İstanbul
Bilgi Üniversitesi

Identification of Fresh & Rotten Fruits and Vegetable Using Deep Learning techniques to reduce food waste

by

Obadah Nidal Ghizawi, 121200038
Hamza Sallam, 120200013
Abdulmoez S. Derrija, 120200136

Supervised by

Elena Battini Sönmez

Submitted to the
FACULTY OF ENGINEERING AND NATURAL SCIENCES
in partial fulfillment of the requirements for the

Bachelor of Science

in the
DEPARTMENT OF COMPUTER ENGINEERING

January 3, 2024

Abstract

To reduce food waste and maximize the nutritional value of fruits and vegetables, freshness levels must be considered. Existing approaches to this problem go one of two routes: either consider a binary class problem where known fruits/vegetables are classified into two classes: fresh or rotten, or three classes categorization: pure-fresh, medium-fresh, or rotten. There is a research paper that focuses on the three-class approach, implementing a VGG16 model to solve this problem. The aim is to challenge this paper by implementing the same multi-class approach. We will use the same dataset utilized by the aforementioned approach which includes 60,000 images of 11 fruits and vegetables divided into three classes of freshness. The classification and categorization of fruits and vegetables will be performed utilizing the VGG16 architecture. The benchmark paper achieved an accuracy of 82% with the VGG16 model and 84% with YOLO v5. After going through many experiments, ***we reached 94.58% test accuracy with VGG16, improving the accuracy of the benchmark paper by 11 percentage points.***

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
VGG16	Visual Geometry Group 16
YOLO	You Only Look Once
RGB	red, green, and blue
ReLU	Rectified linear unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
CE	Cross Entropy
AdaM	Adaptive Moment

1 Introduction

Nowadays, food waste stands as a formidable problem with far-reaching effects on social health and environmental sustainability. Many edible products are thrown away each year, due to consumer behavior and inefficiencies in the food supply chain. Approximately one-third of all food produced for human consumption (˜ 2.5 billion tons) is lost or wasted globally each year[6, 7]. As shown in Figure 1, fruits and vegetables are the top edible foods that are most likely to be thrown away with a percentage of 39%. To reduce such wastage, automated cutting-edge technologies become a must. Our project addresses this critical concern by aiming to implement an advanced deep-learning model capable of identifying fruits and vegetables and categorizing their freshness. This system can be utilized in retail establishments and smart fridges, providing consumers with real-time information about the freshness of the food. The system can also be integrated into distribution centers and warehouses, reducing the possibility that perishable goods would expire before being purchased by customers.

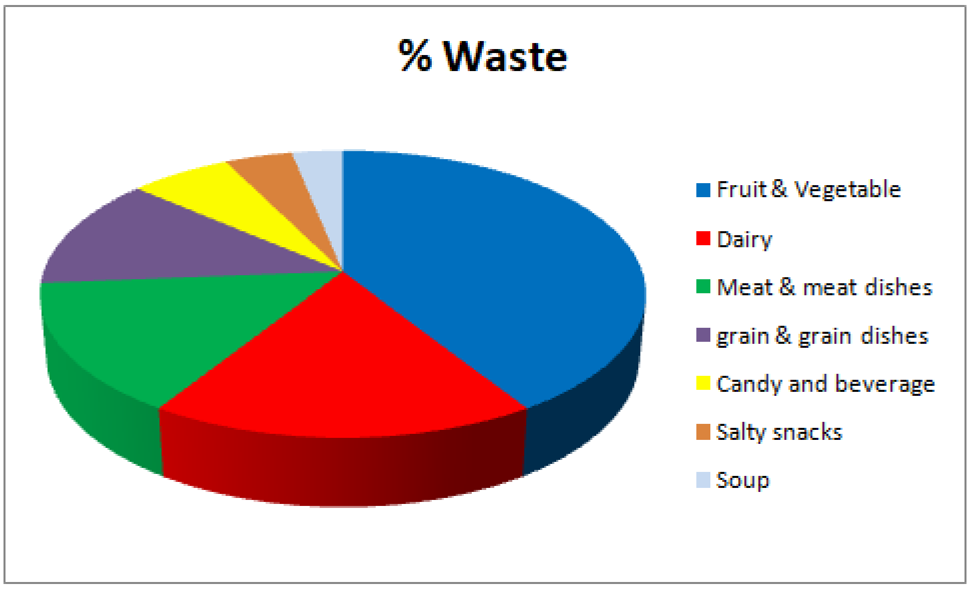


Figure 1: Percentage of food waste by category [1]

Although there are numerous methods to implement a classification model for the identification of fruits and vegetables, the number of classes observed plays an important role. Existing solutions categorize the freshness levels as fresh and rotten only, making it a binary classification problem. Another approach categorizes freshness into three levels fresh, medium, and rotten, introducing a multi-class problem for a more sophisticated and practical dimension to freshness assessment. Our project builds upon the work presented in the solution [8] proposed. The study utilizes two models: VGG16 and YOLOv5. VGG16 is primarily used for classification, and YOLOv5 is used for classification and localization of the object in the image. Our objective is to challenge its efficacy and accuracy. Through different experimentation and fine-tuning, we seek not merely to replicate but to exceed the accuracies delineated by the benchmark paper. Furthermore, we also aim to implement different architectures and different datasets to reach the maximal performance.

The rest of the work is organized as follows. Section 2, the Related Works, overviews the previous work and existing solutions done for the problem. Section 3, outlines the project design process. Section 4 summarizes the dataset used for this project. Section 5, the Methodology, contains the necessary information on neural networks, the model architectures, and our approach. Section 6 defines all metrics used to evaluate the performance. Section 7 details the experiments done. Section 8, shows our preliminary work's current results. Finally, Section 9, concludes this project and discusses future work.

2 Related Works

When examining the relevant publications and literature in the chosen field, it is apparent that most studies focus on a binary class classification problem with either fruits or vegetables. Most studies focus on either category, with some focusing on a specific fruit or vegetable. However, the study chosen as the benchmark paper focuses on both fruits and vegetables with a three-class classification: fresh, medium-fresh, and rotten. This is a lot more interesting as it is more descriptive of the actual state of the fruit/vegetable, and covers a more broad and applicable field of study. According to [8], they gather a dataset comprising 60,000 images of 11 different fruits and vegetables, divide them into three classes each (fresh, medium-fresh, and rotten), then use VGG16 and YOLOv5 architectures for image recognition and categorization. They achieved an accuracy of 82% using only VGG16, and 84% when combining VGG16 and YOLOv5 architectures. Within the VGG16 model, they use the Softmax activation function in the model's output layer due to the problem's multi-class nature (fresh, medium-fresh, and rotten). Additionally, they use the Categorical Cross Entropy Loss Function due to multi-class classification. Regarding YOLO, they use the Leaky ReLU activation function in hidden layers along with the sigmoid activation function in the final output layer. Furthermore, in [9], they utilized several CNN architectures and compared the results. The architectures examined are ResNet50, DenseNet, MobileNetV2, InceptionV3, and VGG16. The accuracies achieved were 90.8%, 87.3%, 88.5%, 87.3%, and 80.4% respectively. They proposed a binary classification of fruits with a relatively small dataset of 524 images.

In [10], they primarily utilized the AlexNet architecture on a dataset of 10,901 images of 3 types of fruits. The AlexNet architecture they used consisted of five convolutional layers and three fully-connected layers. Furthermore, they use transfer learning and fine-tune the CNN. They also use the softmax classifier for final classification. They classified the images in a binary classification. They were able to achieve an accuracy of 99%. Next, in [11], they implemented 5 different CNN architectures and compared the results. They proposed a binary classification of 5 types of fruits with a dataset of 5,658 images. The architectures tested were InceptionV3, Xception, VGG16, MobileNet, and NASNetMobile. The accuracies achieved were 97.34%, 97.16%, 96.54%, 95.47%, and 75.29% respectively.

In [12], the authors did not utilize a pre-defined architecture as most others did but utilized a regular Sequential CNN. They used three convolutional layers, with the filters increasing incrementally from 32 to 64 to 128 on each layer. They also used max-pooling layers with a kernel size of 2x2. Furthermore, they utilized dropout with a dropping rate of 0.25 to avoid overfitting the model. Additionally, they used the ReLU activation function for the fully connected/dense layer and the SoftMax activation function for the output layer. Their model classifies 3 different fruits into a binary class selection, meaning there are 6 total classes, with a dataset of 13,600 images from Kaggle. Lastly, in [13], utilize ResNet, VGG-11, and GoogLeNet architectures with the AlexNet architecture as the base network and YOLO for extracting the region of interest from digital images to classify the freshness grade for fruits. They examined 6 different fruits with a dataset of size 4,000. They have a unique proposition that aims to classify fruits and vegetables into a grade from 0.0 to 10.0, but this also comes with the problem of an extremely limited dataset. This is why they had to gather their dataset, which came with its limitations. On average, their mean squared error (MSE) average values of training and validation sets for the ResNet architecture are 3.582 and 4.058 respectively.

Overall, our model repeats the experiment of Fahad et al. [8] in the sense that we take a three-class approach to each fruit and vegetable, meaning we have 33 classes in total. This approach is unique as most models focus on either fruits or vegetables, but we want to create a more holistic and applicable model, therefore we challenged the 33-class issue proposed by Fahad et al. In the future, we may decide to increase the number of different fruits and vegetables. Further research and development may be necessary to finalize and fully realize the potential of our approach, however, our initial results are very promising and demonstrate the feasibility of our approach.

3 Project Design

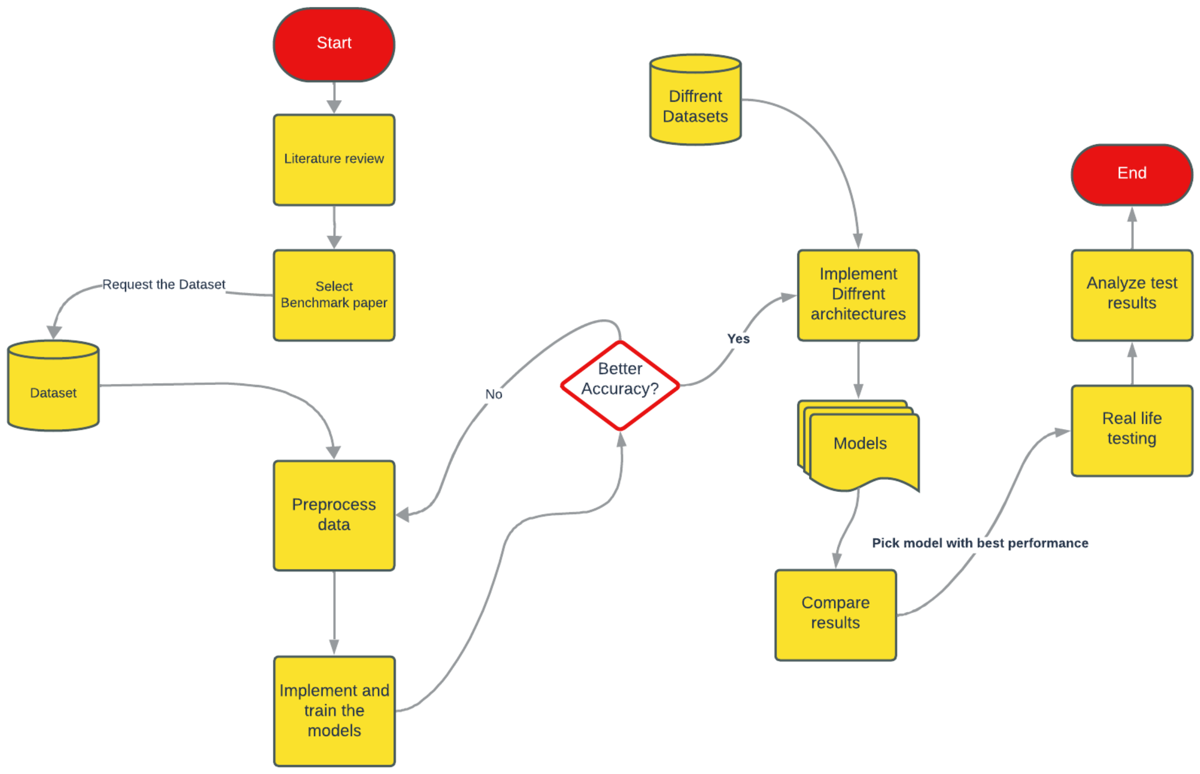


Figure 2: General project workflow plan

Figure 2 shows the general workflow of the project. We start with a literature review and select a benchmark paper that serves as a reference for existing methodologies. After preprocessing the data, we train the model proposed in the chosen benchmark paper[8]. If our result is better than theirs, the next step involves exploring and implementing different neural network architectures to further enhance the model's performance. Additionally, the project aims to diversify the data set used for training and testing, experimenting with different sets of fresh and rotten images to ensure robustness. Finally, we identify the most effective model that yields optimal performance in accurately classifying the condition of fruits and vegetables and test the model with real-life data.

4 Dataset

The dataset we chose to utilize was originally used by [8] in their work. It is publicly available for the research community and can be obtained upon request. It contains a total of 60,059 RGB images of 11 types of fruits and vegetables. six fruits (Apple, Banana, Guava, Orange, Lemon and Tomato) and five vegetables (Brinjal, Cucumber, Chili, Pepper, and Potato) The number of each fruit/vegetable in each category is shown in table 1.

The size of each image is 1800x1800. The images are diverse, containing differing backgrounds and either singular or multiple objects. One point to note is that different fruits and vegetables may appear similar because of their shape or color. For example, a lemon can be confused with an orange, the shape of a cucumber is similar to a green banana, etc. This makes it a challenging data set. The dataset has been split into a training and testing set with a ratio of 70-30, with 70% of the entire dataset for training, 20% of the training subset for validation, and 30% of the entire dataset for testing. The sizes of the subsets are 33,388, 8,332, and 18,336 for training, validation, and testing respectively.

Table 1: Count of each fruit and vegetable in 3 different categories

Name	**Fresh**	**Medium**	**Rotten**
Apple	4082	2975	2785
Banana	2530	899	1999
Orange	2335	1287	1988
Tomato	4261	4715	859
Guava	768	1696	773
Lemon	2281	2881	4033
Brinjal	999	1240	528
Chilies	1083	1392	805
Cucumber	775	857	682
Pepper	2478	927	980
Potato	2115	971	1080
Total	23,707	19,840	16,512

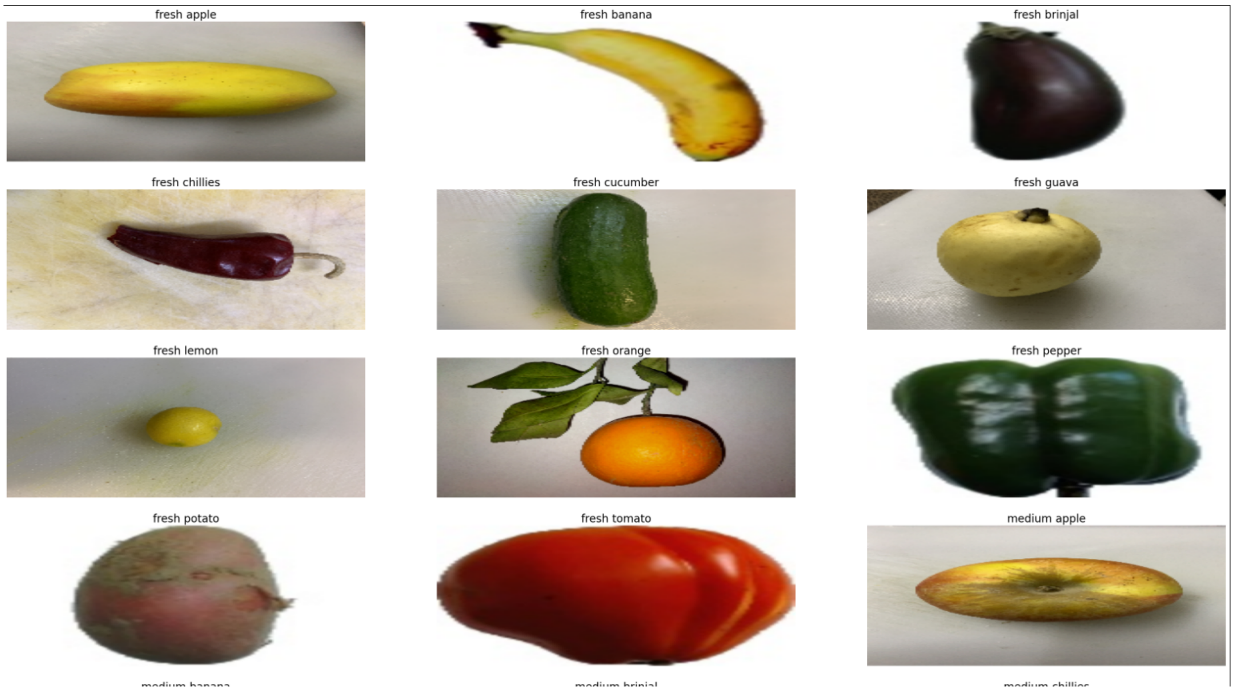


Figure 3: Some instances from the dataset

5 Methodology

5.1 Neural networks

Neural networks, also known as artificial neural networks (ANNs) are a subset of machine learning and are at the heart of deep learning algorithms [14]. The name and structure of neural networks are inspired by the human brain and closely mimic the way that biological neurons signal to one another. ANNs contain an input layer, one or more hidden layers, and an output layer, as shown in figure 4.

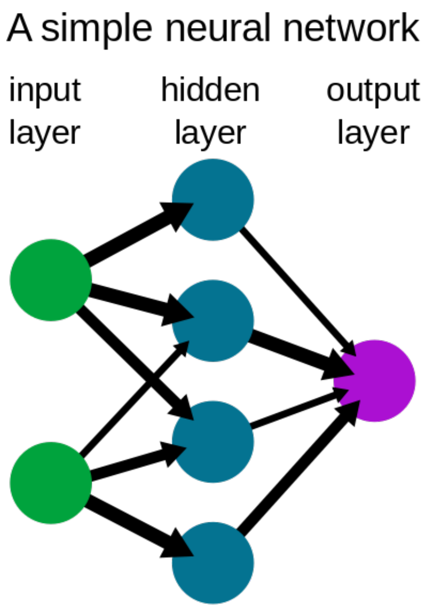


Figure 4: Simple neural network [2]

Each node within an ANN connects to another and has an associated weight and threshold. If the output of any individual node exceeds a specified threshold value, the node is activated and the data contained within the node is transferred to the next layer of the network. Otherwise, if the threshold value is not exceeded, no data is transferred to the next layer of the network [14].

5.2 Convolutional Neural Networks (CNN)

A Convolutional Neural Network (CNN) is a specialized kind of neural network that is superior at distinguishing unique features in images, speech, or audio signal inputs [15]. CNNs have three main types of layers: convolutional layers, pooling layers, and fully connected (FC) layers. Fully connected layers are also called dense layers.

The convolutional layer is the first layer in a convolutional network. The convolutional layer can be followed by additional convolutional layers or pooling layers, however, the final layer will always be the fully connected/dense layer. With each additional layer, the CNN increases in complexity. Generally, the first few layers in a CNN focus on extracting simple features such as color and edges, and as the image data progresses through the layers in a CNN, larger elements or shapes of the object start to be recognized which leads to the identification of the intended object [15].

5.2.1 Convolutional layer

The convolutional layer is a fundamental building block of a CNN, where the majority of the computation occurs [15]. The convolutional layer only requires the input data, a filter, and a feature map. For example, we have a colored image represented as a matrix of pixels in 3D. This means that we will have three dimensions: width, height, and depth of the image which correspond to its RGB values. Additionally, we have the kernel, or filter, which is moved across the receptive fields of the image, checking if the desired feature is present. This process is known as convolution [15]. The result of the convolution is stored in the feature map. This is shown in diagram 5.

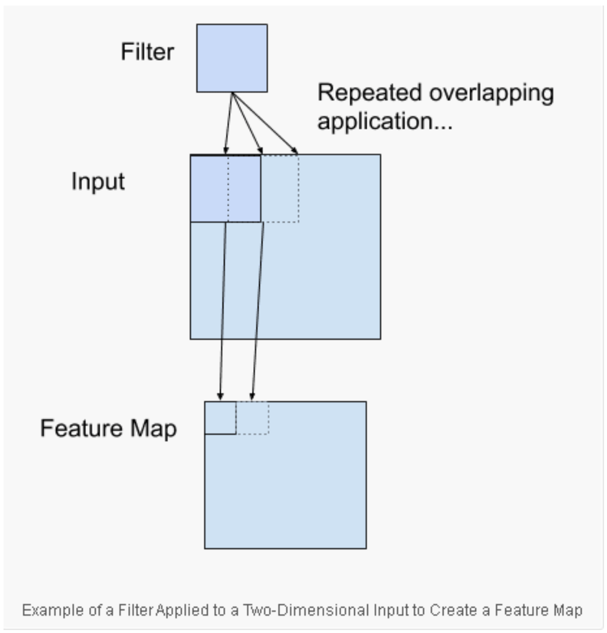

Filter
Repeated overlapping application...
Input
Feature Map
Example of a Filter Applied to a Two-Dimensional Input to Create a Feature Map

Figure 5: Example of a Filter Applied to a Two-Dimensional Input to Create a Feature Map [3]

5.2.2 Pooling layer

The pooling layer downsamples the number of parameters in the input and extracts crucial information. Pooling results in a lot of information loss, however, there are several benefits to utilizing it. Pooling layers help CNNs reduce complexity, improve efficiency, and limit the risk of overfitting [15].

$$Y_{ij} = \max_{m,n}(X_{p\cdot i+m,p\cdot j+n}) \tag{1}$$

Equation 1 portrays a max-pooling layer. Y represents the output of the pooling layer at position (i,j), and max denotes the maximum operation over the elements of the specified region. X represents the input feature map at position $(p \cdot i + m, p \cdot j + n)$ in the original input.

5.2.3 Dense Layer

The dense layer, or fully connected layer, in a neural network, connects all nodes in the output layer directly to a node in the previous layer. Each node contains several inputs, weights, a bias term, and an activation function [15].

$$Y = \mathrm{a}(X \cdot W + b) \tag{2}$$

Equation 2 displays the dense layer output formula, Where 'a' represents the activation function used in the layer, X is the input vector, W is the weight matrix, b is the bias, and $\cdot$ denotes the dot product.

5.2.4 Regularization

Regularization is a set of methods that serve primarily as a means to reduce overfitting in machine learning models. This is often done with a tradeoff of a marginal decrease in training accuracy for an increase in model generalizability [4].
Dropout is a special regularization technique that is used within this study. Dropout regularizes neural networks by randomly dropping out nodes, along with any input and output connections, from the network during training. This is shown in 6. The nodes dropped during a certain epoch are randomly chosen. Dropout results in an enhancement in the robustness and generalizability of a model, and ultimately reduces overfitting.

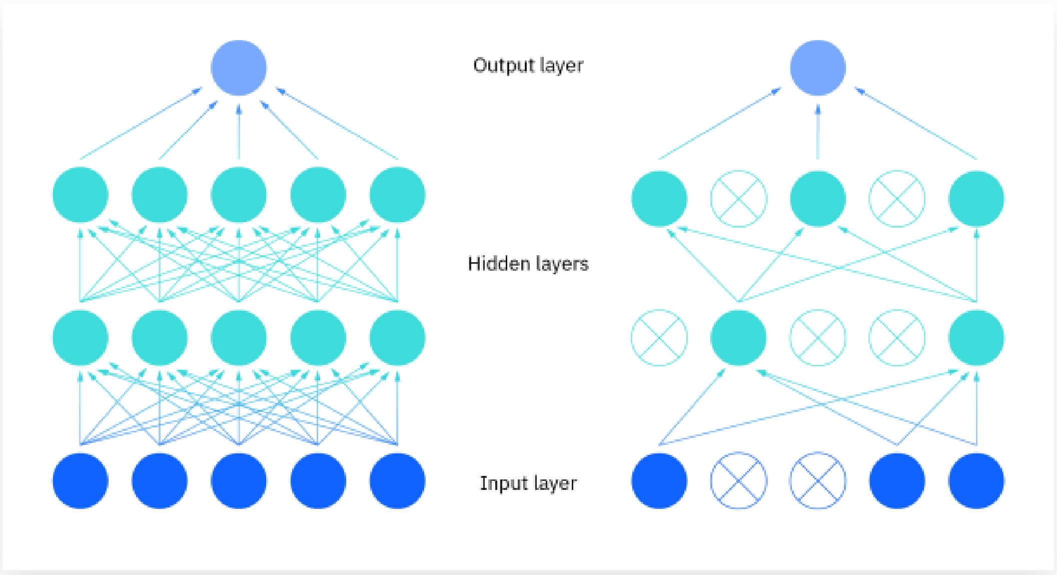


Figure 6: Diagram of Dropout Within a Neural Network [4]

5.2.5 Activation functions

Activation functions are non-linear which allows for the approximation of complex functions and also provides non-linearities into a network [16]. Since real-life data is often non-linear, and models generally focus on real-life applications, activation functions are crucial for the success of any feasible neural network.

$$\text{ReLU}(x) = \max(0, x) \tag{3}$$

Equation 3 shows the Rectified Linear Unit (ReLU) activation function. The equation states that the ReLU of x is the maximum of 0 and x. If x is positive, the ReLU function returns x; otherwise, it returns 0. This is a widely used function in real-time implementations. Its popularity generally stems from the computational simplicity of the function, linear behavior, and the capability of outputting a true zero value (representational sparsity) [17]. Due to the aforementioned reasons, we will be utilizing the ReLU activation functions in most of our fully connected/dense layers.

$$s(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{4}$$

Equation 4 shows the softmax activation function, where z is the output vector, and C is the number of classes. The softmax activation function transforms raw outputs from a neural network into a vector of probabilities, essentially a probability distribution over the input classes. In a multi-class classification problem, softmax is essential in the output layer of a neural network since it makes raw data from the neural network readily interpretable [18]. Due to the nature of our model, we will also be using the softmax activation function in the output layer.

5.3 Loss function

The loss function, also known as the cost function, measures the difference, or error, between actual and predicted values at a certain position [19]. This improves a machine learning model's efficacy by providing feedback to the model so it can adjust the weights and biases throughout the network to minimize the error and find a local or global minimum. This process continuously iterates, moving along the direction of the steepest descent until the cost function is close to or at zero, at which point the model will stop learning [19].

$$L_{CE} = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i) \tag{5}$$

Equation 5 shows the categorical cross entropy formula, where y_i is the true labels ranging from 0 to 1,$\hat{y}_i$ is the model prediction and C is the number of classes. Categorical Cross Entropy, also known as ”softmax loss”, is a softmax activation plus a Cross-Entropy loss used for multiclass classification [20]. Since this loss function specializes in multiclass classification, we will be using it in our model.

5.4 VGG16

Visual Geometry Group 16 (VGG16) is a convolutional neural network architecture that was introduced in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) event in 2014 by Simonyan et al, a group in the Department of of Engineering Science, University of Oxford. They showcased their model in paper [21], which won 1st and 2nd place for object detection and classification. It was able to classify more than a million images of 1000 different classes with 92.7% accuracy[21]. VGG16 is considered one of the best computer vision models [22].

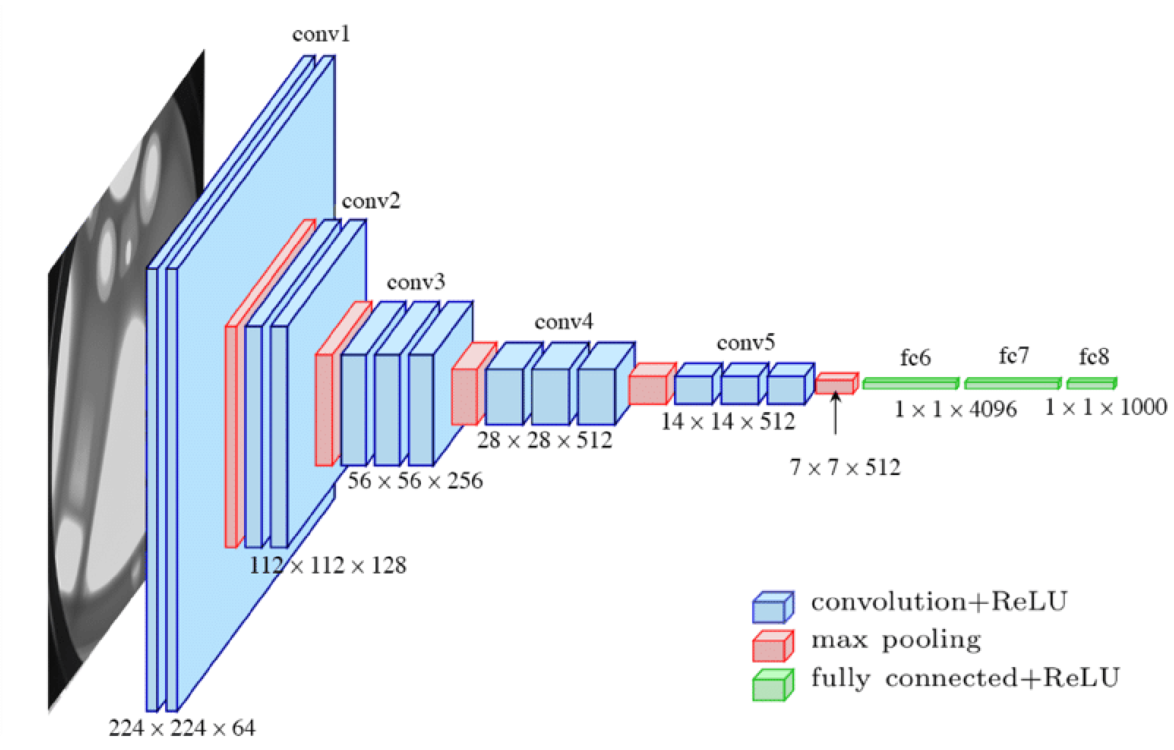


Figure 7: VGG16 architecture [5]

As shown in figure 7, VGG16 consists of thirteen convolutional layers, five max-pooling layers, and three dense layers which sum up to 21 layers. VGG16 has sixteen learnable layers, which explains the 16 in the name. The details of the architecture are as follows:

- VGG16 takes input image of size 224x224 with 3 RGB channels.
- Instead of using a large number of hyper-parameters, VGG16 uses convolutional layers with 3x3 filters and a stride of 1.
- Max pooling layer with 2x2 filter with the same padding and stride 1
- Pooling is crucial given the rapid growth of the available number of filters from convolutional layers 1-5. The first convolutional layer has 64 filters, then rapidly grows to 128, 256, and eventually 512 in the final convolutional layers.
- There are three fully connected layers. The first two fully connected layers have 4096 channels, and the output layer has 1000 channels, one for every class.

5.5 Our approach

A predefined VGG16 architecture exists from the Keras library that already implements all of the layers until the last max pooling layer. We merged the predefined architecture and froze its learnable layers. We then added two dense layers with ReLU [3] as the activation function and Softmax [4] for the output layer. A dropout of a rate of 20% was used for regularization. The loss function used for calculating error is the categorical cross-entropy function shown in formula 5 because the model has multiple classes as an output. AdaM optimization was used as our gradient descent method with a learning rate of 0.0001. The model was trained for 15 epochs.

6 Performance Metrics

To evaluate the performance of the model, two fundamental metrics are used, the confusion matrix and the classification report. The confusion matrix provides a comprehensive summary of a model's predictions and the classification report provides a more detailed overview of the model's performance. These metrics help us check the model's ability to identify both positive and negative cases and assess the presence of misclassification cases. the formulas for each metric are shown below.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{F1} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

$$\text{Loss} = \frac{FP + FN}{TP + TN + FP + FN} \tag{10}$$

- True positive (TP): correctly predicted positive instance
- False positive (FP): incorrectly predicted positive instance
- True negative (TN): correctly predicted negative instance
- False negative (FN): incorrectly predicted negative instance

The formula shown in 6 calculates the accuracy by dividing true predictions by all predictions. In contrast, equation 10 calculates the error or the misclassification by dividing the incorrect predictions by all predictions. Equation 7 shows the precision, which is the capability to detect ONLY present instances, while the Recall shown in 8 detects ALL present instances. Equation 9 calculates the F1 score, which is what percent of positive predictions were correct. In addition to these metrics, Support was also calculated, which is the number of actual occurrences of the class in the data set. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the model and could indicate the need for rebalancing. Support doesn't change between models but instead diagnoses the evaluation process[23].

7 Experiments

Through four different experiments as shown in table 2, each varying in terms of epochs, the number of hidden layers, and the number of neurons in each layer. Experiment 1, with 15 epochs and 2 hidden layers (32 and 16 neurons), showed moderate performance with a training accuracy of 75.84% and a test accuracy of 91.28%. Experiment 2 increased the complexity with 64 and 32 neurons in its layers, resulting in a significant improvement in both training and test accuracy, along with a high F1 Macro score of 0.9102.

In Experiment 3 we increased the model's capacity with 128 and 64 neurons, reaching the highest training accuracy of 98.32% and an impressive test accuracy of 94.58%. The F1 Macro score also peaked at 0.9207 indicating a balanced performance across different classes. Finally Experiment 4, the most complex model with three layers of 128, 64, and 32 neurons and 20 epochs, slightly trailed behind Experiment 3 in terms of training accuracy and test accuracy, with also a lower F1 Macro score.

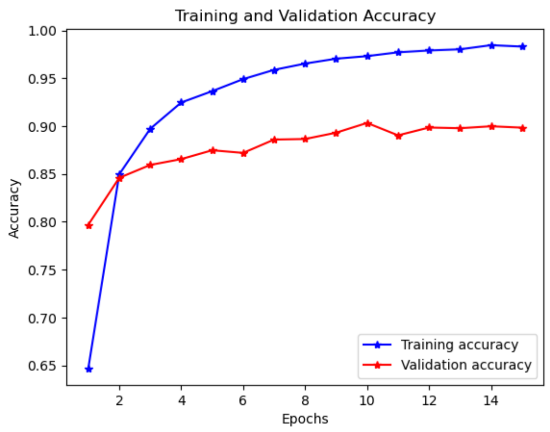
Based on these results, experiment 3 appears to be the most effective, achieving the highest test accuracy and F1 Macro score, which are important indicators of a model's ability to generalize well on new data. The increase in neurons and the maintenance of the number of epochs in Experiment 3 seems to offer the best balance between complexity and performance, which makes it the standout choice among the conducted experiments.

Experiment	Parameters			Metrics			
	epochs	No. of Dense Layers	No. of Neurons in the dense layers	Training Acc	F1 Macro	Test Acc	Loss
1	15	2	32,16	0.7584	0.8539	0.9128	0.3426
2	15	2	64,32	0.9169	0.9102	0.9296	0.3870
3	15	2	128,64	0.9832	0.9207	0.9458	0.2140
4	20	3	128,64,32	0.9675	0.9148	0.9390	0.2493

Table 2: Experimental Results

8 Current Results

After picking the best performance model from the experiments shown in section 7, As shown in Figure 8, we achieved a training accuracy of 98.32%, validation accuracy of 89.85%, and a loss of 0.05% and validation loss of 0.36% with no sign of overfitting. For evaluation, the confusion matrix and the classification report were computed as shown in figure 9 and table 2. We achieved an average precision of 0.93, an average recall of 0.92, an average F1 score of 0.92, and an overall accuracy of 94.58%.



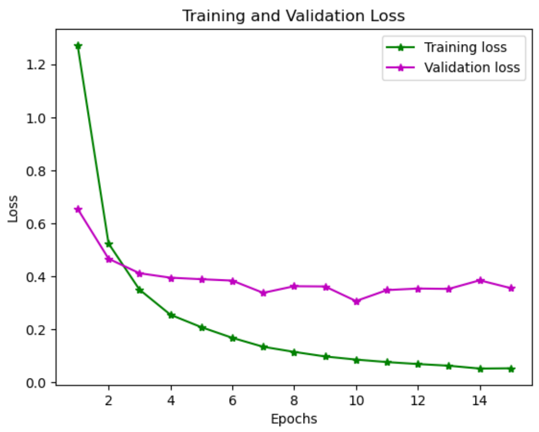


Figure 8: Training Accuracy and loss

The confusion matrix is shown in figures 9. We can see that the best performance model (experiment 3 shown in table 2) generally performs well on certain classes, such as fresh banana, fresh lemon, medium tomato, and rotten potato, where the number of correct predictions is high (noted by the high numbers along the diagonal). However, there are a total of 994 misclassifications with a loss of 0.214%; for instance, fresh apples are often misclassified as medium apples and fresh tomatoes as medium tomatoes. These errors could be due to similarities in features between fresh and medium states of the same item which can also be confusing for humans; For example, what one person sees as a fresh apple might be seen as a medium fresh apple by another, depending on personal standards or experiences. Factors such as color, texture changes, and minor blemishes can make the classification subjective.

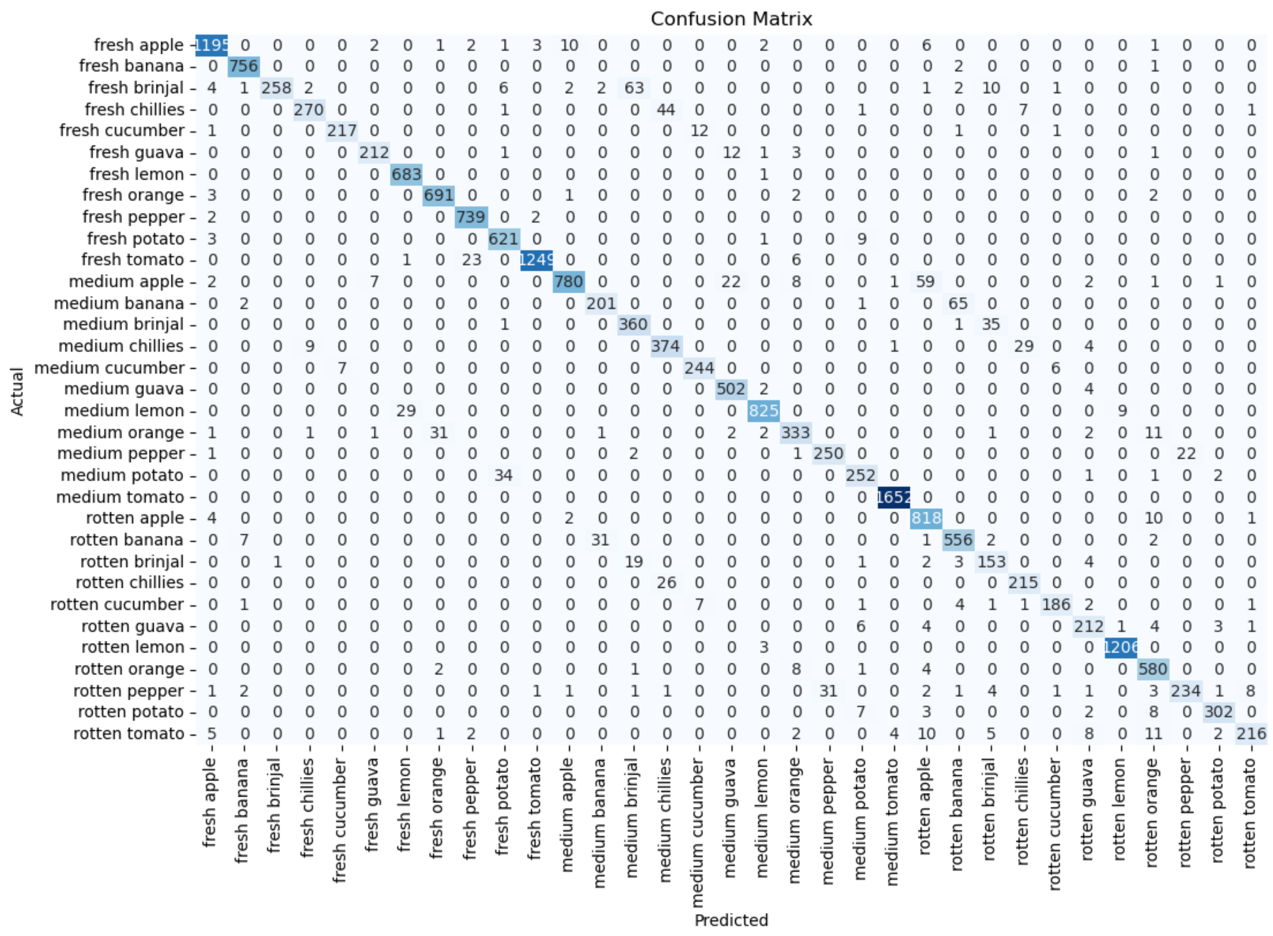


Figure 9: Confusion matrix for VGG16

Class	**Precision**	**Recall**	**F1-Score**	**Support**
fresh apple	0.9779	0.9771	0.9775	1223
fresh banana	0.9831	0.9960	0.9895	759
fresh brinjal	0.9961	0.7330	0.8445	352
fresh chillies	0.9574	0.8333	0.8911	324
fresh cucumber	0.9688	0.9353	0.9518	232
fresh guava	0.9550	0.9217	0.9381	230
fresh lemon	0.9579	0.9985	0.9778	684
fresh orange	0.9518	0.9886	0.9698	699
fresh pepper	0.9648	0.9946	0.9795	743
fresh potato	0.9338	0.9795	0.9561	634
fresh tomato	0.9952	0.9765	0.9858	1279
medium apple	0.9799	0.8834	0.9291	883
medium banana	0.8553	0.7472	0.7976	269
medium brinjal	0.8072	0.9068	0.8541	397
medium chillies	0.8404	0.8969	0.8677	417
medium cucumber	0.9278	0.9494	0.9385	257
medium guava	0.9331	0.9882	0.9598	508
medium lemon	0.9857	0.9560	0.9706	863
medium orange	0.9174	0.8627	0.8892	386
medium pepper	0.8897	0.9058	0.8977	276
medium potato	0.9032	0.8690	0.8858	290
medium tomato	0.9964	1.0000	0.9982	1652
rotten apple	0.8989	0.9796	0.9375	835
rotten banana	0.8756	0.9282	0.9011	599
rotten brinjal	0.7251	0.8361	0.7766	183
rotten chillies	0.8532	0.8921	0.8722	241
rotten cucumber	0.9538	0.9118	0.9323	204
rotten guava	0.8760	0.9177	0.8964	231
rotten lemon	0.9918	0.9975	0.9946	1209
rotten orange	0.9119	0.9732	0.9416	596
rotten pepper	0.9141	0.7986	0.8525	293
rotten potato	0.9711	0.9379	0.9542	322
rotten tomato	0.9474	0.8120	0.8745	266
Accuracy			0.9458	18336
Macro Avg	0.9272	0.9177	0.9207	18336
Weighted Avg	0.9474	0.9458	0.9454	18336

Table 3: Classification Report of VGG16

9 Conclusion & Future work

The primary objective of this project is to decrease food waste, a pressing concern in our daily lives. Inspired by the work of [8], we utilized VGG16 and several deep-learning techniques and were able to accomplish our goal of beating [8] in both efficacy and accuracy. We achieved an accuracy of 94.58%, which beats [8] by approximately 12.58%.

Since we accomplished our goal within the first few months of launching this project, there will be a lot to do within the next few months. Future work will involve the integration of YOLOv5 into our model, experimenting with different architectures, and further refining our methodology to maximize model accuracy. Furthermore, we plan to diversify and revamp our dataset by potentially looking for new fruits, vegetables, and potentially even foods to incorporate into our model. This project sets the stage for continued advancements in leveraging deep learning to address food waste.

References

[1] "The expert center, the-food-waste-problem-and-how-to-reduce-it," 2022. https://ask-bioexpert.com/blog-post/the-food-waste-problem-and-how-to-reduce-it/.

[2] Wikipedia, "Neural network," 2023. Accessed: December 26, 2023.

[3] J. Brownle, "How do convolutional layers work in deep learning neural networks?," 2020. https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/ , Accessed: December 31,2023.

[4] "What is regularization? — IBM — ibm.com." `https://www.ibm.com/topics/regularization`. [Accessed 01-01-2024].

[5] K. Le, "An overview of vgg16 and nin models," 2021. https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484.

[6] 'school of public health. Harvard, "Food waste," 2017. https://www.hsph.harvard.edu/nutritionsource/sustainability/food-waste/.

[7] S. Stephanie, Greenly, "Global food waste in 2023," 2022. https://greenly.earth/en-us/blog/ecology-news/global-food-waste-in-2022.

[8] L. G. Fahad, S. F. Tahir, U. Rasheed, H. Saqib, M. Hassan, and H. Alquhayz, "Fruits and vegetables freshness categorization using deep learning.," *Computers, Materials & Continua*, vol. 71, no. 3, 2022.

[9] T. Göksu, Z. Kaya, and S. Sahmoud, "Classification of fruit images as fresh and rotten using convolutional neural networks," in *2023 3rd International Conference on Computing and Information Technology (ICCIT)*, pp. 297–301, IEEE, 2023.

[10] U. Amin, M. I. Shahzad, A. Shahzad, M. Shahzad, U. Khan, and Z. Mahmood, "Automatic fruits freshness classification using cnn and transfer learning," *Applied Sciences*, vol. 13, no. 14, p. 8087, 2023.

[11] M. S. Miah, T. Tasnuva, M. Islam, M. Keya, M. R. Rahman, and S. A. Hossain, "An advanced method of identification fresh and rotten fruits

using different convolutional neural networks," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–7, IEEE, 2021.

[12] M. Kulkarni, A. Chaudhari, S. Shegokar, K. Rudrawar, S. Joshi, S. Tibile, and I. Tayade, "Fruit freshness detection using cnn," in *2022 International Conference on Futuristic Technologies (INCOFT)*, pp. 1–5, IEEE, 2022.

[13] Y. Fu, M. Nguyen, and W. Q. Yan, "Grading methods for fruit freshness based on deep learning," *SN Computer Science*, vol. 3, no. 4, p. 264, 2022.

[14] "What are Neural Networks? — IBM — ibm.com." `https://www.ibm.com/topics/neural-networks`. [Accessed 01-01-2024].

[15] "What are Convolutional Neural Networks? — IBM — ibm.com." `https://www.ibm.com/topics/convolutional-neural-networks`. [Accessed 01-01-2024].

[16] https://community.ibm.com/community/user/ai-datascience/people/pavan-saish naru, "Understanding Activations &amp; Optimization- Neural Networks — community.ibm.com." `https://community.ibm.com/community/user/ai-datascience/blogs/pavan-saish-naru/2023/01/27/optimization-hyperparameters`. [Accessed 01-01-2024].

[17] "An Introduction to the ReLU Activation Function — builtin.com." `https://builtin.com/machine-learning/relu-activation-function`. [Accessed 01-01-2024].

[18] B. P. C, "Softmax Activation Function: Everything You Need to Know — Pinecone — pinecone.io." `https://www.pinecone.io/learn/softmax-activation/`. [Accessed 02-01-2024].

[19] "What is Gradient Descent? — IBM — ibm.com." `https://www.ibm.com/topics/gradient-descent`. [Accessed 02-01-2024].

[20] "Cross Entropy Loss: Intro, Applications, Code — v7labs.com." `https://www.v7labs.com/blog/cross-entropy-loss-guide`. [Accessed 02-01-2024].

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] Great.learning, “Everything you need to know about vgg16,” 2021. https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918, , Accessed: December 28,2023.

[23] S. Kohli, “Understanding a classification report for your machine learning model,” 2019. https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397, Accessed: December 25,2023.