VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi-590014, Karnataka

A Mini Project Report on

Fruit-Vegetable Image Recognition

Bachelor of Engineering
In
Computer Science and Engineering

Submitted by

Name:Anindya Sindol (3GN22CS009)
Name:Advaith Deshmukh (3GN22CS003)
Name:Rohit Madiwal (3GN22CS038)

Under the guidance of
Prof. Ashwini Mankal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GURU NANAK DEV ENGINEERING COLLEGE
MAILOOR ROAD, BIDAR-585403, KARNATAKA

Phone: 08482- 226949 **Website: www.gndecb.ac.in**

GURU NANAK DEV ENGINEERING COLLEGE

BIDAR-585401, KARNATAKA

Phone: 08482- 226949 Website: www.gndecb.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project Report entitled "**FRUIT-VEGETABLE RECOGNITION**" is a bonafide work carried out by **ANINDYA SINDOL(3GN22CS009),ADVAITH DESHMUKH(3GN22CS003),ROHIT MADIWAL(3GN22CS038)**,in practical fulfillment of the requirements for the award of Bachelor of Engineering in VI **Semester Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during the year 2024-25. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini project has been approved as its satisfies the academic requirements in respect of major project work prescribed for the said degree.

Signature of Guide **Signature of HOD**

Prof.Ashwini Mankal **Dr.Anuradha A**

Name of the Examiner **Signature with Date**

1.
2.

ACKNOWLEDGEMENT

The mini project report on " **FRUIT-VEGETABLE RECOGNITION**" is the outcome of guidance, moral support and devotion bestowed onus throughout our work. Fort this we acknowledge and express out profound sense of gratitude and thanks to everybody who have been a source of inspiration during the project work

First and foremost, we offer our sincere phrases of thanks with innate humility to our Dr.Suresh R Reddy,Principal & Dr.Dhananjay M ,Vice Principal for their support and encouragement. We feel deeply indebted to our Dr.Anuradha A,HOD CSE for right help provided from the time of inception till date. I would take this opportunity to acknowledge our Mrs. Ashwini Mankal ,Asst Professor CSE Dept who not only stood by us as a source of inspiration, but also dedicated his time to enable us to present the project on time.

1.Name: ANINDYA SINDOL(3GN22CS009)

2.Name:ADVAITH.DESHMUKH(3GN22CS003)

3.Name:ROHIT MADIWAL(3GN22CS038)

ABSTRACT

This report delves into the advancements and applications of image recognition technology, a pivotal aspect of artificial intelligence that enables computers and systems to identify and process images in a way that is similar to human visual perception. The objective of this study is to explore the underlying algorithms, particularly deep learning and convolutional neural networks (CNNs), that have significantly enhanced the accuracy and efficiency of image recognition tasks. We examine a range of real-world applications, from facial recognition for security purposes to automated image tagging in social media, and assess the implications of these technologies on privacy and ethical standards. Additionally, we highlight the challenges faced in image recognition, such as bias in training data and the need for robust datasets to improve model performance. By analyzing current trends and future directions in image recognition, this report aims to provide a comprehensive overview of its significance in various industries and its potential to transform human-computer interaction. The findings underscore the importance of continued research and development in this dynamic field, as well as the necessity for ethical considerations in its deployment.

TABLE OF CONTENT

Introduction

In today's digital landscape, the ability to interpret and understand visual data has emerged as a transformative capability across various sectors. Image recognition, a subset of computer vision, utilizes advanced algorithms and machine learning techniques to identify and classify objects within digital images. From enhancing security systems with facial recognition to enabling automated quality control in manufacturing, the applications of image recognition technology are both diverse and impactful.

The rapid advancements in hardware and software, coupled with the exponential growth of available image data, have fueled significant progress in the field. Deep learning, particularly convolutional neural networks (CNNs), has revolutionized how machines process and understand visual information, achieving accuracy levels that were previously unattainable. As a result, image recognition systems are becoming increasingly integral in fields such as healthcare, retail, autonomous vehicles, and social media.

This report will delve into the fundamental principles of image recognition, examine the underlying technologies that drive its success, and explore its numerous applications and implications in various industries. Additionally, we will discuss the challenges that persist in the realm of image recognition, including ethical considerations, data privacy, and the need for robust training data. As we navigate through the intricacies of this technology, it becomes evident that image recognition is not just a functional tool, but a pivotal innovation shaping our future.

TECHNOLOGY USED

Public Datasets: Datasets such as ImageNet, CIFAR-10, and MNIST.

Custom Data: Domain-specific images collected via APIs, web scraping, or manual uploads.

Custom architecture optimized for the target dataset.

Layers specifically tuned for feature extraction, activation, and pooling.

Implementation of data augmentation to improve generalization. The system will be trained, validated, and tested on a benchmark dataset to ensure accuracy and efficiency

Preprocessing

Resize images to a consistent size.

Normalize pixel values to a range (e.g., 0–1).

Apply techniques like noise reduction or histogram equalization.

Model Selection

Choose a model architecture based on the complexity of the task:

Convolutional Neural Networks (CNNs): Standard choice for image recognition.

Example architectures: ResNet, Inception, VGG, EfficientNet.

Transformers: Modern architectures like Vision Transformers (ViT) for advanced task **Pretrained Models:** Use transfer

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

Server Specifications

For a robust image recognition system, the following hardware specifications are recommended:

Processor (CPU)

- Minimum: Intel Core i5 or AMD Ryzen
- Prefer multi-core processors (at least 6 cores) for parallel processing.

Graphics Processing Unit (GPU):

- Minimum: NVIDIA GeForce GTX 1050 or equivalent
- Recommended: NVIDIA GeForce RTX 2060 or higher for faster processing
- Alternatively, consider using GPUs with CUDA support for training deep learning models.

SOFTWARE REQUIREMENTS

Development Tools and Libraries

- Programming Language:
  - Python (recommended) or JavaScript

Frameworks:

- TensorFlow, Keras, PyTorch for deep learning
- OpenCV for image processing tasks

Development Environment

- IDE:
  - Google Colab , PyCharm, or Jupyter Notebook for Python development.

Version Control:

- Git and GitHub for code versioning and collaboration.

Additional Software

- Containerization: Docker for creating isolated environments for application deployment.
- Cloud Services (optional)**:
  - AWS, Google Cloud, or Microsoft Azure for scalable storage and computing power.

SYSTEM DESIGN

Key Features of the Interface

1. **Image Upload and Selection**
   - **Drag-and-Drop Zone**: Allows users to drag and drop an image file for recognition.
   - **File Browser Button**: Enables image selection from the local storage.
   - **Camera Input** (Optional): Provides real-time image capture functionality.
2. **Recognition Results**
   - **Labeled Object Highlights**: Detected objects in the image are highlighted with bounding boxes, each labeled with the object name and confidence score.
   - **Confidence Visualization**: A color gradient or bar representing the confidence score (e.g., red for low, green for high).
3. **Visualization of Performance**
   - **Heatmaps**: Show regions of interest in the image to illustrate which areas contributed most to the recognition.
   - **Analytics Dashboard**: Provides a summary of:
     - Number of objects detected.
     - Accuracy metrics (e.g., precision, recall).
     - Processing time per image.
4. **Interactive Controls**
   - **Zoom and Pan**: Allows users to closely inspect regions of the processed image.

   - **Filter Options**: Users can filter results based on confidence score thresholds or specific object categories.

5. **Logs and History**
   - Displays a history of processed images with their recognition results.
   - Option to download the logs in a report format (e.g., PDF, CSV).

3. Layout Design

1. **Header Section**:
   - Title of the system (e.g., *Image Recognition Dashboard*).
   - Navigation links (e.g., Home, About, Help).
2. **Main Interface**:
   - **Left Panel**:
     - Image upload tools and options.
     - Settings (e.g., model selection, confidence threshold).
   - **Center Panel**:
     - Display of the uploaded image with recognition results overlaid.
   - **Right Panel**:
     - Analytics and additional result information (e.g., detected objects, confidence scores).
3. **Footer**:
   - Links to documentation, FAQs, and support.

4. User Workflow

1. **Image Input**: User uploads an image or selects a real-time input.

2. **Processing**: The system processes the image and generates recognition results.
3. **Results Display**: The processed image is displayed with overlays and details.
4. **Analysis and Export**: The user can analyze the results and export them if needed.

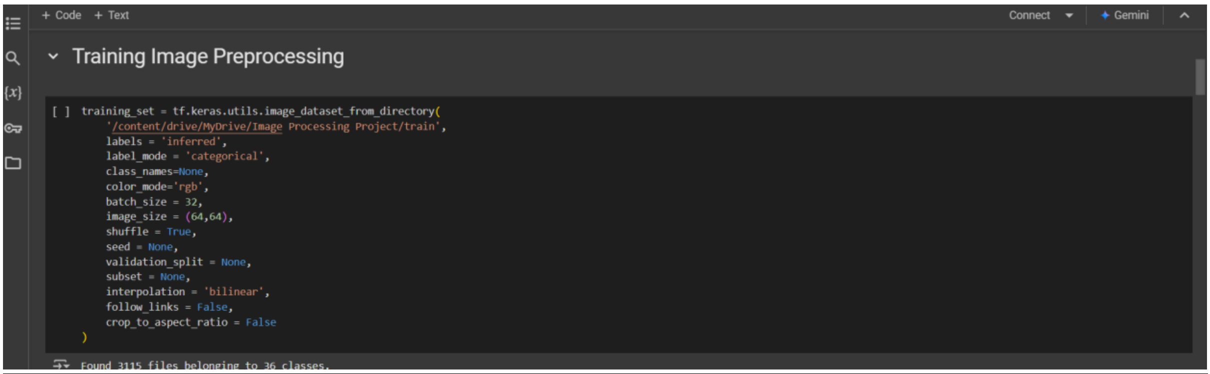
5. Visualization Tools

1. **Graphs and Charts**:
    - **Bar Charts**: Display the count of each recognized object.
    - **Pie Charts**: Represent the proportion of different categories detected.
2. **Interactive Visuals**:
    - Dynamic bounding boxes that can display details on hover.
    - Toggle to compare original and processed images.
3. **Performance Metrics**:
    - Real-time display of processing speed.
    - Historical trend graphs for recognition performance over time.

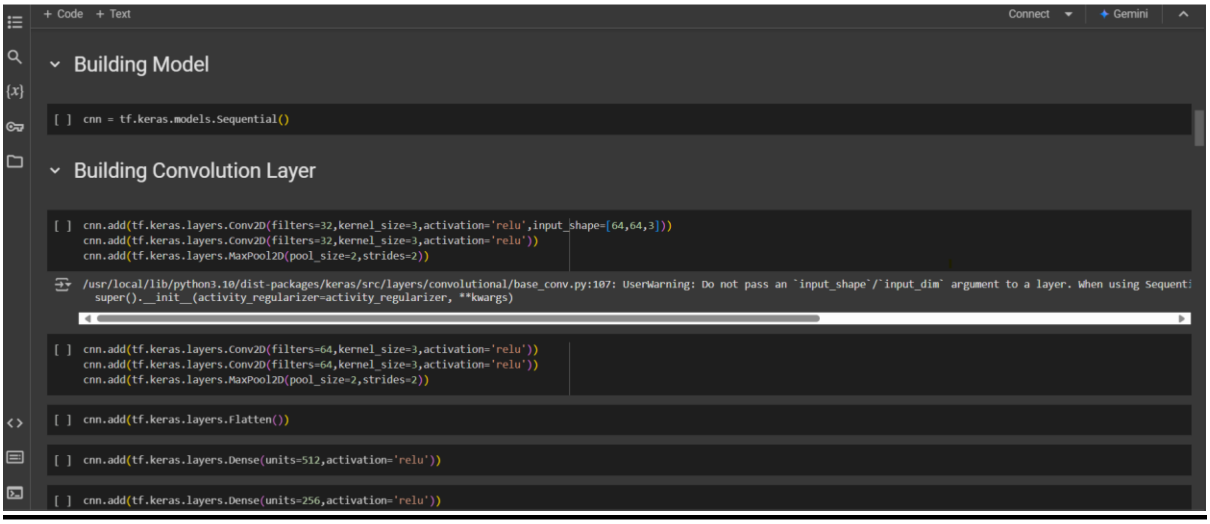
IMPLEMENTATION

Step 1: Import Required Libraries and dataset:

Step 2: Load a Pre-Trained Model and Then validate:



Step 3:Building model :



Building Model

```
cnn = tf.keras.models.Sequential()
```

Building Convolution Layer

```
cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation='relu',input_shape=[64,64,3]))
cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequenti
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

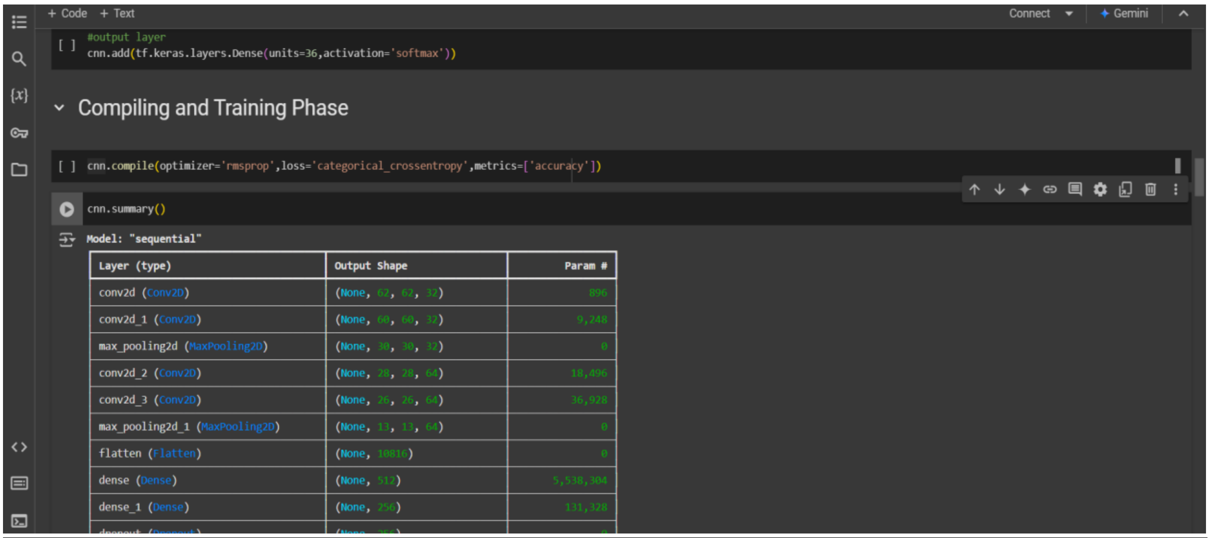
```
cnn.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3,activation='relu'))
cnn.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3,activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
```

```
cnn.add(tf.keras.layers.Flatten())
```

```
cnn.add(tf.keras.layers.Dense(units=512,activation='relu'))
```

```
cnn.add(tf.keras.layers.Dense(units=256,activation='relu'))
```

Step 4:Compiling and Training Phase:



```
#output layer
cnn.add(tf.keras.layers.Dense(units=36,activation='softmax'))
```

Compiling and Training Phase

```
cnn.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
cnn.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18,496
conv2d_3 (Conv2D)	(None, 26, 26, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 64)	0
flatten (Flatten)	(None, 10816)	0
dense (Dense)	(None, 512)	5,538,304
dense_1 (Dense)	(None, 256)	131,328

Step 5:Accuracy and training visualization:

Calculating Accuracy of Model Acheived on Validation set

```
[ ] print("Validation set Accuracy: {} %".format(training_history.history['val_accuracy'][-1]*100))
```

```
Validation set Accuracy: 95.72649598121643 %
```

Accuracy Visualization

Training Visualization

```
epochs = [i for i in range(1,33)]
plt.plot(epochs,training_history.history['accuracy'],color='red')
plt.xlabel('No. of Epochs')
plt.ylabel('Training Accuracy')
plt.title('Visualization of Training Accuracy Result')
plt.show()
```

Step 6: Testing and Evaluation

+ Code + Text Connect Gemini

Test set Evaluation

```
test_set = tf.keras.utils.image_dataset_from_directory(
    '/content/drive/MyDrive/Image Processing Project/test',
    labels = 'inferred',
    label_mode = 'categorical',
    class_names=None,
    color_mode='rgb',
    batch_size = 32,
    image_size = (64,64),
    shuffle = True,
    seed = None,
    validation_split = None,
    subset = None,
    interpolation = 'bilinear',
    follow_links = False,
    crop_to_aspect_ratio = False
)
```

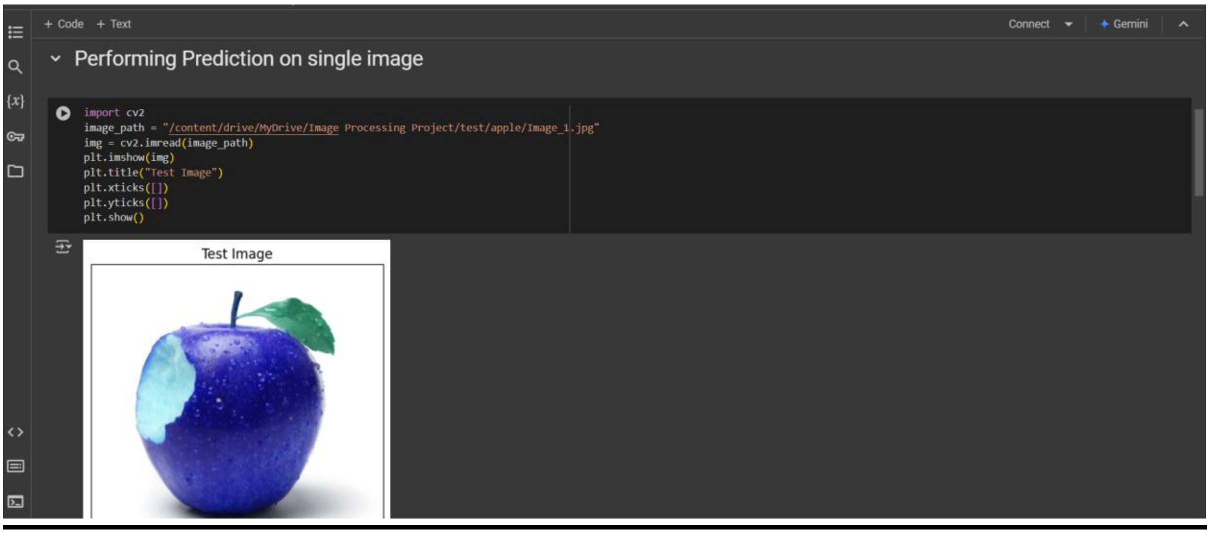
```
Found 359 files belonging to 36 classes.
```

+ Code + Text

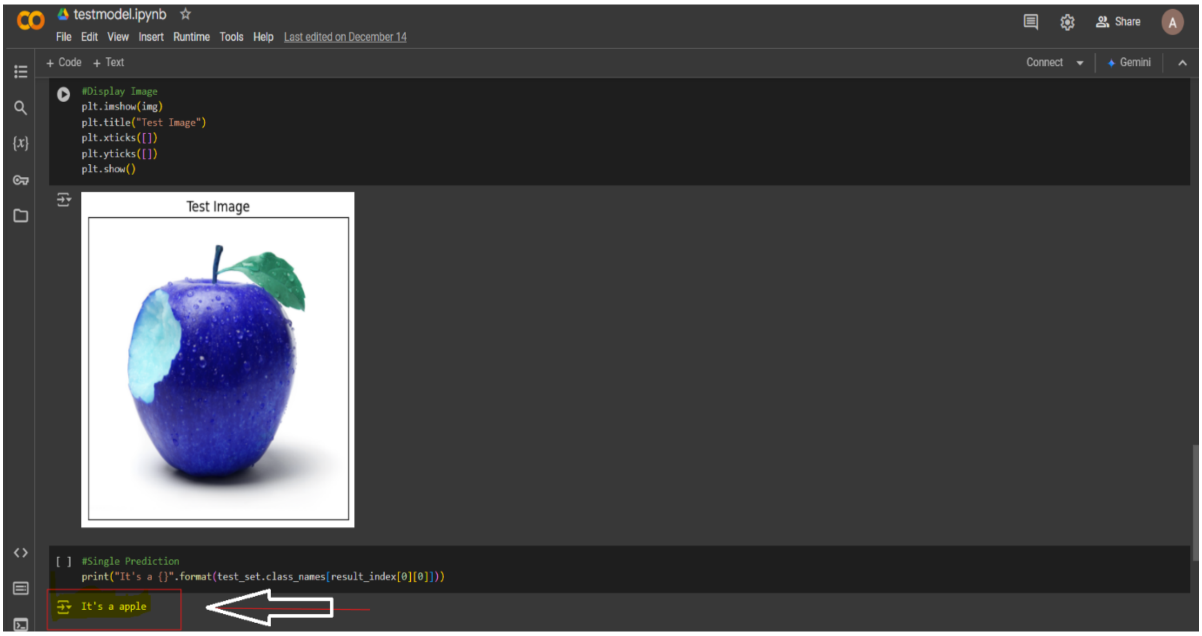
```
[ ] test_loss,test_accuracy = cnn.evaluate(test_set)
```

```
12/12 ━━━━━━━━━━━━━━━━━━━━ 46s 4s/step - accuracy: 0.9572 - loss: 0.5453
```

Step 7:System Testing and Deployment:



<u>Interpretation after results</u>



CONCLUSION:

With regards and honor’s to regularly increasing measures of data which is accessible and important data, it is very complex to say what data to search for and where to explore for it. Based on machine, methods have been generated to inspire the exploration and recovery process; suggestion is also one of the methods, which directs clients in their examination of data which is accessible by focusing for and the most important and efficient data have been suggested. Classification outlines have their initiating in a collection of zones of investigation, adding figures recovery and many more. They make the usage policies. Having beginning shown the considerations normal in data and information treatment of structures (data systems, choice emotionally supportive networks and classification structures) and set up a no mistakable capability between the proposal and personalization, so we by then displayed the most broad spread approaches used in building classifications for customers near to enormous sum methods used concerning classification systems. These thoughts were then shown by a talk of their valuable applications in gathering of spaces. Conclusively, we made thought of diverse methods used in surveying the idea of classification systems

REFERENCE

1. Buolamwini, J., & Gebru, T. (2018). "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification." *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency.*

2. Krizhevsky, A., Stuever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems.*

3. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE.*

4. Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry.* MIT Press.

5. Szegedy, C., et al. (2013). "Intriguing Properties of Neural Networks." *arXiv preprint arXiv:1312.6199.*

6.Youtube link:

(10603) Fruits and Vegetables Recognition System | Machine Learning Project - YouTube

7.Dataset link:https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition