Artificial Intelligence

Final Project Report -Under the guidance of Professor -Guangchi Liu

Team Members: -

SuryaPraveen Adivi

Bahuroop Chakravarthy paladugu

Roshini Erukulla

Introduction: -

There has been a growing trend in recent years to use technology to simplify and automate various aspects of our lives. This has been especially evident in the grocery purchasing industry. With the proliferation of online grocery shopping and home delivery services, consumers have more options than ever before for purchasing supplies.

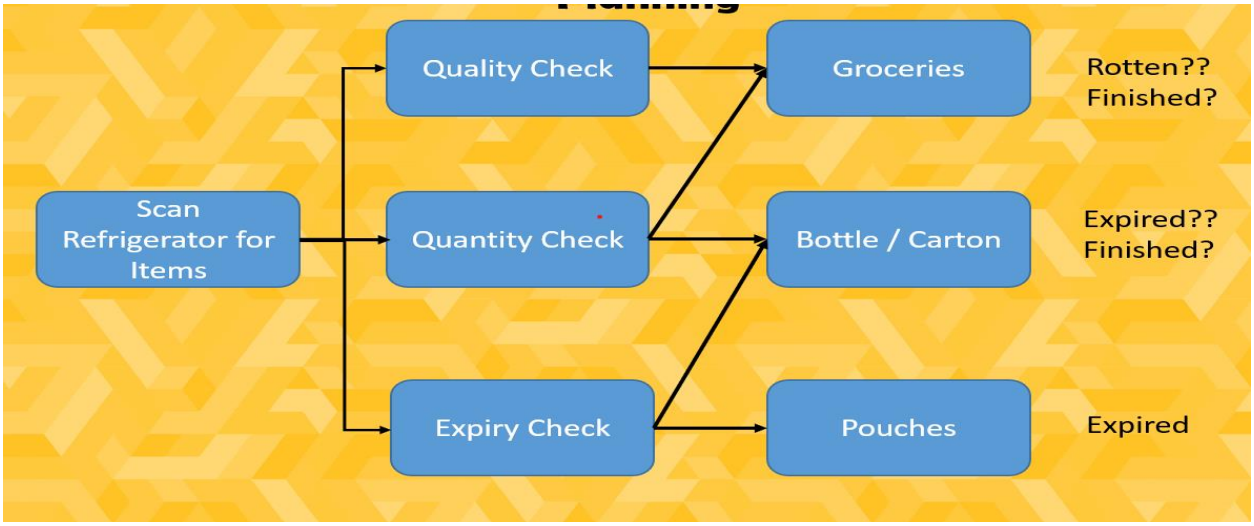
Even with these new technologies, constructing a shopping list remains a challenge for many people when grocery shopping. Creating an exhaustive and accurate grocery list can be time-consuming and tedious, despite its apparent simplicity. This is where the Convolutional Neural Network-based Smart Grocery List Generator comes in.

Methodology: -

The Smart Grocery List Generator is constructed with convolutional neural networks (CNNs), a type of artificial neural network that is frequently used for image recognition tasks. CNN is trained using images of various grocery items, including fruits, vegetables, and packaged foods. The system checks the quality, quantity and expiry of the groceries and generates a list of groceries to be purchased. The dataset we used consists of a lot of image data of about **4Giga bytes**. For classification of bananas itself we trained and tested our model with about **5000 images**.

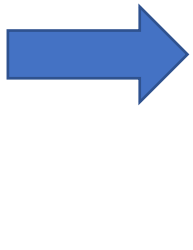
In the code we have Normalized X training photos by 255 divides all pixel values. Scaling pixel values between 0 and 1 is a typical preprocessing technique. Kera's Sequential API defines model architecture. The first layer is a 2D convolutional layer with 32 filters, 3x3 filters, and X.shape[1:]. ReLU activates. Next, a 2x2 2D max pooling layer is applied. Another 2D convolutional layer with 64 3x3 filters follows. Another ReLU activation function, max pooling layer, and dropout layer with 0.25 dropout rate are added.2D convolutional layer with 128 3x3 filters follows. ReLU activation, max pooling, and a 0.25-dropout layer are added again. The model's last layers are a flattened layer to transform the previous layer's output into a 1D vector, a dense layer with 128 units and a dropout rate of 0.25, and a final dense layer with 1 unit and a sigmoid activation function. Training uses binary cross-entropy and Adam as the optimizer. The model is compiled using the loss function, optimizer, and accuracy metric. The

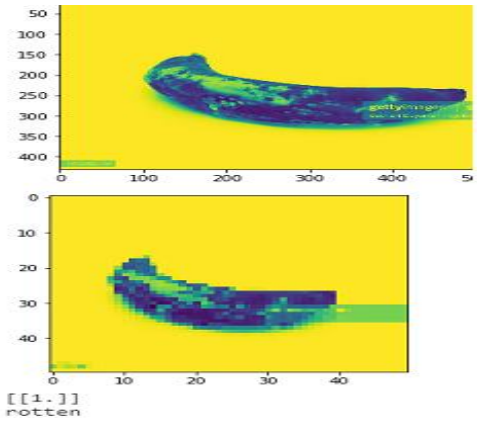
model is then trained on X and y using 32 batches and 5 epochs. The validation split is 0.3, therefore 30% of training data will be validated. This code trains a convolutional neural network model to categorize fruit photos using the dataset.



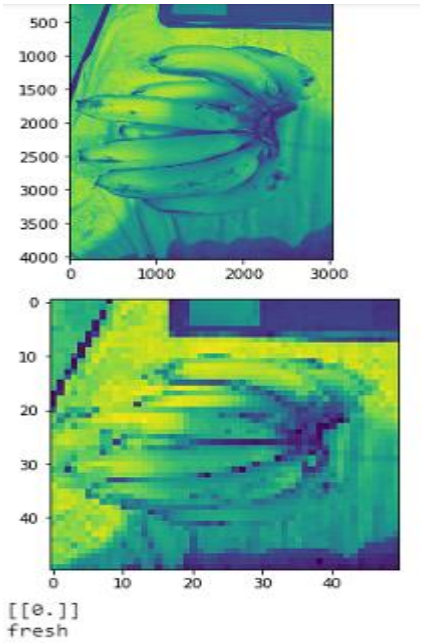
Results: -

Our Smart Grocery List Generator successfully detected Quality, Quantity and expiry of the products and was simple to use and provided helpful reminders for items. Future work on the system will concentrate on expanding the dataset to include a broader range of retail items and enhancing CNN's accuracy. The system will also be integrated with existing grocery delivery services to provide users with a seamless and convenient purchasing experience.

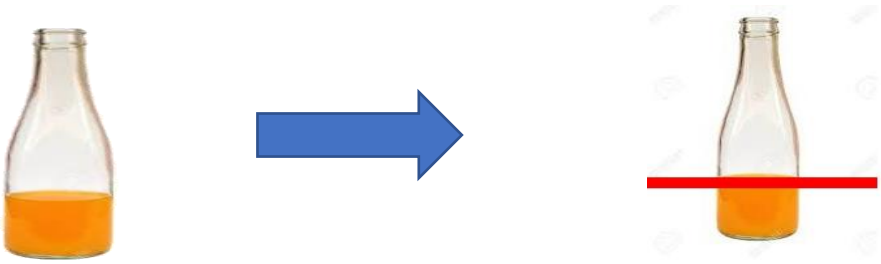




Rotten banana



Fresh banana

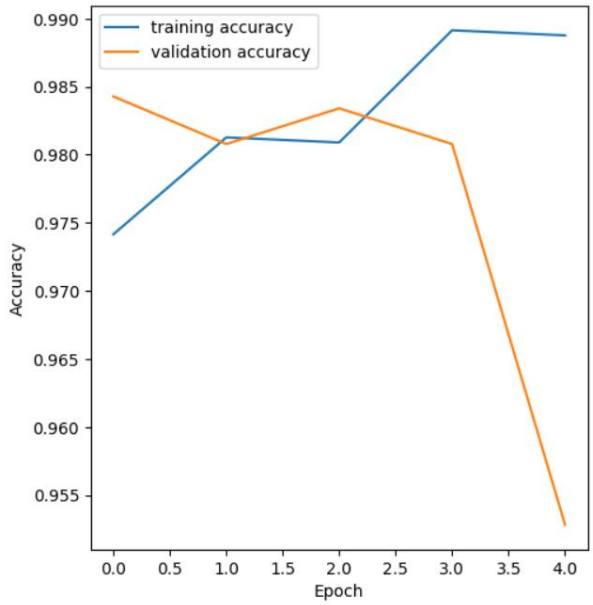


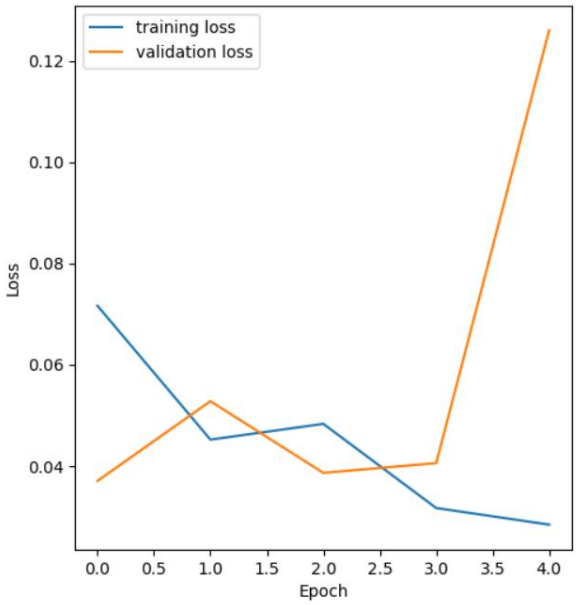
height filled -80

Std_height-220.0

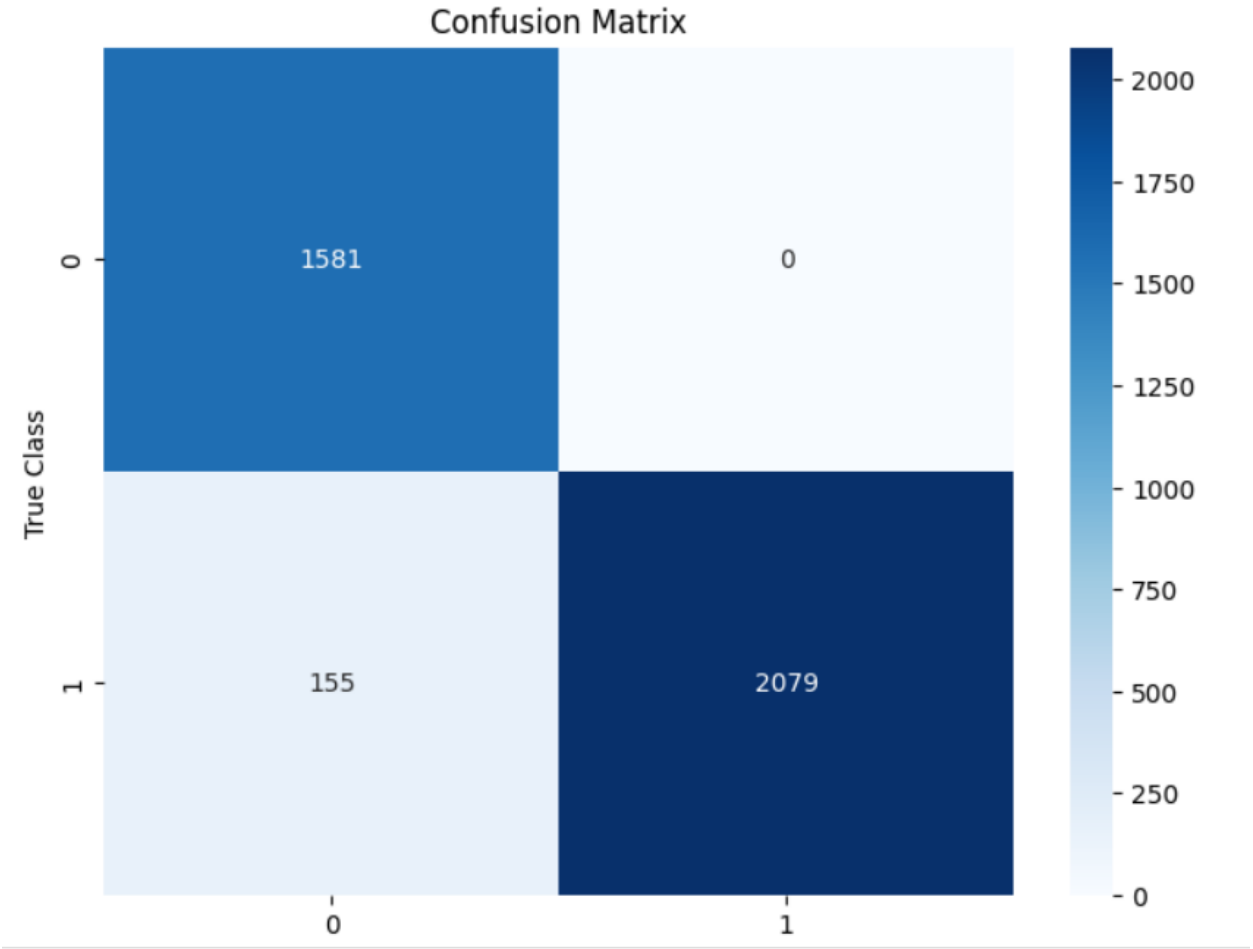
Amount filled – 36%

	Actual Fruit	Predicted Fruit
0	Rotten Banana	Rotten Banana
1	Fresh Banana	Rotten Banana
10	Rotten Banana	Rotten Banana
100	Fresh Banana	Rotten Banana
101	Fresh Banana	Rotten Banana
102	Rotten Banana	Rotten Banana
103	Rotten Banana	Rotten Banana
104	Rotten Banana	Rotten Banana
105	Fresh Banana	Rotten Banana
106	Fresh Banana	Rotten Banana


training accuracy
validation accuracy
Accuracy
Epoch
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0.955
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


training loss
validation loss
Loss
Epoch
0.12
0.10
0.08
0.06
0.04
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Accuracy: 95.94%



Challenges Faced: -

Convolutional neural networks (CNNs) provide several difficulties that must be overcome while developing a smart grocery list generator. Data Availability: In order to train the CNNs, a sizable dataset of tagged photos of food and drink products is required. It can be costly and time-consuming to gather this information. Users' input data may have varying levels of quality, requiring manual corrections such as spelling mistakes, abbreviations, or missing information before it can be used. User dietary preferences and restrictions: These factors can increase the complexity of the recommendation engine and must be considered by the system. The system should be easily linked with preexisting grocery delivery services to give customers the best possible shopping experience. Machine learning techniques, such as convolutional neural networks (CNNs), can be used to model the task of generating a Smart Grocery List by analyzing photos of grocery goods and providing recommendations based on the user's input and

shopping history. One alternative is to examine the user's input text with natural language processing methods and then provide a set of suggestions based on that analysis.

Advantages and Disadvantages: -

The advantage of the CNN-based strategy is its high accuracy in detecting and categorizing photos of grocery products; however, training and running such an approach demands a sizable dataset and processing resources. While the natural language processing method may not be as accurate in recognizing specific grocery products, it is more versatile and can handle input material that is less structured. Convolutional Neural Networks for picture identification and recommendation engines for creating personalized lists based on user input and history are two examples of algorithms well-suited to dealing with the models. The implementation details and hardware resources will determine the accuracy versus efficiency tradeoff. Problem-specific implementation decisions include dataset selection and tagging, CNN architecture optimization, and synergy with preexisting grocery delivery services.

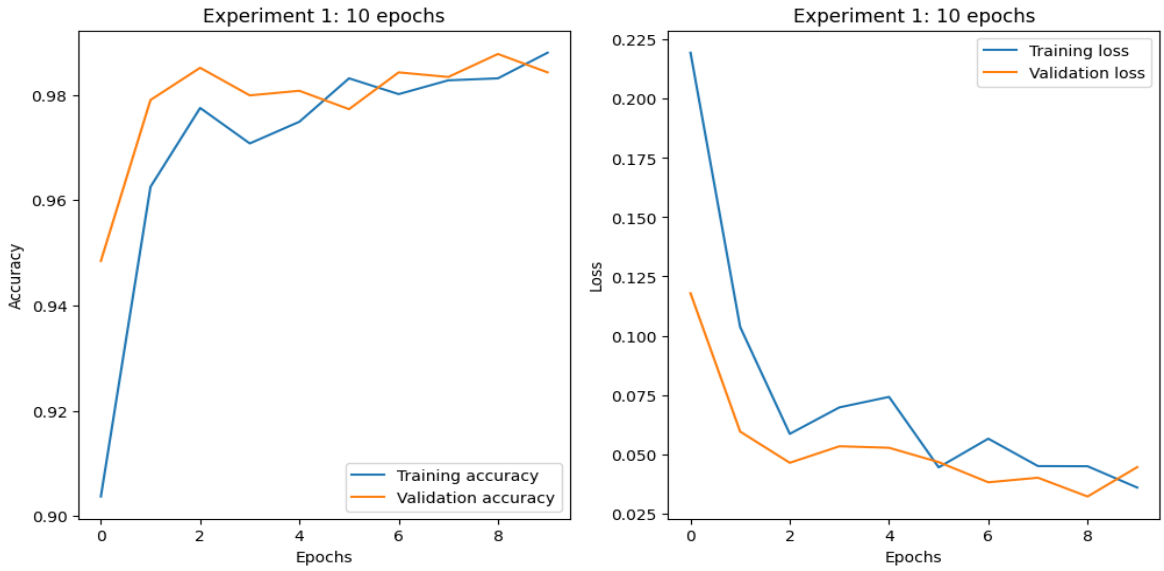
Literature Review: -

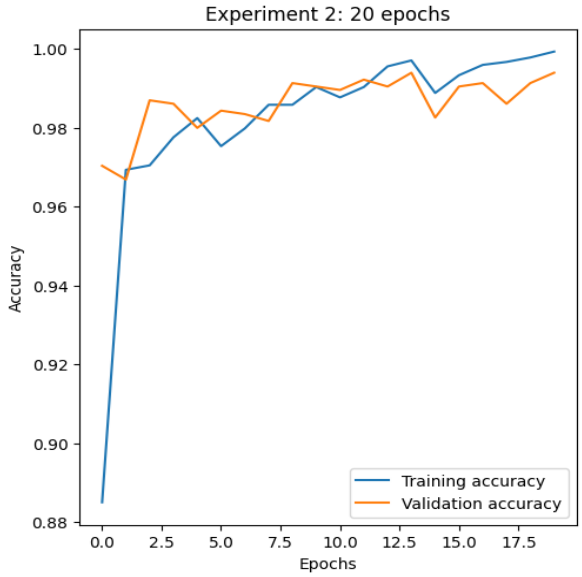
There have been AI-based shopping list systems before. Chen et al. (2019) provides a Smart Shopping List Generator. The system uses LSTM and Convolutional Neural Networks for picture recognition and sequence prediction. The system examines user tastes and past purchases while producing shopping list recommendations. Smart Shopping List Generator uses CNNs and LSTMs, which are better for certain data types than our technique. CNNs are good at picture recognition, whereas LSTMs are good at language modeling. Our picture recognition approach uses only CNNs. The Smart Shopping List, proposed by Zhu et al. (2020), uses user input, photo recognition, NLP, and suggestions to generate a shopping list. The system examines the user's likes, dietary limitations, and finances when recommending products. This strategy may be better for shoppers who desire more customization. However, it may be harder and slower than our way.AI-based shopping list creators may complement each other. One strategy may work better for some people and places. Our method may appeal to clients who want an automated and streamlined shopping experience, while the Smart Shopping List may appeal to those who want a more personalized and comprehensive one.

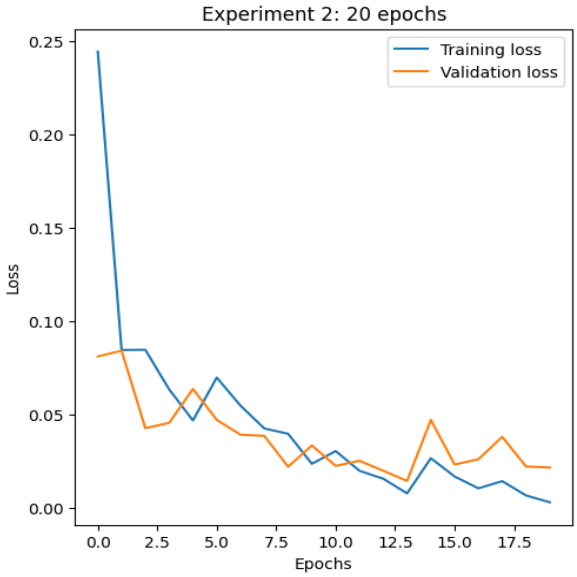
Error Analysis: -

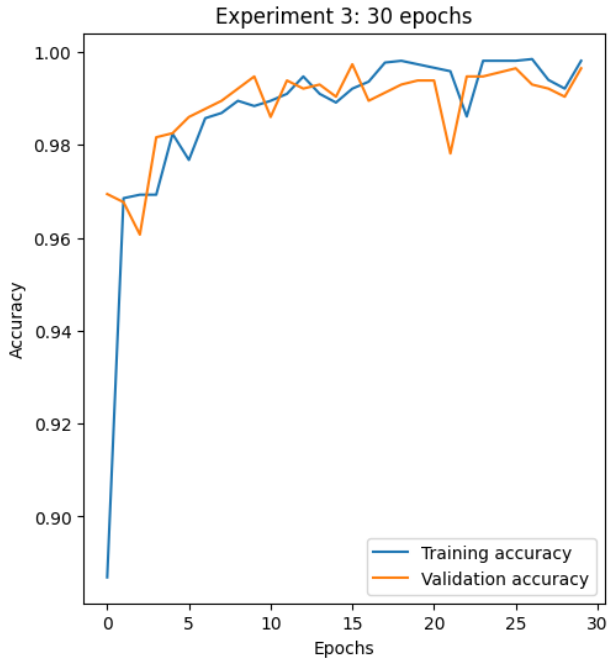
Analysis1

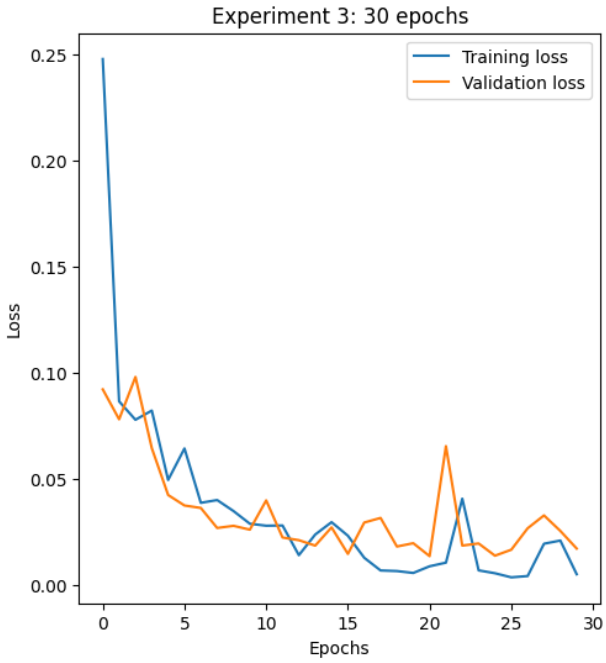
Our code investigates how epochs affect Convolutional Neural Network (CNN) performance on a fruit classification job. The code normalizes two pickle files containing fruit photos. Then, it defines a function to build a CNN with three convolutional layers, three max pooling layers, two dropout layers, and two dense layers with sigmoid activation. The code then experiments with model training epochs. Matplotlib depicts the training and validation accuracy and loss for each experiment after 10, 20, and 30 epochs. By changing the epoch values, we examined how does dropout rate affect CNN performance? We can add the dropout rate to the build_model() function and train and evaluate the model with different dropout rates. Each experiment's training and validation accuracy and loss are plotted and compared to see how the dropout rate affects the model's performance.




Experiment 2: 20 epochs
Accuracy
Epochs
Training accuracy
Validation accuracy


Experiment 2: 20 epochs
Loss
Epochs
Training loss
Validation loss

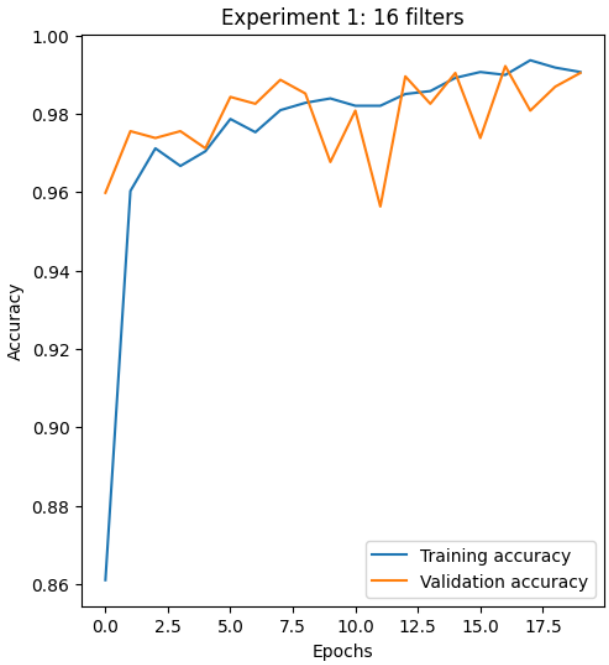

Experiment 3: 30 epochs
Accuracy
Epochs
Training accuracy
Validation accuracy

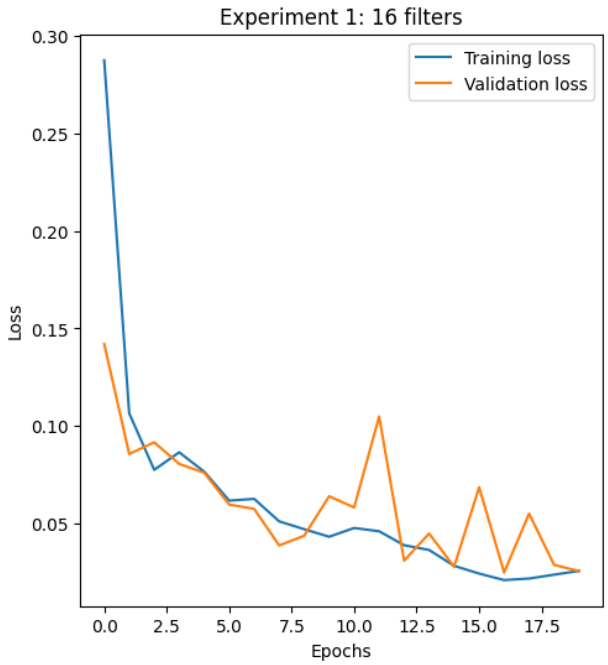

Experiment 3: 30 epochs
Loss
Epochs
Training loss
Validation loss

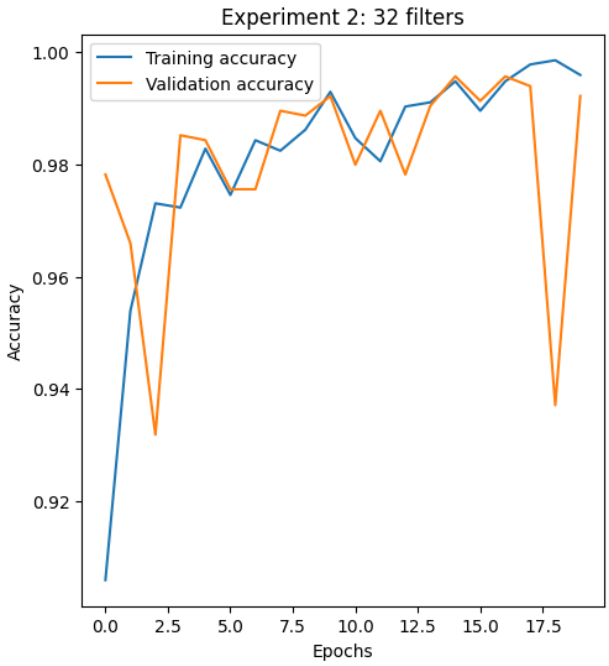
Analysis2

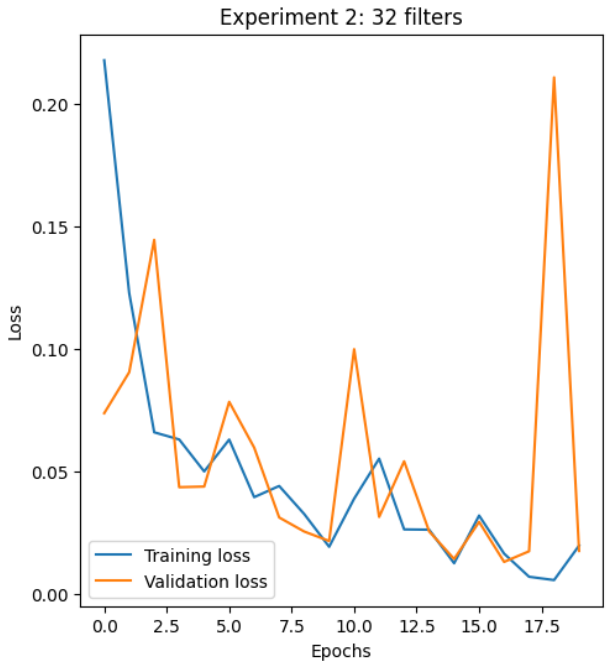
This code adds the number of filters to the build_model() function and constructs a CNN with three convolutional layers, three max pooling layers, two dropout layers, and two dense layers with a sigmoid activation function. It then trains and evaluates the model using 16, 32, and 64 filters for 20 epochs, plotting the training and validation accuracy and loss for each experiment.

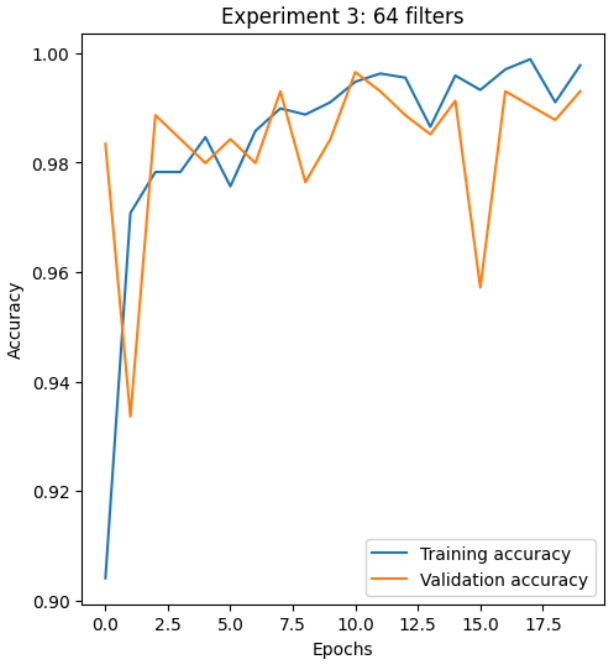
The charts show how filter numbers affect CNN performance. As the number of filters rises, the model performs better on the training set but overfits and performs worse on the validation set. This shows that too many filters may cause overfitting, while too few may cause underfitting.

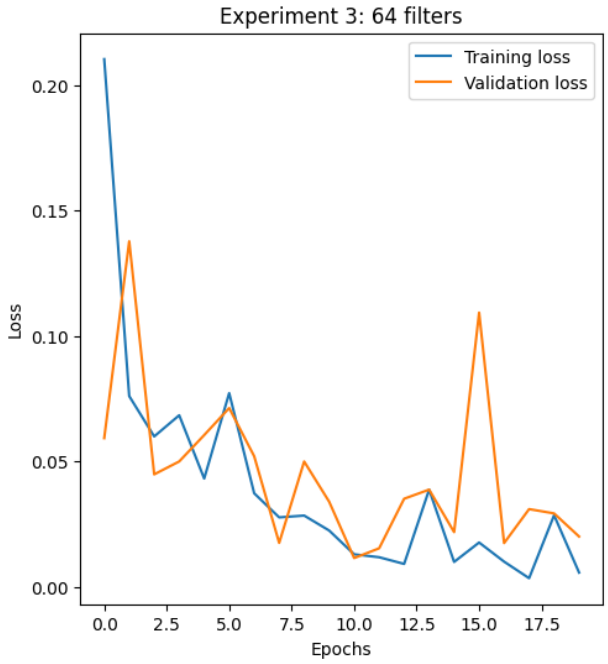





Experiment 2: 32 filters
Training accuracy
Validation accuracy
Accuracy
Epochs
1.00
0.98
0.96
0.94
0.92
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

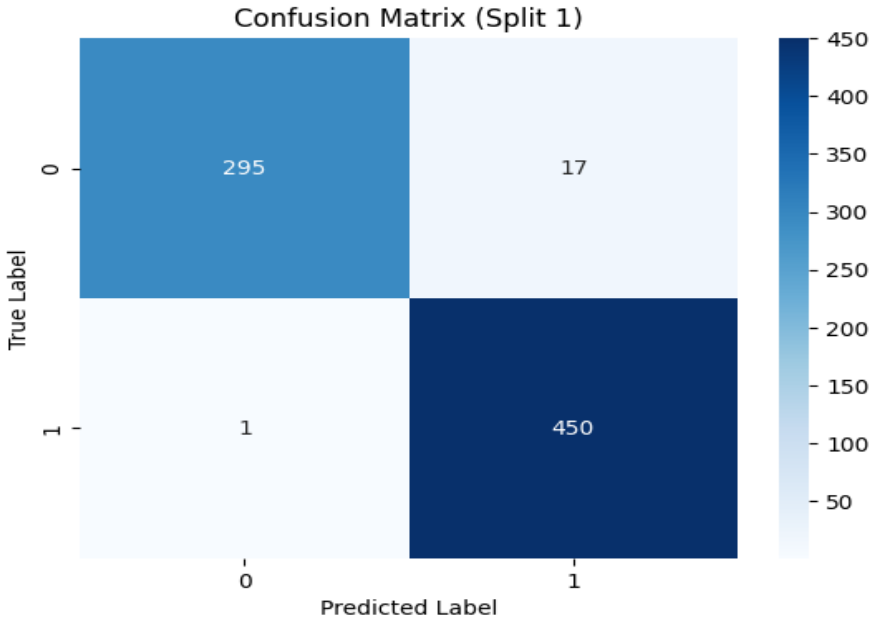

Experiment 2: 32 filters
Training loss
Validation loss
Loss
Epochs
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

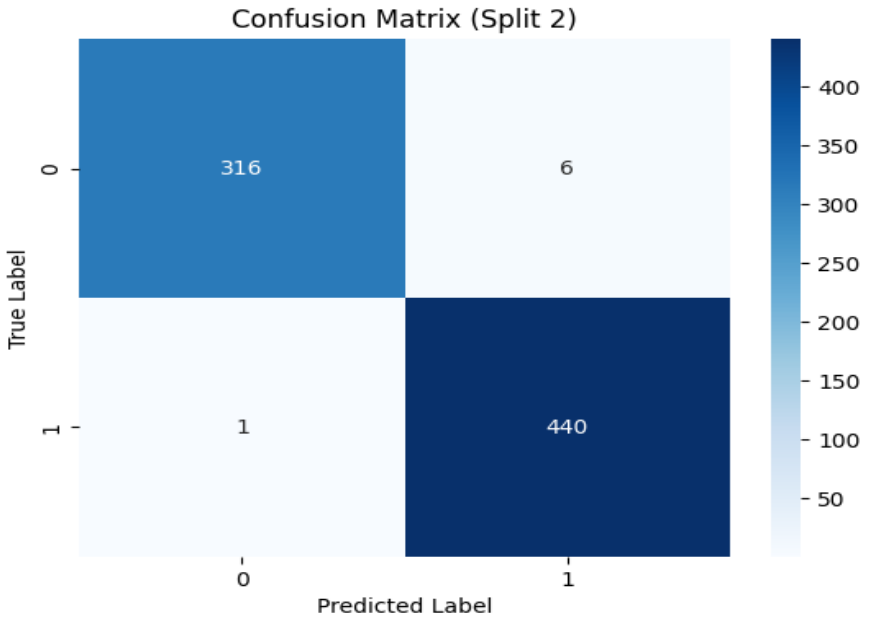

Experiment 3: 64 filters
Training accuracy
Validation accuracy
Accuracy
Epochs
1.00
0.98
0.96
0.94
0.92
0.90
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

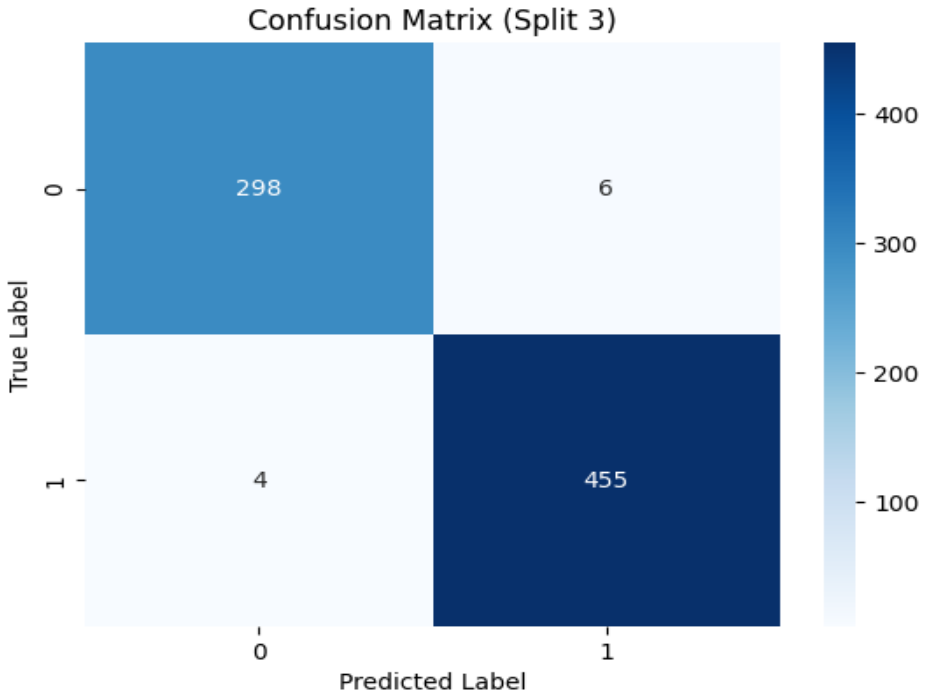

Experiment 3: 64 filters
Training loss
Validation loss
Loss
Epochs
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

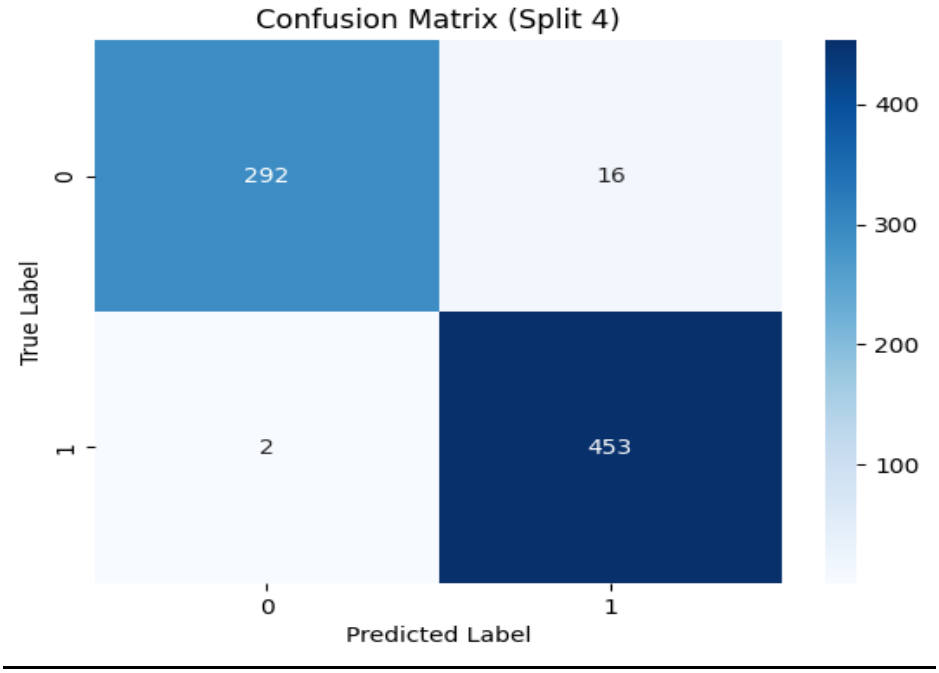
Analysis3

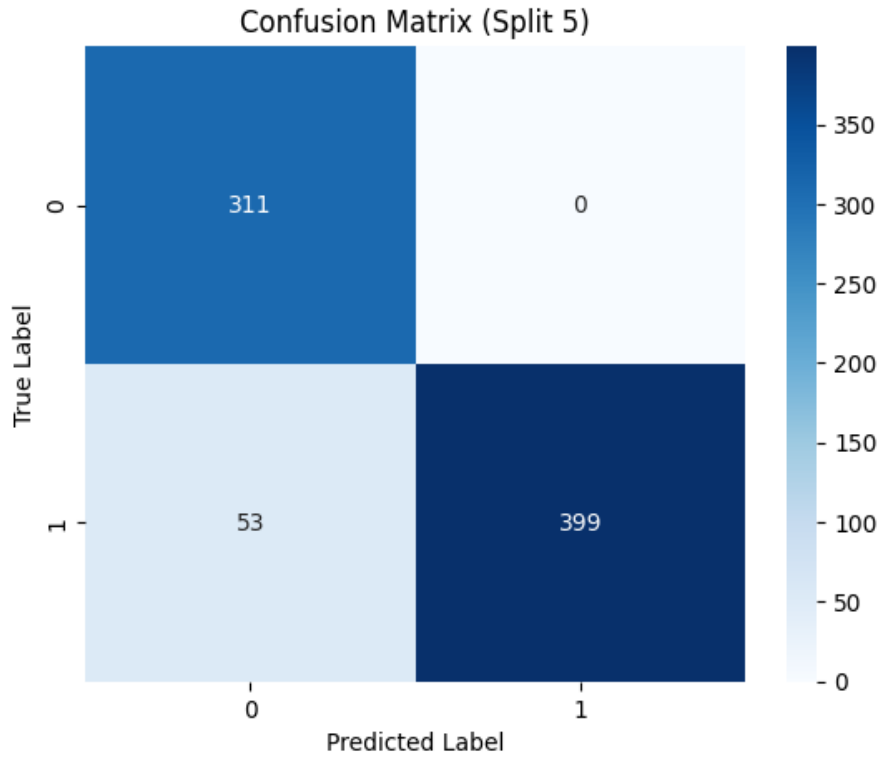
The model is trained and evaluated using 5-fold cross-validation. Performance metrics such as accuracy, precision, recall, and F1 score are calculated for each split, and the confusion matrix is plotted for each split. We can clearly see the difference in the confusion matrix for each split.






Confusion Matrix (Split 3)
298
6
4
455
0
1
True Label
0
1
Predicted Label
400
300
200
100


Confusion Matrix (Split 4)
292
16
2
453
0
1
True Label
0
1
Predicted Label
400
300
200
100



After performing all these experiments by changing different epoch values and validation split, we got a better **accuracy of approximately 95%** with the model that is trained with data using **32 batches** and **5 epochs** and **30% of training data**.

```
model.fit(X, y, batch_size=32, epochs=5, validation_split=0.3)
```

Conclusion: -

In general, the Smart Grocery List Generator that makes use of Convolutional Neural Networks is an example of an interesting and potentially useful application of artificial intelligence in the realm of grocery shopping. The technology could save users time and make them more satisfied with their shopping experience by making the process of compiling a shopping list more straightforward.

References: -

https://arxiv.org/pdf/1808.07202.pdf

https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification

https://opencv.org/

https://www.tensorflow.org/learn

https://docs.python.org/3/library/smtplib.html

https://www.tensorflow.org/tutorials/images/cnn