kiwi Fruits	6.034	5.765	1.467	1.417
Pear	3.984	6.507	1.936	4.899
Banana	1.636	1.659	0.984	0.864

Orange	2.982	3.233	0.645	0.847
Average	3.582	4.058	1.329	1.842

The fruit freshness features at the first layer of ResNet-152 are shown in Figure 6.5. Each feature is presented in the size of $7 \times 7 \times 3$ in adjusted RGB scale from normalized weights.

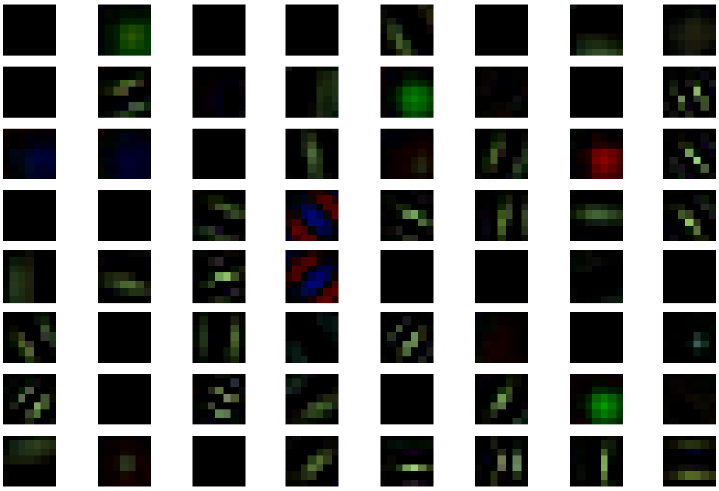


Figure 6.5: The fruit freshness features from the first layer of ResNet-152

ResNet-152 displays high fluctuations on the validation despite of apparent convergence trend during the training period. The regression is not stable as indicated by the standard deviation.

ResNet152 shows convergence in the early period of training on the validation set, on the training, it is stable. Regression stability, as indicated by standard deviation, remains flat with small degrees of fluctuation.

The convergence is weak for the validation set but still visible. At the final epoch of training, the resulting gap is small in comparison to other base networks'.

Despite of having relatively high degrees of error (MSE) in contrast to other base networks, the convergence is evident for both training and validation.

Bananas are the most accurate in freshness grading regression. Despite fluctuations, both training and validation sets show convergence during the training and are stable as indicated in standard deviation.

Fluctuations of the convergence trends on the validation set are observable while the trend on the training set is flat. Both show convergence during the training period.

For VGG-11, again, banana degrading is the most accurate one in freshness grading, while the images of apples, kiwis and pears are the most difficult. However, VGG-11 tends to suffer less from overfitting as indicated in the metrics where the result gaps between the training and validation sets are small. VGG-11 displays high stability in regression, where even for the images of apples, kiwis fruits and pears, both training and validation display robust regression output in standard deviation. The average training and validation MSEs are close to the other three base networks.

On average, the MSEs for training and validation are 3.665 and 3.934, respectively, the standard deviations are 1.361 and 1.266, respectively.

Table 6.6: The metrics for evaluating the performance of VGG-11network

Measurement Items	**MSE**		**Standard Deviation**	
	Training	Validation	Training	Validation
Apple	4.504	4.625	2.038	2.078
Dragon fruit	2.823	3.129	1.374	1.129
Kiwi	5.726	5.670	1.546	1.101
Pear	4.226	5.341	1.717	1.712
Banana	1.796	1.831	0.844	0.607
Orange	2.900	3.012	0.647	0.967
Average	3.665	3.934	1.361	1.266

The features for fruit freshness grading by using VGG-11 network are of the size 3 × 3 × 3. Despite the filter size is small, the learned features are similar to what other networks have produced, mostly greenish/yellowish hues that draw similarities with most fruit natural visual displays.

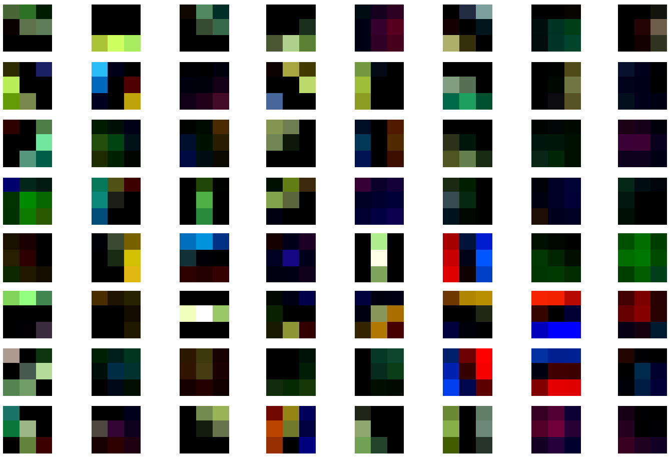


Figure 6.6: The features of fruit freshness at the first layer of VGG 11 network

The validation for dragon fruit degrading is less converged compared to the training, as same as other base networks based on different types of fruit images. However, the network remains converged as indicated by the training set.

VGG-11 network based on the orange dataset shows convergence but less evident on the validation set. However, the output is stable as shown in the standard deviation plot.

Similar to other performance metrics, VGG-11 network for Kiwis fruits grading exhibits strong convergence during training but displays fluctuations not evident to a sign of convergence.

The banana training is accurate in regression as shown in the convergence and is stable in output. Despite fluctuations, the validation converges by using the VGG-11, this is evident in the training set. However, the output is not stable during the training and validation.

6.5 A YOLO Regression CNN Demonstration

This demonstration employed YOLOv3 for localization and classification, GoogLeNet for freshness-related feature extraction and grading score regression. The results are consistent with aforementioned metrics that banana yields the best results in terms of classification, while others failed at various degrees.

The background we setup for this demonstration is totally foreign to the training and the validation (e.g., no white shirt backgrounds are present in our source dataset).

Figure 6.7 shows the localization and freshness grading result of a fresh banana image. The bounding box surrounds the region of interest despite trivial inclusion of irrelevant areas. The freshness score is above 8.0 and can be regarded fresh.

During the tests, banana exibits less misclassification results in comparison to other fruit species. This is likely a result of banana distinct morphological features in contrast to others of an elliptical shape.

The experiment of an orange prediction was conducted. The background is a wooden shelf against a white-painted wall. The predicted bounding box has full occlusion of the fruit object with one part of the wooden beam enclosed. The prediction for fruit species and freshness is accurate.

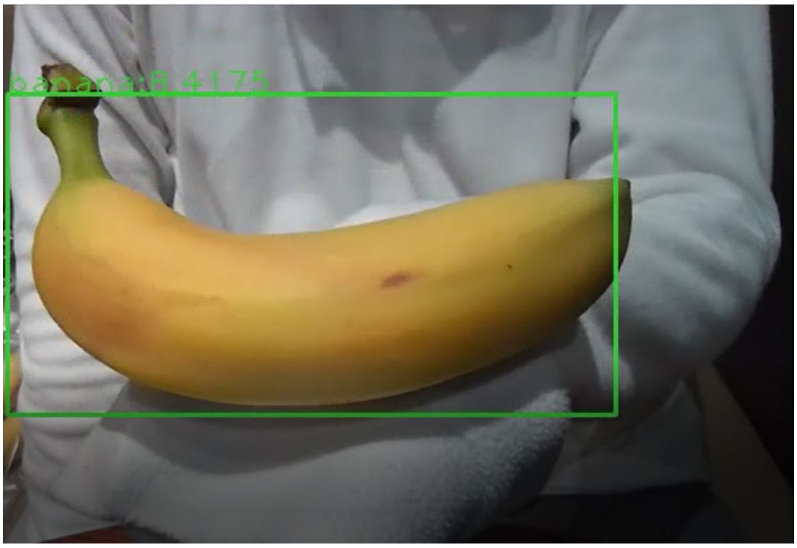


Figure 6.7: A test on banana localization, classification, and freshness grading

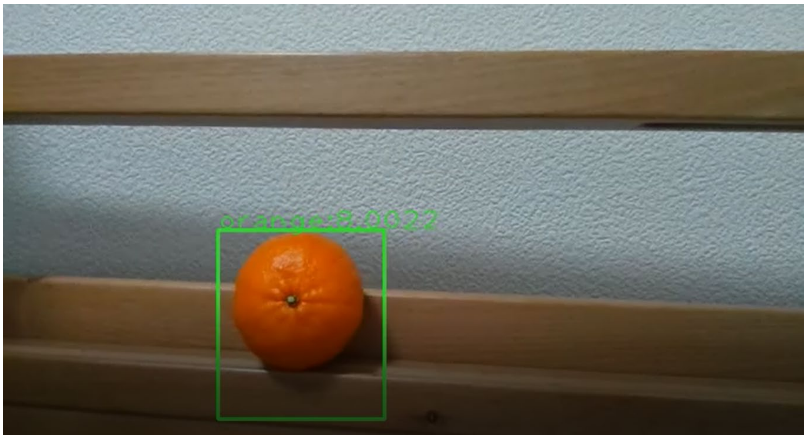


Figure 6.8: A test on orange localization, classification and freshness grading

The orange for the second test is visually decayed in contrast to the first test on this orange (e.g., observed wrinkled textures and brown spots)

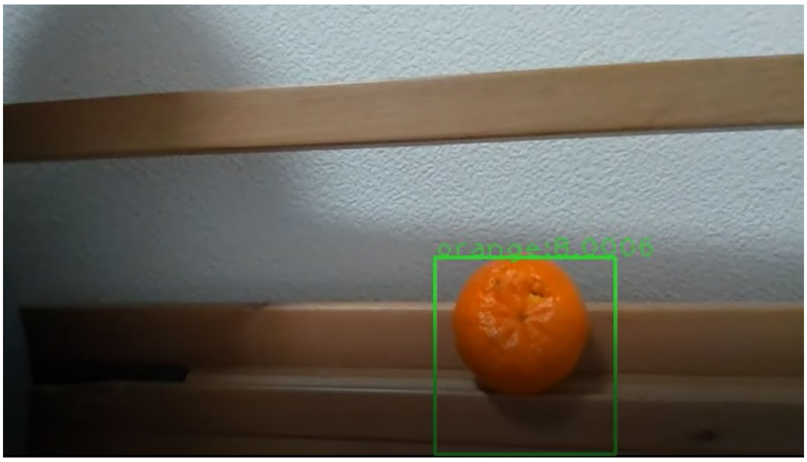


Figure 6.9: A test on an orange for localization, classification, and freshness grading

The next example is that of an apple. YOLO failed the classification of fruit images, but localization, as indicated in the regions of interest, is satisfactory. Dragon fruit is comparable in texture and hue with apples, provided that, in the case of this apple image, they are comprised of dominant red colour and sporadic greenish/yellowish pigments. The freshness grading is accurate in reflecting the overall visual freshness features.

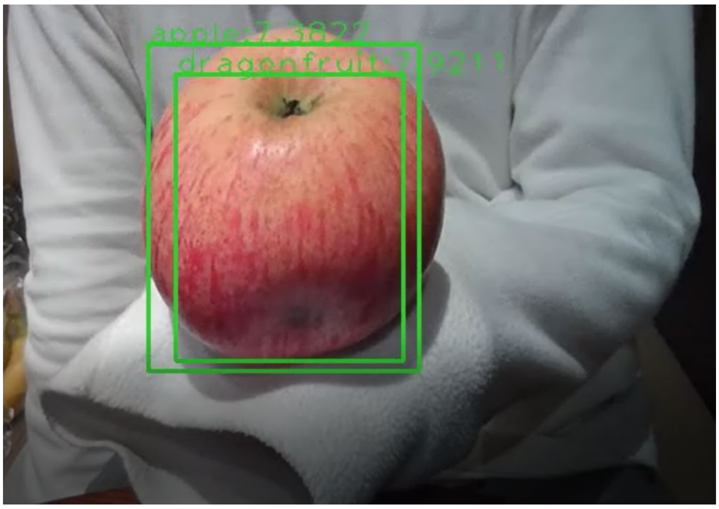


Figure 6.10: A test on apple localization, classification, and freshness grading

The ambient light condition was adjusted and it is noticeable that YOLO is prone to the object as an orange. There is no significant change of grading score. Apple remains as a strong candidate for YOLO prediction.

In this test, two irrelevant objects were introduced: A handle of a chair and a corner of a book shelf, YOLO successfully resists these noises.

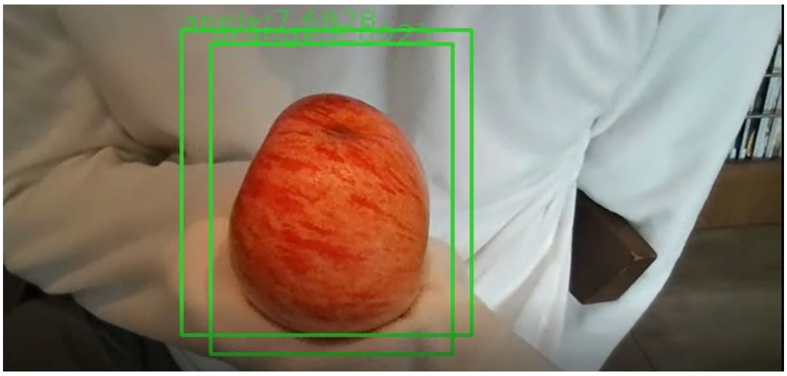


Figure 6.11: A second test on apple localization, classification, and freshness grading

The final test was a placement of three apples on a mat under an unpleasant lighting environment (an intentionally imposed dark ambience) where blurring is observed in the figure. The YOLO succeeded in recognizing two apple objects with negligence of the apple placed in the middle of the image frame. The grading reflect the fruit freshness nature.

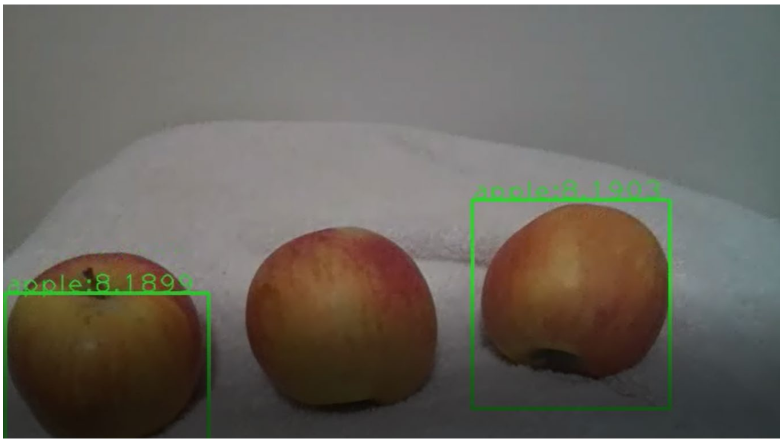


Figure 6.12: A second test on apple localization, classification, and freshness grading

6.6 Reflections

In comparison to what the linear regressor has proposed, all deep learning approaches see significant improvements on all performance metrics. The four deep learning approaches have similar performance with trivial gaps. By training AlexNet in MSE, the validation of VGG export the lowest error. Table 6.7 is a summary of overall proposed model regression performance (measured in MSE).

Table 6.7: A summary of MSE for the performance of various classifiers

Classifiers	**Training**	**Validation**
Linear Regressor	4.749	5.128
AlexNet	3.500	4.099
GoogleNet	3.625	4.404
VGG	3.665	3.934
ResNet	3.582	4.058

The fluctuations shown as in the validation are likely a result from the insufficient dataset, where we collected about 4, 000 images over 6 types of fruit that give about 600 to 700 images for each type of fruit. Under the proposed 10-grade scale, there are only about 60 to 70 images per fruit class. After splitting the data into training and test datasets, for the validation, there are only one or two images per grade for each fruit class. It is reasonable to assume that this contributes to the convergence exhibited in the test.

Another issue is labelling. We proposed a methodology first to map the visual fruit decay features onto a concept to which degree the fruit is considered spoiled, then, invited a small number people to rate the fruit freshness levels so that the libelling would be fair as the freshness grading is of collective intelligence of a group independent raters. The established evidence reveals that the human understanding of fruit freshness could be hardly linear, given two fruit objects with appeared similar spoilage levels, different raters may have distinct ratings. In addition, as observed, human ratings appear to be “fuzzy”.

The grading/labelling process manually can be regarded as of a fuzzy logic, where raters tend to first develop a fuzzy concept of fruit freshness (Singh, et al., 2013), then map the concepts onto numerical ratings, rather than possessing a numeracy mindset to rate the fruit freshness in the first place.

Figure 6.13 shows how the 3rd party raters graded the fruits. It demonstrates that how raters did not regard the decaying process as continuous. When presented with different fruits of similar freshness, the characteristics are visible in distinct decay levels, raters assigned freshness grades distant from the visually resemblant ones. This phenomenon might explain why the MSEs of the regression CNNs fluctuate around 3.0 to 6.0 with a mean of (approximate) 4.0, fruit images with close freshness levels are likely having ground truth with of 2 grades.

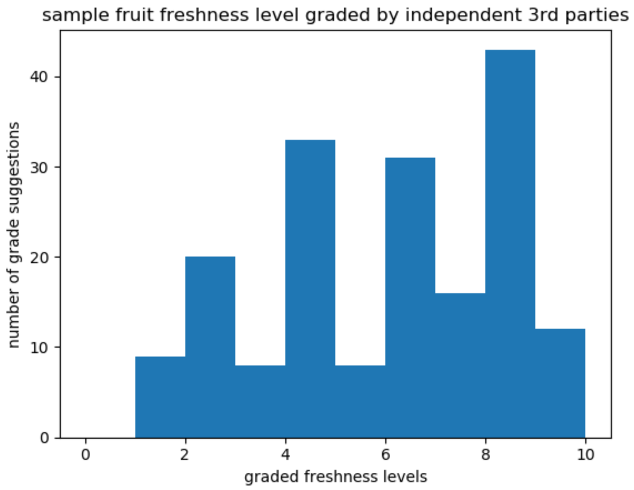


Figure 6.13: The level of fruit freshnesses graded by independent 3rd parties

Despite minor performance gaps, VGG claims the crown for its lowest error in the validation set test, and displayed resistence to overfitting as indicated in the performance metrics. VGG is the heaviest in the number of neural network parameters that is likely an explanation to its output results with lower error rate than others.

VGG is different in the first layer filter size which indicates a possibility of filters' ability of interpretation on source information being dependent on the shape, since the preliminary research discovered the contributions of invididual spoilage features of various sizes to the overall freshness perception, that the granularities of spoilage features require attention and small-size filters can just satisfy that.

The convergence plots of the aforementioned regression CNNs display various convergence trends with low levels of fluctuations and speed, all are converged at the end of iteration. The training trends see faster convergence than that of the validations'.

Another issue discovered during the experiment is the supposed-to-be-better CNN structures, e.g., ResNet, VGG-11 Net, and GoogLeNet, should have outperformed the traditional ones, e.g., AlexNet. This is not the case in this experiment, as all CNN structures have displayed similar performance metrics. This might be an issue resulted from insufficient visual information in the source images, where all CNNs have learned the features of fruit freshness and produced similar results. The fast convergence indicates that the visual features of fruit freshness are not easy to be learned so that the high-level abstract features are similar among all CNNs when making a final prediction on fruit freshness matters.

Chapter 7 Conclusion and Future Work

In this chapter, we summarize the discoveries of our experiments, including how the data collection was approached and the explanation for structure of a hierarchical model comprised of YOLO for classification and localization, and individual regression CNNs for fruit species. Discussions were made on what remains unsolved and should have been finished for expected better performance of the fruit freshness grading system.

7.1 Conclusion

This chapter reviews fruit freshness challenges and the proposal deep learning models. Provided that there lack existing fruit freshness datasets, this thesis firstly discusses on the collection of fruit images, as well as image augmentation and enhancement for expected better performance of the training results. Four augmentation/enhancement schemes are implemented, where fruits are enhanced with balanced contrast and sharpening, and are rotated with a fixed angle and added random noises to prevent overfitting. Image segmentation (the drawing of bounding boxes) for regions of interest extraction is deliberately calculated to respect the balance between maximal inclusion of object information and exclusioin of background noises.

This thesis reveals that the fruit freshness grading is highly nonlinear, as demonstrated in the failed traditional computer vision approach. A naïve linear regression model was constructed to measure critical fruit freshness features (increasing darkening of the fruit skin and variations of colour transitions), conceived in the intuition that fruit spoilage occurs with biochemical reactions that result in such visual feature changes (natural pigment degradation and deformation).

Efforts have been committed to construction of a hierarchical deep learming model, in which fruits are first classified and localized in YOLO, the regions of interest are cropped from the source images and fed into regression CNNs. Trainings on each CNN for each type of fruit for four different CNN structures (GoogleNet, ResNet, AlexNet, VGG11) are independent.

The performances of the aforementioned neural networks are recorded. Strong convergence trends are observed in all base networks. Judged by using the MSE and standard deviation, VGG is at the top of the performance metrics. YOLO demonstrates convergence in increasing confidence scores and dropping error rates over training iterations, the final classification results are recorded for each type of fruits that show high accuracies (all above 80% in) for all six fruit species, where bananas are the most accurate in prediction.

Reflective of the performance results, explanantions are provided for freshness grading factoring in discussions of possible dimensions that contribute to the scoring. Spoilage visual features are complex in nature and granularities collectively form the degradation level in human perception, to which neural networks with different sizes are expected to perform differently in respects to their attentions to trivialities of spoilage visual features. Labelling and source data volume are likely two contributors having significant roles to play to freshness grading.

This experiment includes a python-based application, by which users can switch on webcam for video streaming and fruit object localization, classification and freshness grading are completed automatically. The demonstrations in Chapter 6.5 are extracted frames from videos recorded in different background scenarios.

7.2 Future Work

To develop a more robust and accurate fruit freshness assessment deep learning model, as a common deep learning practice, a large volume of source data is required. The data should include noises. In this research project, each collection of fruit images has about 600 to 700 images, that is, in total, we have about 4000 images over 6 kinds of fruit with each type of fruit being rated in 10 grades.

A 10-level fruit freshness grading mechanism is proposed. In this experiment, there was no engagement of professionals to provide assistance in grading, instead, in order to ensure fairness of the grading/labelling process, we committed the best effort to show how a fruit decays in the intuition of an ordinary way by inviting a 3rd party raters for consulting the grading issues. There is no empirical evidence showing that the judgements are fair and accurate, how much the judgements are compliant with the law of fruit biological decaying. In the next step, consultation for advice from professionals in this subject rather than randomly selecting people should be conducted.

YOLOv3, offered in this thesis, localizes objects through rectangular bounding boxes which introduces significant background noises. Our future work will take segmentation with polygon bounding regions into consideration that should reduce such disturbances.

References

Akinmusire, O. O. (2011). Fungal Species AssΔociated with the Spoilage of Some Edible Fruits in Maiduguri Northern Eastern Nigeria. *Advances in Environmental Biology, 5*(1), 157-161.

Arakeri, M., & Lakshmana, P. (2016). Computer Vision Based Fruit Grading System for Quality Evaluation of Tomato in Agriculture Industry. *Proceedings of the International Conference on Communication, Computing and Virtualization, Mumbai, India;, 79*(2016), 426-433.

Bampidis, V., & Robinson, P. (2006). Citrus By-products as Ruminant Feeds: A Review. *Animal Feed Science and Technology, 128*, 175–217.

Barret, D. M., Bealiue, J. C., & Shewflet, R. (2010). Color, Flavor, Texture, andNutritional Quality of Fresh-CutFruits and Vegetables: DesirableLevels, Instrumental and SensoryMeasurement, and the Effects ofProcessing. *Critical Reviews in Food Science and Nutrition, 50*, 369–389.

Barrett, D. M., Beaulieu, J. C., & Shewfelt, R. (2010). Color, Flavor, Texture, and Nutritional Quality of Fresh-Cut Fruits and Vegetables: Desirable Levels, Instrumental and Sensory Measurement, and the Effects of Processing. *Critical Reviews in Food Science and Nutrition,, 50*, 369-389.

Bhargava, A., & Bansal, A. (2018). Fruits and Vegetables Quality Evaluation Using Computer Vision: A Review. *Journal of King Saud University - Computer and Information Sciences*, 1-16.

Bottou, L., & Bousquet, O. (2012). The Tradeoffs of Large Scale Learning Cambridge. *Optimization for Machine Learning*, 351–368.

Bresilla, K., Perulli, G. D., Boini, A., Morandi, B., Grappadelli, L. C., & Manfrini, L. (2019). Single-Shot Convolution Neural Networks for Real-Time Fruit Detection Within the Tree. *Frontiers in Plant Science*, *10*, pp. 611-623. doi:10.3389/fpls.2019.00611

Brosnan, T., & Sun, D.-W. (2002). Inspection and Grading of Agricultural and Food Products by Computer Vision Systems: A Review. *Computers and Electronics in Agriculture, 36*(2), 193-213.

Cunha, J. B. (2003). Application of Image Processing Techniques in the Characterization of Plant Leafs. *IEEE International Symposium on Industrial Electronics, 1*, 612-616.

Davey, M. W., Stals, E., Ngoh-Newilah, G., Tomekpe, K., Lusty, C., & Markham, R. (2007). Sampling Strategies and Variability in Fruit Pulp Micronutrient Contents of West and Central African Bananas and Plantains (Musa Species). *Journal of Agricultural and Food Chemistry, 55*(7), 2633–2644.

Everingham, M., Gool, L. V., Williams, C. K., Winn, J., & Zisserman., A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision, 88*(2010), 303–338.

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional Single Shot Detector. *Computer Vision and Pattern Recognition*, 1-3.

Girshick, R. (2015). Fast R-CNN. *International Conference on Computer Vision*, 1440-1448.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Gomez-Sanchis, Gomez-Chova, L., Aleixos, N., Camps-Valls, G., Montesinos-Herrero, C., & Molto, E. (2008). Hyperspectral System for Early Detection of Rottenness Caused by Penicillium Digitatum in Mandarins. *Journal of Food Engineering, 89*(1), 80-86.

Haeberli, P., & Voorhies., D. (1994). Image Processing by Linear Interpolation and Extrapolation. *IRIS Universe Magazine*, 13-24.

Hahnloser, R., & Seung, H. S. (2006). Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks . *Neural Computation, 15*(3), 621-638.

Hahnloser, R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital Selection and Analogue Amplification Coexist in A Cortex-inspired Slicon Circuit. *Nature*, 947–951.

Hahnloser., R., & Seung, H. S. (2002). Selectively Grouping Neurons in Recurrent Networks of Lateral Inhibition. *Neural Computation, 14*(11), 2627-2646.

Hargava, A. B. (2018). Fruits and Vegetables Quality Evaluation Using Computer Vision: A Review. *A. Journal of King Saud University Computer and Information Sciences*, 1-67.

Hartman, J. (2010). Apple Fruit Diseases Appearing at Harvest. *Plant Pathology Fact Sheet*, 1-5.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

He, K., Zhang, X., Ren, S., & Sun., J. (2014). Spatial Pyramid Poolingin Deep Convolutional Networks for Visual Recognition. *European Conference on Computer Vision*, 346-361.

He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., & Li, X. (2017). Single Shot Text Detector with Regional Attention. *IEEE International Conference on Computer Vision (ICCV)*, 1-12.

Herbach, K., Stintzing, F., & Carle, R. (2006). Betalain Stability and Degradation-Structural and Chromatic Aspects. *Journal of Food Science, 71*, 41-50.

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *Annals of Statistics, 53*(1), 73–101.

Joseph, R., & Ali, F. (2016). YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.

Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library.* Newton: O'Reilly Media, ISBN 1491937998.

Ketkar, N. (2017). *Introduction to PyTorch.* Berkeley: Apress, ISBN 978-1-4842-2766-4.

Kleene, S. (1956). Representation of Events in Nerve Nets and Finite Automata. *Annals of Mathematics Studies*, 3–41.

Knee, M. (1972). Anthocyanin, Carotenoid, and Chlorophyll Changes in Peel of Cox's Orange Pippin Apples during Ripening on and off the Tree. *Journal of Experimental Botany, 23*, 184-196.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 84-90.

Lequeu, J., Fauconnier, M.-L., Chammai, A., Bronner, R., & Blee, E. (2003). Formation of Plant Cuticle: Evidence for The Occurrence of the peroxygenase Pathway. *Plant Journal, 36*, 155–164.

Lin, T.-Y., RoyChowdhury, A., & Maji, S. (2017). Bilinear CNNs for Fine-grained Visual Recognition. *International Conference of Computer Vision*, 1-14.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). SSD: Single Shot MultiBox Detector. *International Conference of Computer Vision*, 1-16. doi:DOI: 10.1007/978-3-319-46448-0_2

Mahaman, B., Maliappis, M., Passamc, H., Sideridis, A., Zorkadis, V., & Koumpouro, Y. (2004). Image Processing for Distance Diagnosis in Pest Management. *Computers and Electronics in Agriculture, 5*(5), 193-213.

McCulloch, W., & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 115–133.

Mditshwa, A., Magwaza, L., Tesfay, S., & Mbili, N. (2017). Postharvest Quality and Composition of Organically and Conventionally Produced Fruits: A Review. *Science of Horticulture*, 148–159.

Mitcham, B., Cantwell, M., & Kader, A. (1996). Methods for Determining Quality of Fresh Commodities. *Perishables Handling Newsletter, 85*, 1-5.

Moallem, P., Serajoddin, A., & Pourghassem, H. (2017). Computer Vision-Based Apple Grading for Golden Delicious Apples Based on Surface Features. 33-40.

Mureșan, H. &. (2018). Fruit Recognition from Images Using Deep Learning. *Acta Universitatis Sapientiae, Informatica., 1-*, 26-42. doi:10.2478/ausi-2018-0002

Nashat, A. A., & Hassan, N. M. (2018). Automatic Segmentation and Classification of Olive Fruits Batches Based on Discrete Wavelet Transform and Visual Perceptual Texture Features. *International Journal of Wavelets, Multiresolution and Information Processing, 16*(1), 185-193.

Ntsoane, M. L., Zude-Sasse, M., Mahajan, P., & Mahajan, D. (2019). Quality Assessment and Postharvest Technology of Mango: A Review of Its Current Status and Future Perspectives. *Science of Horticulture, 345*(30), 77-85.

Ozyildiz, E., Krahnst-over, N., & Sharma, R. (2002). Adaptive Texture Andcolor Segmentation for Tracking Moving Objects. *Pattern Recognization*, 2013-2029.

Pan, C., Yan, W. (2018) A Learning-Based Positive Feedback in Salient Object Detection. In the Proceedings of IVCNZ.

Pan, C., Yan, W. (2020) Salient Object Detection Based on Perception Saturation. Multimedia Tools and Applications. (DOI: 10.1007/s11042-020-08866-x)

Pandey, A., Alok, A., Lakhwani, D., Singh, J., Asif, M. H., & Trivedi, P. K. (2016). Genome-Wide Expression Analysis and Metabolite Profiling Elucidate Transcriptional Regulation of Flavonoid Biosynthesis and Modulation Under Abiotic Stresses in Banana. *Scientific Reports, 6*, 313-321.

Pandey, R., Naik, S., & Marfatia, R. (2013). Image Processing and Machine Learning for Automated Fruit Grading System: A Technical Review. *International Journal of Computer Applications, 81*(16), 29-39.

Péneau, S., Linke, A., Escher, F., & Nuessli, J. (2009). Freshness of Fruits and Vegetables: Consumer Language and Perception. *British Food Journal, 111*(3), 243-256.

Prakash, K. K. (2018). Spoilage Detection in Raspberry Fruit Based on Spectral Imaging Using Convolutional Neural Network. *Dissertation M.Sc. in Computing (Data Analytics)*. doi:10.21427/D7D23R

Rashmi, P., Sapan, N., & Roma, M. (2013). Image Processing and Machine Learning for Automated Fruit Grading System: A Technical Review. *International Journal of Computer Applications, 81*(16), 0975 – 8887.

Rawat, S. (2015). Food spoilage : Microorganisms and theirprevention. *Asian Journal of Plant Science and Research, 5*(4), 47-56.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition* , 779-788.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137 - 1149.

Rossum, G. v. (1995). *Python tutorial, Technical Report CS-R9526,*. Amsterdam: Centrum voor Wiskunde en Informatica (CWI), ISSN 0169-118X.

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 210–229.

Schwartz, S. J., & Von Elbe, J. (2006). Kinetics of Chlorophyll Degradation to Pyropheophytin in Vegetables. *Journal of Food Science, 48*(4), 1303-1306.

Shen, D., C. Xin, C., Nguyen, M., Yan W. (2018). Flame Detection Using Deep Learning. ICCAR'18

Shukla, A. K. (2017). *Electron Spin Resonance in Food Science* (1st ed.). Waltham : Academic Press, ISBN: 9780128133644.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, 1-14.

Sindhi, K., Pandya, J., & Vegad, S. (2016). Quality Evaluation of Apple Fruit: A Survey. *International Journal of Computer Applications, 136*, 32-36.

Singh, H., Gupta, M. M., Meitzler, T., Hou, Z.-G., Garg, K. K., Solo, A. M., & Zadeh, L. A. (2013). Real-Life Applications of Fuzzy Logic. *Advances in Fuzzy Systems*, 1-3.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2014). Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9. doi:10.1109/CVPR.2015.7298594

Tang, X., Du, D. K., He, Z., & Liu, J. (2018). PyramidBox: A Context-assisted Single ShotFace Detector. *European Conference on Computer Vision*, 1-17.

Tewari, V. K., Arudra, A. K., Kumar, S. P., Pandey, V., & Chande, N. S. (2013). Estimation of Plant Nitrogen Content Using Digital Image Processing. *International Commission of Agricultural and Biosystems Engineerin, 15*(2), 78-86.

Tournas, V. H., & Katsoudas, E. (2005). Mould and Yeast Florain Fresh Berries, Grapes and Citrus Fruits. *InternationalJournal of Food Microbiology, 105*(1), 11-17.

Tripathi, M. K., & Maktedar, D. D. (2019). A Role of Computer Vision in Fruits and Vegetables Among Various Horticulture Products of Agriculture Fields: A Survey. *Information Processing in Agriculture, 147*, 70-79.

Uchida, K., Tanaka, M., & Okutomi, M. (2018). Coupled Convolution Layer for Convolutional Neural Network. *Neural Networks, 105*, 197-205.

Uijlings, .., Sande, K. v., Gevers, T., & Smeulders, A. (2012). Selective Search for Object Recognition. *International Journal of Computer Vision*, 1-14.

Wang, T., Anwer, R. M., Cholakkal, H., Khan, F. S., Pang, Y., & Shao, L. (2019). Learning Rich Features at High-Speed for Single-Shot Object Detection. *International Conference of Computer Vision*, 1971-1980.

Warde-Farley, D., Goodfellow, I. J., Courville, A., & Bengio, Y. (2013). An Empirical Analysis of Dropout in Piecewise Linear Networks. *arXiv e-prints*, 1-7.

Wu, J. (2017). Introduction to Convolutional Neural Networks. *National Key Lab for Novel Software Technology, Nanjing University*, 1-27.

Yan, W., Kankanhalli M. (2002) Detection and Removal of Lighting & Shaking Artifacts in Home Videos. Proceedings of ACM International Conference on Multimedia, 107-116.

Yan, W. (2019) Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer, London.

Zeng, G. (2017). Fruit and Vegetables Classification System Using Image Saliency and Convolutional Neural Network. *Proceedings of the Technology and Mechatronics Engineering Conference*, 324-331. doi:10.1109/ITOEC.2017.8122370

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2017). Single-Shot Refinement Neural Network for Object Detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4203-4212.

Zheng, K.,Yan, W., Nand, P. (2018) Video Dynamics Detection Using Deep Neural Networks. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(3): 224-234

Zheng, Y., Yu, C., Cheng, Y., Zhang, R., Jenkins, B., & VanderGheynst, J. (2011). Effects of Ensilage on Storage and Enzymatic Degradability of Sugar Beet Pulp. *Bioresour Technol, 102*(2), 1489-1495.

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2017). Random Erasing Data Augmentation. *Computer Vision and Pattern Recognition*, 1-14.