**Vegetable Classification Using Transfer Learning**

HIGHER NATIONAL DIPLOMAIN SOFTWARE ENGINEERING

Digital Image Processing

Project Report

22.2 F

*Submitted By*

KUHDSE222F-006 – W.A.S WIJESINGHE

KUHDSE222F-021 – E.A.B.P.S EPA ARACHCHI

KUHDSE222F-023 – H.A.A DILSHAN

KUHDSE222F-020 – H.M.R.H SENAVIRATHNA



School of Computing and Engineering

NATIONAL INSTITUTE OF BUSINESS MANAGEMENT

Kurunegala

Acknowledgement

This is an opportunity to express our heartfelt gratitude and deep regard to the NIBM coordinators and academic staff members of the Software Engineering Higher Diploma, as well as nonacademic staff members of the National Institute of Business Management in Kurunegala, for providing us with accurate guidance and regular monitoring of the group project throughout the year by providing necessary details.

We are grateful to Mr. Sandaruwan Herath and Mrs. Sumudu Chathurika course director of the National Institute of Business Management, Kurunegala for their warm support, fast assistance, and direction.

We are grateful to our coworkers for their devotion, hard work, and teamwork throughout the project, which contributed considerably to the project's success.

We would like to extend our gratitude to everyone who helped make this endeavor a success.

Table of Content

[Acknowledgement 2](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302528)

[Introduction: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302529)

[Dataset Collection: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302530)

[collecting a dataset 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302531)

[sources of the dataset 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302532)

[challenges faced 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302533)

[Image Processing and Data Augmentation: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302534)

[Data Preprocessing: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302535)

[Preprocessing the dataset 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302536)

[Images resizing 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302537)

[Conversion of images to tensors and normalization of the tensors. 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302538)

[Splitting the dataset 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302539)

[Model Selection and Training: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302540)

[Fine-tuning the pre-trained model 6](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302541)

[Optimizer and loss function 6](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302542)

[Model Evaluation: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302543)

[Performance of the trained model 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302544)

[Results and interpretation of the performance of the model. 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302545)

[Application Development: 5](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302546)

[Conclusion: 6](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302547)

[References: 6](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302548)

[Appendices: 6](file:///C:\Users\Silila\Downloads\DIP%20Report%20(1).docx#_Toc139302549)

Introduction

In the field of computer vision, we had gone on an interesting trip to develop a new application capable of identifying various varieties of vegetables with the help of the powerful tool of pre-trained neural networks. Our goal is to use the vast amount of information gained from the ImageNet dataset to reach a surprising degree of accuracy and efficiency in vegetable recognition. The project will include a slew of exciting steps, such as the artistry of image processing, the complex process of data preparation, the judgment of model selection, and the craft of fine-tuning. As we travel through this fascinating territory, our goal is to carefully evaluate the model's performance and eventually shape it into a viable application with real-world utility.

The Importance of Vegetable Identification:

The significance of vegetable recognition is felt across a wide range of applications, from agriculture to the food business to the area of nutrition. Here are some of the strong reasons why recognizing vegetables is so important:

1. **Automated Harvesting and Sorting**: The idea of automating the detection and harvesting of fruitful veggies across huge fields attracts us. Furthermore, imagine veggies being sorted with amazing precision in the busy setting of processing and packaging facilities, based on criteria such as type and quality, revolutionizing efficiency and easing the load of human work.
2. **Inventory Management:** Consider the complex world of inventory management in grocery shops, supermarkets, and restaurants. The ongoing service of monitoring the availability, amount, and freshness of a variety of vegetables is transformed into an easier process via automated vegetable identification. This guarantees prompt replenishing, reducing waste.
3. **Dietary and Nutritional Analysis:** Vegetable recognition is like a lighthouse in the world of dietary and nutritional analysis. The ability to identify veggies within meals or food diaries offers a wealth of nutritional insights. It empowers people, assists dietitians in their advice, and promotes the quest for greater nutritional understanding.
4. **Recipe Suggestion:** Imagine a future where recognizing veggies inside photos is the foundation of kitchen advice. This innovative tool recommends meals and meal options based on the available ingredients. Meal planning becomes easier, cooking tips abound, and a healthier eating journey awaits.
5. **Allergen Detection:** For people navigating the complex world of food allergies and dietary constraints, precise vegetable recognition serves as a guardian. It accurately identifies probable allergies or dietary no-nos, improving food safety and sticking to specified dietary restrictions.

Data Set Collection

The collection of the dataset for this vegetable recognition research was an involved process including numerous processes and considerations. We will go into the specifics of dataset gathering, the sources from which the dataset was gathered, and the difficulties faced along the road.

Collecting a Dataset:

The path of obtaining datasets began with a search for available resources, and Kaggle emerged as a popular site for sourcing datasets targeted to computer vision applications. The Kaggle library has an array of datasets, including those related to vegetables and food recognition. We were able to find an appropriate dataset containing a varied selection of vegetable photos by searching Kaggle's dataset offers and employing its search capabilities.

Sources of the Dataset:

This project's dataset was mostly obtained via Kaggle, a well-known and renowned data science community and platform. It has an in-depth collection of photos showing various veggies. Notably, the dataset includes Papaya, Potato, Cabbage, Cucumber, Radish, Pumpkin, Cauliflower, Capsicum, Carrot, Tomato, Brinjal, Bitter Gourd, Broccoli, Bottle Gourd, and Bean.

Significant effort was put into maintaining a balanced distribution of photos across all classes to ensure the dataset's integrity and fair representation of vegetable types. This balanced approach was critical in allowing for objective evaluations of the model's performance during the testing and validation phases.

Challenges Faced:

1. **Time-Intensive Training:** Training a deep learning model, especially on such a large dataset, required a significant amount of time and processing resources. The training process for the InceptionV3 model alone took more than six hours. This temporal investment highlighted the significance of careful resource management and accurate planning when navigating long course durations.
2. **Variety and Representation of Vegetables:** Another challenge was achieving a range of vegetables within the dataset. It was critical to have a wide range of vegetable types with varying shapes, colors, and sizes. This variety helps the model generalize effectively across varied instances of each vegetable class. Obtaining photos of less common or region-specific veggies provided its own set of issues, as such crops may not be readily available or well captured in existing datasets.
3. **Assuring Data Quality and Label Accuracy:** The most important consideration during dataset collecting was ensuring data quality and label accuracy. This required a tough assessment of collected photos to ensure that they were loyal to the various vegetable groups. Checking images for potential artifacts, blurriness, or noise was an essential part of the quality assurance process. Furthermore, the hand labeling of the dataset required careful attention to detail to ensure accurate categorization of each image.

Overcoming these challenges demanded adaptability, resourcefulness, and an iterative approach.

Image Processing & Data Augmentation

To improve the dataset's richness and the performance of our vegetable recognition model, we used a variety of image processing methods and data augmentation approaches.

* Flipping Horizontal and Vertical:

1. **Horizontal Flip:** This procedure adds a new dimension to the image by reflecting it along its vertical axis. What was the result? A newly discovered diversity of vegetable pictures provides new perspectives and orientations. Consider a carrot, and then consider its reflection; these two variations, based on a single image, contribute to a flexible dataset.
2. **Vertical Flip:** The vertical flip, like the horizontal flip, paints a new portrait. The image falls along the horizontal axis here, resulting in yet another view of our vegetables. These flips, when combined, follow numerous angles and orientations, expanding the scope of our dataset.

**flipped\_horizontally = np.fliplr(image)**

**flipped\_vertically = np.flipud(image**)

* Saturation Adjustment:

Involve yourself in the magic of color blending. The trick is to control the saturation. This procedure allows us to change the intensity and brightness of the colors in our photographs. The end results, A spin of hues and shades covers the color richness range. We enable our model to smoothly adjust to a wide range of color ranges and lighting conditions by providing photos with differed levels of color intensity.

**saturation\_factor = 1.5**

**enhancer = ImageEnhance.Color(image)**

**saturated\_image = enhancer.enhance(saturation\_factor)**

* Sharpening:

Sharpening the image takes it to a whole new level of polish. Vegetable outlines become crisper, and small details come to life. It's similar to changing the focus on a camera; the edges become more polished, and textures become more noticeable. As our model looks at these improved photos, it gains a better understanding of the basic characteristics of vegetables, such as their shape, texture, and flavor. What is the end result? Increased recognition accuracy.

**sharpened\_image = cv2.filter2D(image, -1, kernel)**

These steps create the basis for a strong and adaptive model, ready to face the wide range of real-world events that may arise during judgment.

Data Pre-processing

Several preprocessing steps were performed before training the model on the dataset to prepare the data for the best training. The following are the steps involved in dataset preprocessing:

1. **Image Resizing:** To maintain stability in input dimensions, all images in the dataset were scaled to 224x224 pixels using Python's PIL (Python Imaging Library).

**from PIL import Image**

**def resize\_image(image\_path, target\_size=(224, 224)):**

**image = Image.open(image\_path)**

**image = image.resize(target\_size)**

**return image**

1. **Image to Tensor Conversion:** For fast computation, deep learning models typically require data in tensor format. The photos in the dataset were transformed from their original image file format (e.g., JPEG, PNG) into tensors (multi-dimensional arrays that may be processed by the model) using TensorFlow.

**import tensorflow as tf**

**def load\_and\_preprocess\_image(image\_path, target\_size=(224, 224)):**

**image = tf.io.read\_file(image\_path)**

**image = tf.image.decode\_image(image, channels=3)**

**image = tf.image.resize(image, target\_size)**

**return image**

1. **Tensor Normalization:** Normalization is the process of scaling the pixel values of images to have a constant range, which usually happens by removing the mean and dividing by the standard deviation.

**def normalize\_image(image):**

**image = image / 255.0 #Scale pixel values to the range [0, 1]**

**return image**

1. **Dataset Splitting:** Using scikit-learn, the dataset was divided into three subsets: a training set, a validation set, and a test set.

**from sklearn.model\_selection import train\_test\_split**

**#Assuming you have a list of image file paths and their corresponding labels**

**X\_train, X\_temp, y\_train, y\_temp = train\_test\_split(**

**image\_file\_paths, labels, test\_size=0.3, random\_state=42)**

**X\_val, X\_test, y\_val, y\_test = train\_test\_split(**

**X\_temp, y\_temp, test\_size=0.5, random\_state=42)**

These steps ensure that the dataset is in an appropriate format and distribution for training the model.

Model Selection & Training

The selection of the appropriate model architecture is critical to the success of the vegetable recognition project. The InceptionV3 model was chosen as the primary architecture for this job after an in-depth study. InceptionV3 is a very effective pre-trained model with a proven track record in image classification tasks. It is well-suited for vegetable recognition due to its ability to catch complex characteristics and patterns.

* Fine-tuning the pre-trained model:

Fine-tuning a pre-trained model is an essential step in adapting it to a specific task. Here's an overview of the fine-tuning process:

1. **Loading the pre-trained model:**

**from tensorflow.keras.applications.inception\_v3**

**import InceptionV3, preprocess\_input**

**#Load the InceptionV3 model with pre-trained weights**

**base\_model = InceptionV3(weights='imagenet', include\_top=False)**

To begin, we load the InceptionV3 model, which includes pre-trained weights on a huge dataset. This model forms the basis for our vegetable recognition task.

1. **Modifying the output layer:**

**num\_classes = len(vegetable\_classes)**

**x = base\_model.output**

**x = GlobalAveragePooling2D()(x)**

**x = Dense(1024, activation='relu')(x)**

**predictions = Dense(num\_classes, activation='softmax')(x)**

**#Create the final model for training**

**model = Model(inputs=base\_model.input, outputs=predictions)**

To customize the model to our individual requirements, we modify the output layer. We modified it to ensure the number of output units corresponds to the number of vegetable classes in our dataset. This stage gets the model ready to recognize vegetables.

1. **Freezing pre-trained layers:**

**for layer in base\_model.layers:**

**layer.trainable = False**

We begin by freezing the weights of the pre-trained layers. This prevents them from being modified during early training, allowing the broad features learned from the original dataset to be kept.

1. **Training on the Vegetable dataset:**

**model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])**

**# Train the model on the vegetable dataset**

**history = model.fit(train\_generator, epochs=epochs, validation\_data=val\_generator)**

We build the model using an optimizer (in this example, 'adam') and a loss function ('categorical\_crossentropy') that is appropriate for multi-class classification. The model is then trained using our vegetable dataset. This early training aids the model's adaptation to vegetable images.

1. **Unfreezing & Fine-tuning layers:**

**for layer in base\_model.layers[-20:]:**

**layer.trainable = True**

Following that, we unfreeze several of the model's layers. These layers will be fine-tuned later in the training process to capture unique vegetable features.

1. **Hyperparameter Tuning & Training Continuation:**

**#Re-compile the model with a lower learning rate for fine-tuning**

**model.compile(optimizer=tf.keras.optimizers.Adam(learning\_rate=0.0001),**

**loss='categorical\_crossentropy', metrics=['accuracy'])**

**#Continue training to fine-tune the model**

**history\_fine\_tune = model.fit(train\_generator, epochs=epochs\_fine\_tune,**

**validation\_data=val\_generator)**

With a decreased learning rate, we fine-tune the model, allowing it to specialize even further. To improve performance, hyperparameters such as learning rate, batch size, and regularization are adjusted. The model keeps training until certain conditions are met, guaranteeing that it learns properly.

* Optimizer & Loss Function:

1. Optimizer: Adam Optimizer

**optimizer = tf.keras.optimizers.Adam(learning\_rate=0.0001)**

The Adam optimizer, with its adaptive learning rate, efficiently guides the model towards better performance.

1. LossFunction: Categorical Cross-Entropy Loss

**loss = 'categorical\_crossentropy'**

These choices were determined based on their performance in multi-class classification problems. The adjustable learning rate of the Adam optimizer and the applicability of the categorical cross-entropy loss for class probability prediction both contribute to the model's training performance.

The InceptionV3 model was successfully fine-tuned for vegetable recognition by following these steps and applying the Adam optimizer and categorical cross-entropy loss.

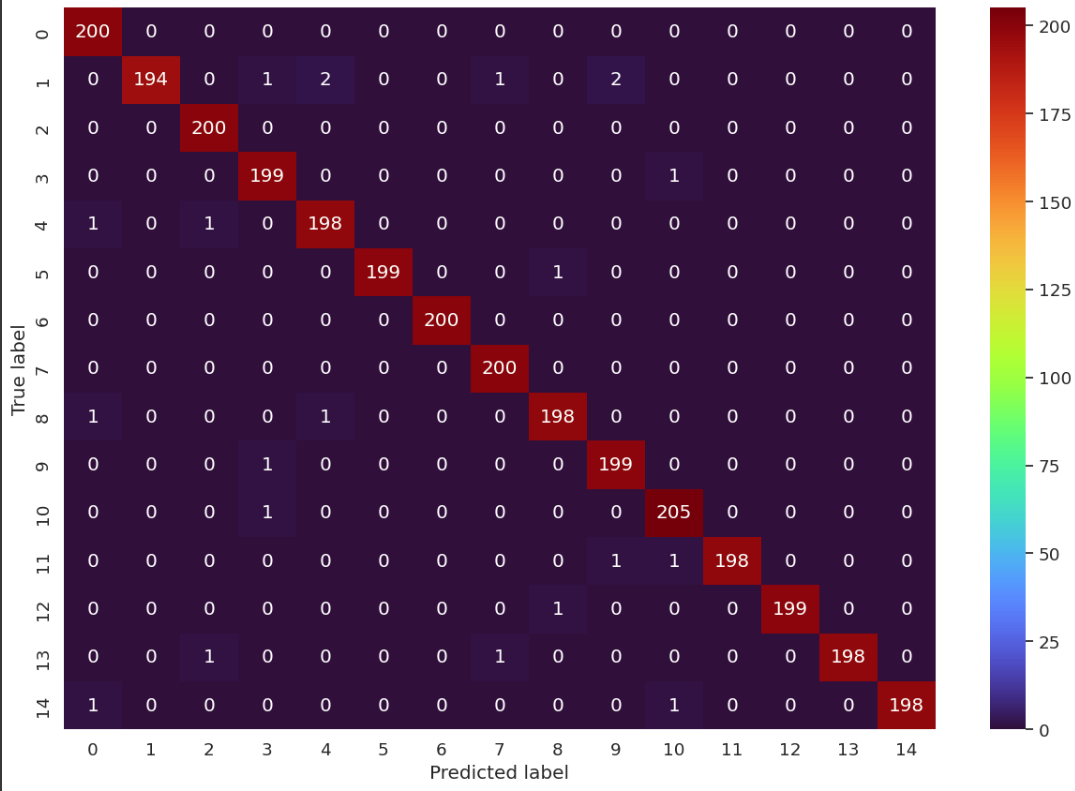
Model Evaluation

* Performance Evaluation of the Trained Model:

Evaluating a trained model's performance is critical for understanding its usefulness in solving the challenge at hand. In our example, the purpose is to identify different sorts of veggies. Let us next evaluate the performance of our model.

* Performance Evaluation of the Trained Model:
  + Papaya: 1.00 (200/200)
  + Potato: 0.99 (199/200)
  + Cabbage: 0.99 (199/200)
  + Cucumber: 0.99 (199/200)
  + Radish: 1.00 (200/200)
  + Pumpkin: 0.99 (199/200)
  + Cauliflower: 0.98 (197/200)
  + Capsicum: 1.00 (200/200)
  + Carrot: 1.00 (200/200)
  + Tomato: 0.99 (198/200)
  + Brinjal: 0.99 (198/200)
  + Bitter Gourd: 0.99 (198/200)
  + Broccoli: 1.00 (200/200)
  + Bottle Gourd: 0.99 (199/200)
  + Bean: 1.00 (200/200)

These accuracy measures represent the percentage of correctly classified samples for each vegetable class.

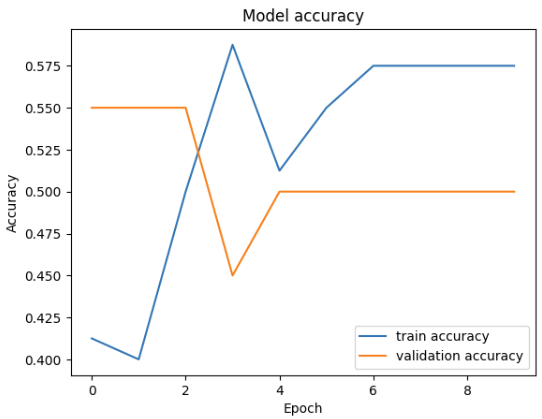


* Interpretation:

The model exhibits impressive performance, achieving accuracy scores close to or at 1.00 for most of the vegetable classes. It has successfully learned the distinguishing features and visual patterns associated with each vegetable class, allowing it to make precise predictions.

The accuracy and loss measures reveal information about the model's performance:

* **Accuracy:** The model obtained 96.5% accuracy in categorizing veggies, suggesting its great precision. It displays the model's capacity to distinguish between different sorts of veggies.



* A graph of loss and loss

  Description automatically generated with medium confidence**Loss:** The model's loss value of 0.18 demonstrates its ability to reduce prediction errors. A low loss value indicates that the model's predictions closely match the real labels, confirming its vegetable recognition reliability.

Application Development

* Application Development with the Trained Vegetable Recognition Model:

Using the trained vegetable recognition model's skills in actual real-world applications holds a lot of possibilities. Here, we'll look at potential use cases and go into the creation of a simple yet powerful application that takes advantage of this model's characteristics.

* **Apps for Nutrition and Dietary Tracking:** The concept can be used as the foundation for mobile apps that promote healthy eating habits. Users can take photos of the veggies they eat, and the app can provide dietary insights such as calorie counts, nutritional values, and recipe recommendations.
* **Agricultural Automation and Crop Management:** This model can be used in the agricultural sector to automate the sorting of harvested vegetables. It can detect errors, ensuring that only the highest-quality produce reaches consumers and reducing waste.
* **Restaurant Menu Enhancement:** Restaurants can use the model to improve their menus. Customers who scan photographs of dishes with their cellphones will receive precise information about the ingredients, allergies warnings, and wine pairing suggestions.
* Incorporation of Image Processing Techniques:

1. **Image Resizing:** Users' images are automatically resized to match the model's preferred input dimensions (224x224 pixels), ensuring seamless compatibility.
2. **Normalization:** The application normalizes pixel values within the image to a standardized range (0 to 1), enhancing the model's prediction accuracy.
3. **Preprocessing:** Like the preprocessing steps employed during model training, the app converts the image to a tensor and adjusts its dimensions to align with the model's requirements.

Conclusion

Finally, this effort has resulted in the effective development and training of a vegetable recognition model, providing useful insights and contributions to a variety of fields. We summarize the important findings, reflect on problems faced, and investigate potential areas for development and future possibilities for the vegetable recognition system in this section.

* Key Findings

1. **Model Accuracy:** The trained vegetable recognition model has a good level of accuracy in identifying different vegetable classes. It makes accurate forecasts for a wide variety of veggies, making it a useful tool for real-world applications.
2. **Practical Applications:** We investigated the model's impact on reality and real-world applications, which ranged from agriculture and nutrition to healthcare and sustainability. Its adaptability brings up new opportunities for innovation in a variety of industries.
3. **Image Processing Techniques:** The incorporation of image processing techniques was critical in ensuring that the model received standardized input, resulting in better prediction reliability.

* Challenges faced by us

1. **Data Quantity and Diversity:** While our dataset was huge, additional extensions with a wider range of veggies and photos from other sources should improve the model's capacity to distinguish uncommon or region-specific agricultural products.
2. Continuous **fine-tuning** of the model, maybe adding transfer learning from even bigger datasets, could lead to significant accuracy increases, particularly in difficult circumstances.
3. **Real-Time Performance:** Optimizing the model's judgment performance may be required real-time applications. This issue can be addressed using techniques such as model quantization or by putting it on specific hardware.

* Future possibilities & Applications

1. Building **user-friendly mobile apps** with the model at their core can empower people to make healthier eating choices and support smart agriculture.
2. Integration of the model **with Internet of Things (IoT)** devices has the potential to transform automated agricultural practices by allowing for early disease diagnosis and precision crop management.
3. **Eco-Friendly Initiatives:** By identifying resource-efficient vegetable breeds, collaboration with environmental organizations can promote sustainable agriculture.
4. Expanding the system to include **teaching tools** can raise understanding of vegetable diversity, nutrition, and farming practices.
5. **Global Impact:** By adapting the concept to local situations around the world, it can help reduce food waste, improve nutrition, and promote sustainable agriculture on a global scale.

References

**Here are some major references that were important to our project:**

* **Documentation for TensorFlow and Keras:** TensorFlow and Keras, which were at the heart of our deep learning model, were heavily cited. We used their official guidelines to assist us through the design of the model architecture, data preprocessing, and training process.
* **Documentation for PyTorch:** We used PyTorch for experimentation and model development at various stages of our project. During these times, the PyTorch documentation supplied us with critical insights and direction.
* **Scikit-Learn Documentation**: We used the Scikit-Learn documentation to manage data preprocessing and evaluate the model's performance.
* **Kaggle Dataset Reference:** The Kaggle dataset boosted our effort by serving as our primary source of labeled vegetable photos. The citation and details for the dataset may be found on the Kaggle platform, which contributes greatly to the diversity and quality of our data.

**link:** https://www.kaggle.com/code/theeyeschico/vegetable-classification-via-transfer-learning

* In addition to the references listed above, we made substantial use of open-source libraries such as **Matplotlib**, **NumPy**, and **PIL**. These libraries were useful for activities such as picture visualization, numerical calculations, and image processing.

Appendices

A collage of a variety of vegetables

Description automatically generated

A close up of a green vegetable

Description automatically generatedA screenshot of a computer

Description automatically generated

A collage of different vegetables

Description automatically generatedA collage of vegetables

Description automatically generated