### **Abstract**

Agriculture plays a vital role in the economic and social fabric in Bangladesh, contributing significantly to the country’s revenue and the livelihood of millions. However, the agricultural sector faces challenges such as the need of efficient crop monitoring and classification, which are essential for optimizing yields and ensuring the right crop management practices, to address this, we developed an image based classification system using deep learning and Convolutional Neural Network (CNN) to identify and classify Bangladeshi seasonal vegetables. Our project implemented a Custom CNN model alongside several fine-tune transfer learning models including ResNet50, DensNet121, VGG16, MobileNetV2, EfficientNetB0, and InceptionV3 to evaluate and compare their effectiveness in accurately classifying seasonal vegetables. Each model was precisely evaluated using matrices such as loss curves, confusion matrix, precision, recall, and F1 score. We used Explainable AI techniques (XAI) such as Grad-CAM, Eigen-CAM, Gran-CAM++, and LIME to help us understand how the models make their predictions. These techniques are used to visually highlight the specific region of an image which influences the model prediction. These techniques make it easier to understand how the model interprets the images. The result indicated

that transfer learning models (ResNet50) outperformed the Custom CNN model by achieving highest classification performance, obtaining an accuracy rate of 99.33% with a loss curve that demonstrates the best fit.

**Introduction**

Over the past few years, agriculture has been greatly improved by technology, leading to more precise and efficient ways to monitor crops, particularly in classifying seasonal vegetables. With advances in artificial intelligence and machine learning, tools for effortlessly identifying and monitoring crops have become more convenient and effective. This progress is primarily essential in regions like Bangladesh, where seasonal vegetable farming plays a central role in agriculture.

To facilitate this transformation, we have developed the dataset named “SeasVeg”, a newly established resource focused on Bangladeshi seasonal vegetables. This dataset comprises ten categories of vegetables commonly cultivated in regions such as Pabna and Dhaka containing Carica Papaya, Momordica dioica, Abelmoschus esculentus, Lablab purpureus, Trichosanthes cucumerina, Trichosanthes dioica, Solanum iycopersicum, Brassica oleracea, Momordica charantia, and Raphanus sativus. It is intended to assist agricultural science research, especially in image based crop classification and supervising.

Our dataset is categorized under the domain of “Agricultural Science”. We selected this domain to enhance our understanding of crop production and its contribution to the agricultural system. This dataset helps us to examine the intricate connection between the vegetable types and their features. Furthermore, it contributes to the progress of precision agriculture delivering insights into how technology can improve crop yield, sustainability, and resource use in Bangladesh's agriculture sector. This dataset aims to help develop innovative solutions that promote efficiency and sustainability in agricultural practices.

Data Preprocessing:

Firstly, we have 1500 images from our dataset Seasveg. All of these 1500 images are classified into 10 classes with 150 images in each class respectively. Next, we meticulously removed a few corrupted images that we had found in the dataset and the quantity of corrupted images was four. Furthermore, our dataset had some outlier images which we resized.

EDA:

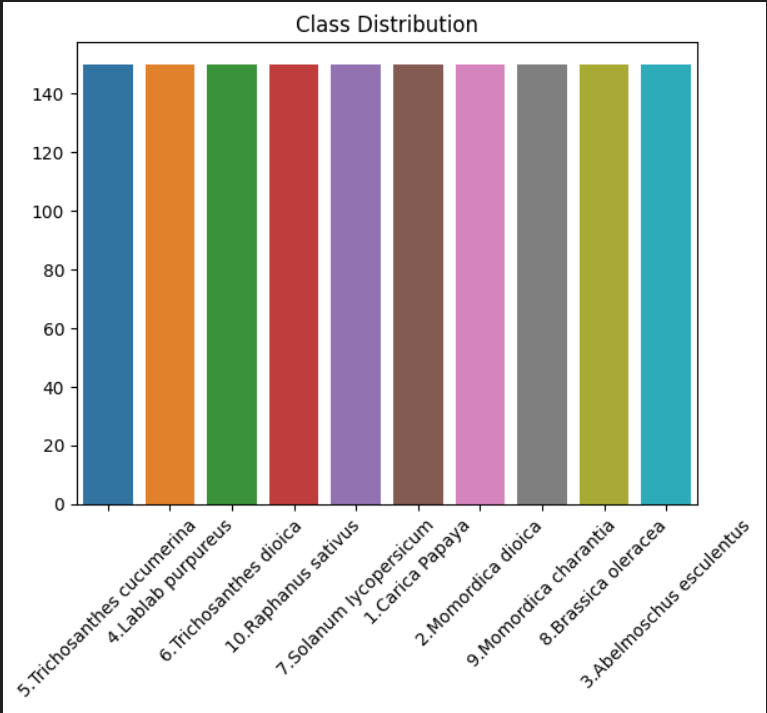
After the dataset has been processed, we performed Exploratory Data Analysis(EDA) where we used data visualization techniques to get a better understanding and in depth visualization of our dataset. The main goal behind this was to observe for any imbalance of images within the classes of the dataset. This was an essential step which needed to be performed in order to ensure the quality and integrity of the dataset which would later on guide us to the selection of appropriate machine learning models.

Methodology:

**Data Description**

Seasonal vegetables are abundant sources of vitamins, fiber and essential nutrients. Proper usage and consumption of these vegetables are vital for maximizing their benefits.This dataset aims to provide a variation of images of Bangladeshi seasonal vegetables, categorized into ten groups namely, Carica papaya, Momordica diocia, Abelmoschus esculentus, Lablab purpureus, Trichosanthes cucumerina, Trichosanthes diocia, Solanum Lycopersicum, Brassia oleracea, Momordica charantia and Raphanus sativus.Figure 1… Figure 2 illustrates the classification based on names and shapes of these vegetables

The dataset also aims to educate the public and motivate individuals to focus on cultivating seasonal vegetables to meet growing demands .



**Image Augmentation**

We implemented a data augmentation pipeline for our image classes in order to incorporate it into the machine learning model seamlessly. We utilized PyTorch to define separate transformations for training, validation and testing phases. The training phase contains resizing and cropping , color adjustments and translation scaling to make the images more robust for the model. Images are later normalized with a mean standard deviation of 0.5. For both validation and testing, simple transformations are used to resize the images to 224x224 followed by tensor normalization.

**Machine Learning Models**

In order to apply computer-aided analysis we implemented certain machine learning techniques on our “SeasVeg” dataset namely Resnet-50 , Densenet-121, Mobile net V2,Efficient net B0 and Custom CNN model.

Resnet-50: It is a deep convolutional neural network with 50 layers and it is designed to address the problem of vanishing gradients in deep networks. Resnet allows the model to skip layers and add individual output to subsequent layers.

DenseNet-121: DenseNet is a connected CNN model with 121 layers. Itsd purpose is to improve gradient flow and feature reuse by connecting every layer to every other layer subsequently. In this case, each layer receives inputs from all previous layers and passes it on all subsequent layers. This is ultimately achieved using all the dense blocks where the subsequent outputs are connected together in a transformation function. The connectivity pattern allows the reduction of parameters compared to traditional CNN.

VGG16: It is a classical CNN model which is efficient for its visual geometry function. VGG16 works through several convolution layers by max pooling all the data and reducing spatial dimensions. The whole network works with a fully connected lat=yer and a simplified classification. Compared to other architectures, VGG16 requires more parameters despite lacking residual connections or dense blocks.

EfficientNet-B0: This CNN model is a base model of the efficient family which works on accuracy and balances computational efficiency by scaling up the depth and resolution of the network. It uses compound scaling which scales the number of layers, channels, and input image size simultaneously. A bottleneck technique is used to optimize the channels and squeeze information throughout the whole pipeline. The base B0 model is a resource-constrained environment for image recognition and object detection.

MobileNetV2: It is a lightweight CNN architecture design for embedded devices which introduces a residual block named inverted residual block where both input and output are within the bottleneck layers while all inner layers are expanded. MobileNetV2 employs split convolutions, depth wise separate convolution into all of its channels and reduces the computation for each image.

Along with these models a custom CNN model is also used to determine the accuracy of the provided images.

Before implementing our machine learning models, we split the dataset into training (70%), validation (20%), and testing (10%) sets . Afterwards, we implemented our machine learning models on the dataset and the parameters we used to train the models were batch size, epoch, Learning Rate , Early stopping, and lastly auto-cast for fast computation. The models are configured with a cross-entropy loss function and an Adam optimizer for better transfer learning. During each epoch, the training phase is continuously computing each gradient while updating for mixed precision, The validation phase checks whether or not the model is performing for the unseen data.Early stop is used to monitor the validation loss for the improvement of the model.

Moreover, we applied Explainable AI Techniques to acquire a better grasp of our model interpretability by visualizing the key features which are further describe below:

Grad-CAM: It mainly highlights the part of an image that are most important for the model’s decision about a specific class, making it easier to understand what the model is focusing on.

Grad-CAM++: It basically improves on Grad-CAM by generating sharper and more detailed heatmaps, especially helpful when there are multiple objects or subtle details in the image.

Eigen-CAM: This shows the overall regions in an image that the model finds important. Without focusing on any specific class, giving a general sense of what the model pays attention to.

**Loss Curve:**

**Underfit:**

Machine learning model is underfit when it has a failure in capturing the observation of the training dataset. In this instance, when the model can not perform well with the training data and faces high training loss while simultaneously performing poorly on unseen data which leads to a high validation loss in the loss curve. Then we can say the model is underfitting.

**Overift:**

A machine learning model is said to be overfit when it is not only learning patterns from the training set but also the noise of the dataset. In this case, if we see in the loss curve that the training loss is very low but the validation loss is very high, we can understand that the model is not performing well for new data. Thus we can say that the model fails to generalize unseen data and it is overfitting.

**Best Fit:**

A machine learning model is said to have best fit when it has the correct balance between the validation and training loss.That means the model is perfectly capturing the training data and also performs well on the unseen data that we have in the validation set.

**Result & Discussion:**

The deep learning architectures ( Resnet50, VGG16, DenseNet121, EfficientNetB0, and MobileNet-V2, Custom CNN model) implemented in our study allocated the dataset in such a way that we used 70% for training, 20% for validation, and 10% for testing set. For our computation purposes, we used Automatic Mixed Precision (AMP) which accelerates the training process by performing computation in lower precision while showing the accuracy in a selective way in higher precision. In order to train our model, the parameters we used were batch size, selective epochs, learning rate, optimizer, activation function, and early stopping. In table 1, we can observe the employed values for the respective parameters while running the pre-trained and custom CNN models:

| Batch Size | Epochs | Early Stopping | Learning rate | optimizer | Activation function |
| --- | --- | --- | --- | --- | --- |
| 16 or 32 | 50 | 5 | 0.001 | Adam | ReLu |

Table1

After running all our custom CNN and pre-trained models, the model that provided us the best accuracy was ResNet50 which was 99.33%. Even though all the other models and custom CNN models were not that far behind, ResNet50 provided the best accuracy in all instances. In Table-2, we can observe the following:

| Machine Learning Model | Accuracy | Precision | Recall | F1-score |
| --- | --- | --- | --- | --- |
| ResNet50 | 99.33% | 0.9937 | 0.9933 | 0.9933 |
| VGG16 | 84.00% | 0.8475 | 0.8400 | 0.8370 |
| DenseNet121 | 88.67% | 0.9090 | 0.8867 | 0.8908 |
| EfficientNetB0 | 99.33% | 0.9938 | 0.9933 | 0.9934 |
| MobileNet-V2 | 98.67% | 0.9875 | 0.9867 | 0.9867 |
| Custom CNN model | 88.00% | 0.8924 | 0.8800 | 0.8802 |

After evaluating the accuracy of our machine learning models, we opted to use a loss curve in order to understand the learning performance of the models overtime. From our evaluation, we have found that the model ResNet50 has the best learning curve among the other pre-trained models and the custom CNN models, that is because we can see that the training loss curve is decreasing steadily while maintaining a consistent flow as well as the validation loss curve having stable fluctuation along with the training loss curve. For this case, we can say that the learning curve for ResNet50 is more steady having lesser fluctuation level in both validation and training loss curve.

On the other hand, for the custom CNN model, we observe that the learning curve has minor complications regarding the validation loss curve. In the learning curve, we can observe that even though the training loss curve is decreasing steadily towards the end, the validation loss curve slightly increases towards the end before stabilizing. It is within our observation that the validation loss curve despite remaining stable has some minor fluctuations as it starts increasing at one point while the training loss curve is decreasing. So we can say it is a better fit rather than a best fit. The following figure 3, 4, and 5 shows the loss curve for the machine learning models.

Sobi:

Furthermore, we applied Explainable AI Techniques in order to visualize the key features that influence the model interpretability. We took a sample image from our dataset and then we applied Grad-CAM , Grad-CAM++ and EIGEN-CAM on the best model which was ResNet-50 for better interpretability on our working dataset. Moreover, it was very evident that among the explainable AI techniques, EIGEN-CAM performed better compared to the others as it was perfectly able to identify the features and seamlessly visualize all the gradients of our image.Figure… shows the effective heatmap for all the Explainable AI techniques  
Afterwards, we also applied LIME(Local Interpretable Model-Agnostic Explanations) for an in-depth visualization of the sample image . In this case ,we took an instance for our sample dataset and LIME effectively was able to identify all our working features of the image. This provided an approximate graphical representation of all the features of our image that was used for our models explainability. In figure… we can observe a graphical visualization of lime.

**Conclusion:**

In our project, we successfully classified seasonal vegetables using some pre-trained machine learning models such as Resnet50, VGG16, DenseNet121, EfficientNetB0, and MobileNet-V2. Among these, Resnet50 demonstrates the best performance and accuracy of 99.33% showcasing its superior compatibility in recognizing patterns and classification within the dataset. Furthermore, we applied XAI such as Grad-CAM, Eigen-CAM, Grad-CAM++ to enhance the interpretability of the models and provide a visual and feature-based explanation to enable a better understanding of the models' decision-making. It is important to note that after applying all explainable AI techniques we acquired a strong interpretation on what features our models used for its decision making and predictions.This approach not only supports the automation of precision agriculture but also paved the way for further explainable and trustworthy AI applications in the agricultural domain.