Pest Detection System Using Machine Learning Techniques

by

Protyusha Barua Prithvi
21241069
Fabliha Zahin
21241068
Sanjida Sultana Anny
18101131

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Protyusha Barua Prithvi
21241069

Fabliha Zahin
21241068

Sanjida Sultana Anny
18101131

Approval

The thesis titled "Pest Detection using Support Vector Machine and Transfer Learning Techniques" submitted by

1. Protyusha Barua Prithvi (21241069)
2. Fabliha Zahin (21241069)
3. Sanjida Sultana Anny (18101131)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 16, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Countries like Bangladesh yield a significant portion of their economy from their agricultural sector. Agricultural pests, on the other hand, have a significant impact on both agricultural production and crop storage. The pest category must be precisely identified, and specific management actions must be adopted as a prevention technique against these pests. As a result, a computer vision-based agricultural pest recognition system must be developed. The implications of certain prospective machine learning algorithms, like Support Vector Machine, Inceptionv3, and Xception, are discussed in this research to achieve insect detection with the complicated agriculture setting. In this study, the dataset used are images of mainly 5 common pests found in a paddy field in Bangladesh. The results achieved from the models were studied based on their accuracy and loss percentage to determine the better approach for such detection to take necessary actions. In this research, SVM outperformed both InceptionV3 and Xception with an accuracy of about 72.5%.

Keywords: Machine learning, Deep learning, Transfer learning, Pest detection, Data augmentation, Loss function, Hyperparameter tuning, Support Vector Machine (SVM), Inceptionv3, Xception, You Only Look Once version 5 (YOLOv5), Convolutional Neural Network (CNN).

Dedication

This thesis paper is dedicated to all the farmers whose hard work brings food to our tables.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, we express our immense gratitude to our supervisor Dr. Amitabha Chakrabarty sir for his kind support and advice in our work. He guided us through every step throughout. And we are also thankful to Shahriar Hossain bhaiya who helped clear out any confusions we had.

And, finally, we thank our parents without whose support it may not have been possible for us to pull through. Because of their constant support and prayers, we are now on the verge of graduating.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CM Confusion Matrix

CNN Convolutional Neural Network

DL Deep Learning

FN False Negative

FP False Positive

ML Machine Learning

SVM Support Vector Machine

TN True Negative

TP True Positive

$YOLO$ You Only Look Once

Chapter 1

Introduction

When we think about agriculture and farming, we always want to obtain maximum yield and produce fresh products all throughout the year. But most of the time, it does not result in so. Pests have always been a nuisance for farmers and play a crucial part in farming.

Diseases and pests have been found to affect the yields of five vital food crops by 10% to 40% on a global scale in recent years, according to a research published by a University of California, Agriculture and Natural Resources scientist as well as some more representatives of the International Society for Plant Pathology. According to a research published in the journal Nature, Ecology Evolution, pathogens and pests cause 10% to 28% losses in wheat, 25% to 41% losses in rice, 20% to 41% losses in maize, 8% to 21% losses in potato, and 11% to 32% losses in soybeans on a global scale. [9]

There have been studies thus far to introduce new and latest models to develop a reliable pest detection system through machine learning and deep learning. Convolutional neural networks, be it as Faster R-CNN or Inception V3, is mostly used while taking an approach to build up the said system.

1.1 Problem Statement

Pests are a problem regardless of the type of crop they have infested upon. In order to eradicate pest infestation from its roots, a proper detection method must be introduced. Thanks to emerging technology, it has been easier in current times to detect pests over a region of crops. Machine learning techniques are ever evolving, so researchers are coming up with different approaches to modify and enhance the detection system by implementing different and advanced models. Owing to developments in hardware technology, deep learning algorithms are being used to handle complex problems in an acceptable amount of time.

However, the fundamental drawback of the deep learning approach is that it is a "black box," making it impossible to comprehend why a deep learning-based algorithm generates a particular prediction. Moreover, deep learning approaches necessitate a larger amount of data in order to provide successful results in the detection of pests. This is a disadvantage because most currently accessible datasets are tiny and lack sufficient pictures, which are required for high-quality choices [7].

1.2 Research Objectives

This research aims to study a model better suited to be used in a pest recognition system to be implemented over a dataset of paddy from Support Vector Machine (SVM), Inception V3 and Xception. The raw dataset consists of images of five different types of pests that are to undergo image processing in order to provide a pre-processed dataset for training and testing the model. The objectives of this research are as follows:

1.To recognise and classify pests from images.

2.To deeply understand the models to be implemented- SVM, Inception V3 and Xception.

3.To train each said model by utilizing labels and features of the given dataset.

4.To evaluate the performance of each model.

5.To offer recommendations on improving the proposed model.

Chapter 2

Literature Review

In a research paper, 10 categories of pests were used to train a transfer learning model called AlexNet and the accuracy was about 93.84% which is comparatively higher than most human specialists specializing in detecting pests and popular neural network algorithms. To find out the validation of the model, 2 other kinds of weeds were used to train it and that gave an accuracy of 98.92% [8]

Several researchers discussed numerous image processing strategies, including median filtering for noise removal, detection by scanning, image extraction, and image filtering, however more accurate techniques are needed. It is difficult to classify insects with high accuracy in big agricultural field crops due to shadow, leaves, mud, branches, and flower buds, among other factors. [25]

A research was carried out to learn about the performance of different machine learning techniques at detecting pests. The models included in this research were ANN, KNN, SVM, BNN and CNN. 91.5% and 90% were the highest accuracies found from the research and both were of the CNN model when they were tested on 9 and 24 insect classes, respectively. [19]

In a specific study, [17] the Faster-RCNN model is used to detect insect targets in order to overcome detection accuracy and processing time limitations. Due to the small size of the acquired insect image samples, the original FasterRCNN model was updated in two ways. To address the issue of the small size of the insect image, the Faster-RCNN model's basic network VGG16 is replaced with a depth residual network (ResNet50), which increases the layer number and decreases the parameters of the basic network, allowing for the extraction of more abundant features. OHEM is used to address the imbalance between the target frame and the background frame in target detection. According to the final trial results, the improved Faster-RCNN model has an average accuracy of 89.64%, which is 4.31% higher than the previous model.

In another paper, [18] the precise identification of a 24-class pest dataset was done by the AF-RCNN which stands for Anchor Free Region Convolutional Neural Network. To start off, a feature fusion tool was used to extract relevant information about the pests. The researchers next provide the proposed model based on the fusion feature maps for producing precise object proposals as probable pest sites.

Finally the proposed model is used to identify the different classes of pests by integrating AFRPN and Fast R-CNN into a single network. They checked the model's performance by testing it out with a dataset containing 20,000 images and 24 types of pests. The results were an 56.44may be as little as 0.07s. The technique used here is reliable and adaptable for accurate and fast pest identification. [18]

The authors of a study paper [21] employed Faster R-CNN and Mask R-CNN to discover pests and diseases that affect the growth of sweet peppers. Their data collection includes 1239 images of sweet pepper strains produced in greenhouses in Chiayi City, Taiwan. The faster RCNN employs a CNN-based region proposal network (RPN). The RPN takes the feature map from the first CNN as input and outputs a bounding box and the probability that the bounding box contains an object. The RPN can be used to deduce the most likely boundary boxes. Despite its imprecision, region of interest (RoI) pooling can be used to evaluate these bounding boxes. Following RoI pooling, each region may be quickly categorized to determine the most precise bounding box coordinates. Mask R-CNN augments the Faster R-CNN network framework with a mask branch to detect the category of each box in the images. Segmentation is accomplished via full convolution in a fully convolutional network (FCN). As a result, the task of object detection is transformed into classification, regression, and segmentation in Mask R-CNNs. As a result, Mask R-CNN is a combination of Faster R-CNN and FCN. Mask R-CNN can be used to determine the target's pixel-by-pixel location within the category of instance segmentation models. Regional recognition accuracy is 89% for the Faster R-CNN, whereas area recognition accuracy is 81% for the Mask R-CNN.

In a study conducted by a group of researchers [31], they used the YOLO (You Look Only Once) technique for pest detection and the AlexNet CNN algorithm for pest classification to detect and classify pests in images. To improve the accuracy of pest detection and categorization in images, two networks were deployed. They employed a variety of pests in this manner, including the Colorado Beetle, the Grasshopper, the Japanese Beetle, and the Ladybug. These pests are found on plants' leaves and flowers. Images of pests were taken from a variety of perspectives and kept in the database. According to their study, the YOLO Based Segmentation technique offers accurate pest detection by deleting the most undesirable area from the input image.

A researcher named Vijayakumar compares YOLOv3 to CNN in another research [27]. CNN, he believes, has a complex design that demands multiple stages of processing, but with Yolo v3 and Darknet 53, these concerns are resolved and performance is increased. Achyut Morbekar [24] is a crop pathologist. His paper, "Crop Disease Detection Using YOLO" explains the operation of YOLOv3, including the bounding box and class probability. The YOLO algorithm may be trained to be more accurate than other algorithms at detecting diseases.

Another group of researchers collaborated on a publication [36] titled "Precision Agriculture" through the use of a YOLO-based pest detection system. They stated that early detection of pests is crucial for the development of successful crop defense measures in Precision Agriculture (PA) settings. They focused on true bugs as potential pests because they can significantly impair production of hazelnut. They achieved an average precision of 94.5% on a custom dataset in outdoor situation and training a YOLO-based (CNN). They conducted a detailed performance evaluation of the detector. They used a data-driven approach based on CNN to identify pest infestation and differentiate them from other things caught in the trap. They specifically employed the YOLO framework [22], which enables real-time processing. This enables the detection process to be completely automated and carried out onboard a mobile robot capable of navigating the field [15], inspecting the traps, and treating the plant appropriately. They developed a pest detection system capable of detecting genuine bugs on sticky traps in a field by training it on a unique dataset acquired in real-world outdoor situations. They were able to test the approach's success by undertaking a rigorous assessment of various data augmentations, reaching an average precision of 94.5 percent on 611 fresh photographs.

Convolutional Neural Networks (CNNs) serve as the foundation for most of the existing object detection methods. There are two primary CNN-based techniques for object detection [34]. At first, a 2-step design where a region proposal method is used to select potential regions in the image input, followed by a classification problem in these regions; and secondly, a 1-stage architecture in which only one detection iteration is used. The first group includes R-CNNs and Faster R-CNNs,and the other includes SSDs and YOLO. Two-stage detectors are more precise than single-stage detectors but they frequently lack real-time processing capabilities. At the time of writing, YOLO is the quickest choice accessible in this regard. [34]

A work proposes a classification and recognition technique for four prominent southern vegetable pests like Whiteflies, Phyllotreta Striolata, Plutella Xylostella, and Thrips, using a BOF-SVM. This work contains four sub-algorithms. The first one computes the character details of pest photos using scale-invariant feature transformation. In the second one, the visual vocabulary is generated using a bag of features. The third one uses support vector machines to calculate the pest classifier. The final step is to identify the pest photos using the classifier. Experiments found that when 80 images from the real world were used to judge a single image category, the accuracy was about 91.56% and it took about 0.39 seconds. This technique operates with the desired speed and precision. Even when the test image has a complex environmental background, such as size, angle, and so on, the accuracy rate of categorization is frequently greater than 90%. Classification of the entire test set takes approximately 30 seconds, with each image taking an average of 0.4 seconds. [6]

The primary objective of this other research is to detect thrips on crop canopy photos using an SVM classification approach. The researchers utilized a novel image processing technique to detect parasites in strawberry plants. The SVM technique with varied kernel functions was used to classify parasites and identify thrips. The SVM structure was created utilizing the main diameter to minor diameter ratio as a region index and the hue, saturation, and intensify color indices. Additionally, mean square error (MSE), the root of mean square error (RMSE), mean absolute error (MAE), and mean percentage error were used to evaluate the categorisation (MPE). The results indicate that using the SVM approach with region index and intensification as color index results in the most accurate classification, with a mean percent error of less than 2.25%. [3]

Another study was conducted with the primary objective of recognizing and quantifying pest-affected areas in leaf pictures. Segmentation of images was used to detect the presence of pests in leaf photographs. The performance of an image segmentation algorithm is defined by how well it simplifies images. The K-means cluster algorithm was used to accurately identify whiteflies, aphids, and thrips in different leaf pictures. The polluted area was determined using a 98.4% precise SVM classifier. [2]

The researchers also proposed an automatic pest identification system based on image processing techniques in another publication. The color feature is utilized to train the SVM to categorize pest pixels and leaf pixels. The algorithm examined one hundred photographs and determined that 95 of them were accurate. The results indicated that the pest identification technique was accurate in identifying the pests. In most cases, the approach produces accurate findings in a reasonable length of time. [1]

Traditional approaches to recognition and deep learning, such as KNN (k-nearest neighbors) and AlexNet, are not preferred by competent researchers due to their shown ineffectiveness. In a recent study, researchers classified ten citrus pests using four different types of advanced deep learning frameworks. Inception-ResNet-V3 achieved the lowest classification error, the highest classification accuracy (98.73%), and the second quickest time per epoch in the experiments (58s). [5]

7 pretrained models VGG-16, VGG-19, ResNet-50, InceptionV3, Xception, MobileNet, and SqueezeNet were tweaked and retrained on the D0 dataset with 40 classes in a paper using appropriate transfer learning and 7 fine-tuning approaches. Later, to improve performance of the models, the best among these, Inception-V3, Xception, and MobileNet, were then ensembled using sum of maximum probabilities technique and named SMPEnsemble.Using the validation dataset, the genetic technique was utilized to determine model weights. According to the test results, the GAEnsemble approach achieved a classification accuracy of 98.81, 95.15, and 67.13 percentages for the D0, SMALL, and IP102 datasets, respectively. On all three datasets, the suggested approach achieves higher accuracy rates than all individual CNN models. [14]

Another study established a detection technique for the early identification of agricultural diseases. Many transfer learning models were evaluated in order to ascertain which would help to make a more precise detection model. The models used in this research were the CNN, VGG16, InceptionV3, and Xception. The latter 3 are pretrained models based on CNN architecture. The Xception model outperformed all other models/architectures on the pest dataset. It scored 77.90% accurately.Inceptionv3 scored 77.19%, becoming the second best model. The remaining networks, VGG16 and CNN, scored 71.74% and 24.28%, respectively, on the test. [32]

A new model for identifying plant diseases based on leaf image classification has been developed. It has a DCGAN and a classifier discovered using MLP neural networks trained with the PILAE algorithm. The DCGAN does two tasks: minor class image synthesis to correct for the dataset's imbalance, and deep feature extraction from all of the dataset's images. The PILAE classifier is capable of delivering outstanding training efficiency and consistency. The PlantVillage dataset's empirical results indicate that the proposed technique works well by giving positive results with different models while remaining very basic. [23]

Another study describes an approach for detecting pests that is based on transfer learning. Nine different insect species are collected for classification and identification. These insects include the primary pests and some natural foes of the field's major food crops, including wheat, rice, and maize. The researchers next expand the insect dataset and develop a model based on transfer learning to apply the knowledge obtained on the ImageNet dataset by VGG16, VGG19, InceptionV3, and InceptionV4 for identification. According to results, the transfer learning training model has better identification performance and data expansion can assist in expanding the sample size and avoiding overfitting. Models that pretrain the model for transfer learning using the VGG19 convolutional neural network get accuracy of 97.39%. This method has a high recognition accuracy, is fast, simple to use, and is robust to translation and rotation, making it an excellent choice for field insect identification and categorization. [16]

In this article, DCNN (deep convolutional neural networks) are utilized to identify 10 different pest species in the rice field. Although the dataset has 3549 of images of pest affecting rice crops, the data augmentation process is performed since Deep Learning works best with larger data sets. Multiple forms of DCNN architecture were used to generate the neural model, and the models were interpreted based on their output and correctness. The transfer learning technique is used by hyperparameters fine-tuning and ResNet-50 model layers. The ResNet-50 model being fine-tuned relatively outperformed other models by 95.012%. The resulting resultant value indicates the model's effectiveness in classifying pest diseases. [35]

Another study's primary objective is to develop the best ML algorithm for detecting tomato plant illnesses in coloured images. In order to address this issue, the researchers study DensNet161, DensNet121, and VGG16. For their investigation, images of diseased plant leaves were classified into six categories. DensNet161 was 95.65%, DensNet121 was 94.93%, and VGG16 was 90.58% accurate. DensNet161 outperforms the other designs in terms of test accuracy with 20 training epochs. It is reasonable to conclude from the research that DensNet has an architecture that is well-suited for the task of identifying plant diseases based on crop pictures. Additionally, they discovered that DensNets produced superior outcomes with fewer parameters. [26]

They developed a pest recognition model using 71 types of 35,000 photos of bugs in a study, leveraging the Inception-v3 and Inception-v4 models in GoogLeNet. Finally, they combined existing methods for comparing and validating effects using several evaluation markers. The results indicate that the deep learning model outperforms all other models in this study when it comes to identifying pests and diseases. The

Inception-v4 support vector machine classification approach has the highest accuracy, with a classification rate of 97.3 percent, followed by Inception-v3. The results indicate that Inception-v4 is the most accurate model of the seven. With an accuracy of only 89.26%, SVM is the least accurate. The Inception-v4 model mentioned in this work has the highest precision, at 97.3%. The Inception-v4 model's quality and utility for pest identification have been established beyond a reasonable doubt. [10]

The purpose of another research is to discover pests in maize crops among 4320 images of six kinds of pests. It offers a new dataset of photographs of pests for supervised classification, which contains both original and edited images. Additionally, it introduces Inception-V3*, a variant of the Inception-V3 residual deep learning model that enables quicker learning and more accurate than that of the actual model. One test was conducted for significant insects and another for rest of the other kinds. Residual models of Inception-V3 and its altered version Inception-V3* along with AlexNet are run using transfer learning, as are pre-trained weights from ImageNet. When cross-validation was used, Inception-V3* model was 97% accurate for all pest classes. Standard Inception-V3 scored 94.8% and AlexNet scored 96.3%, on the other hand. [11]

ResNet34, ResNet50, and Inception-V3 all achieved about 88% accuracy, which a group of academics believes can provide farmers with a competitive edge over other traditional models. Meanwhile, they learn that prediction speed is not always proportional to the number of parameters. For example, AlexNet has three times the number of parameters as ResNet34 yet runs in around a sixth of the time. They assume that this phenomenon is caused by the higher layer count in neural networks, which results in greater forward propagation steps and, thus, increased calculation time with fewer parameters. While VGG-11 and VGG-19 have a greater number of parameters, their performance is not as good. They believe that as the dataset increases in size, the models will improve in performance. Finally, they confirm that, based on the preceding experimental results, Inception-V3 is capable of performing better on this targeted problem. [29]

The authors conduct a review and experiments on seven different transfer learning models, including VGG16, ResNet50, MobileNet, DenseNet, InceptionV3, Xception, and InceptionResNet, and compare their accuracy, precision, F1 Score, and training time, with the goal of developing a future effective and efficient automated classification system. Additionally, another researcher's CNN model is being trained and evaluated. The studies analyzed Fruit 360, a 120-class dataset. In the initial part of this research, models were trained on a subset of the dataset encompassing 21 classes. VGG16 and ResNet50 are the top two models. As a result, the last phase of the experiment focuses exclusively on these two models throughout the entire 120-class dataset. Overall, the results indicate that VGG16 is the most accurate model, with a training accuracy of 99% and a testing accuracy of 95%. [28]

There is a scarcity of available samples of agrarian plant illnesses and pests. Own curated data sets are far less in comparison to open standard libraries. Even with over 14 million images in ImageNet datasets, the issue of small samples is the most critical constraint for plant disease and pest diagnosis. In reality, as plant disease are rather hard to find while the expense for the search is relatively high, it leads to DL methods being useless in such recognition. [33]

The authors of an article suggest a transfer learning-based pest detection and recognition diagnostic system 10. This technique is capable of training and testing ten distinct pests and has a 93.84% accuracy rate. This transfer learning approach was compared to that of human experts and a standard neural network model. This experiment uses fewer than 500 photographs and the model's recognition accuracy is 93.84%. This dataset is fairly little in comparison to a conventional neural network model. Indeed, Ferentinos18 trained a CNN model using 87 848 images of plant illness, achieving the highest performance of 99.53%. As a result, the accuracy increased by only 5.7% even with over 82 000 photographs. As a result, it appears as though transfer learning is a superior model to CNN. When retraining layers for different pest categories, it is possible to build a more accurate model by transferring pre-trained models that have been trained on millions of pictures. As a result, a huge number of samples is not required to provide a suitable result. [8]

With an accuracy of 87%, the created model was able to classify 13 different types of paddy pests and diseases in a study. This precision, the researchers feel, demonstrates the feasibility of an automatic classification system for paddy pests and diseases. They acquired 4,511 photographs in four languages using search engines and complemented them to build a diverse data collection. The CaffeNet model was fed this dataset, which was subsequently processed using the Caffe framework. In the trial, the model achieved an accuracy of 87%, which is greater than the 7.6% gained from random selection. [4]

Using eight pre-trained deep learning models, the authors extract deep characteristics from photos in another paper (VGG16, VGG19, Resnet50, and so on). They conducted testing on 28,011 photos of 34 distinct illnesses and pests in hot pepper. However, the accuracy rate obtained were near to the one obtained from k-nearest neighbor technique. They achieved sickness detection accuracies of approximately 88.38–93.88% and pest recognition accuracies of approximately 95.38–98.42%. They obtained the second and third best results, respectively, by leveraging features that were fixed out from the VGG16 and VGG19 models. They were able to achieve 85.6% for illnesses and 93.62% for pests by utilizing the ones from the Resnet50. When compared the proposed technique beat the CNN model by 8.62% in diseases and 14.86% in pests. [30]

A study employs a deep CNN-based system to address the classification problem of identifying plant pests. Researchers analyzed a dataset from Turkey that had photographs of actual plant diseases and pests. To begin, they used this dataset to extract deep features from the transfer learning models AlexNet, VGG16, VGG19, GoogleNet, ResNet50, ResNet101, InceptionV3, InceptionResNetV2, and SqueezeNet. Classification of the deep features discovered in these deep models was performed using SVM, ELM, and KNN. They then employed transfer learning-based deep learning models to fine-tune the results. To address the issue, they deleted the architecture's final three tiers and replaced them with new layers that utilized pre-trained CNNs. Finally, they assessed the performance data using transfer learning and deep feature extraction techniques. As a result, the ResNet50 model combined with the SVM classifier attained the highest degree of accuracy, 97.86%. [13]

The research proposes a crop pest recognition system that employs numerous deep convolutional neural networks to classify ten common crop pest species accurately (CNNs). (1) A manually collected and validated crop pest dataset is described and shared; (2) a fine-tuned GoogLeNet model is proposed to deal with the complicated backgrounds of farmland scenes, with pest classification results superior to the original model; and (3) the fine-tuned GoogLeNet model achieves a 6.22% improvement over the state-of-the-art method. Thus the proposed model has the potential to be used in real-world situations and to enhance further crop disease research. [20]

Researchers created an excellent deep CNN model for recognizing types of insects using 3 publically available datasets. The very first dataset used was the National Bureau of Agricultural Insect Resources (NBAIR), which contains 40 different types of field crop insects, while the second and third datasets (Xie1, Xie2) contain images of 24 and 40 different types of insects, respectively. Pre-trained deep learning models such as AlexNet, ResNet, GoogLeNet, and VGGNet were used to evaluate, check and compare the results of the proposed model with the insect categorization. Transfer learning was used to fine-tune the pre-trained models. Data augmentation techniques like translation, reflection, rotation and scaling help to reduce overfitting. To improve accuracy, the suggested model's hyperparameter efficacy was studied. The suggested CNN model achieved the greatest classification accuracy of 96.75%, 97.47%, and 95.97%, respectively, for the NBAIR insect dataset (40 classes), the Xie1 insect dataset (24 classes), and the Xie2 insect dataset (40 classes). [12]

It can be seen in the table below, many Machine Learning algorithms and tools have emerged and are working very impressively at detecting small objects like pests which is one of the grave problems in agricultural sectors. Hybrid models have performed better in some cases and accuracy was significantly higher with a larger dataset used. Based on our literature review, we have decided to work with SVM to analyse the performance of a Machine Learning algorithm and for studying Transfer Learning, we have chosen InceptionV3 and Xception which have proven to be very able achitectures.

Reference	Model	Dataset	Accuracy
8	AlexNet	types of weeds	98.92
19	CNN	Wang	91.5
17	Faster RCNN*	classes of insects	89.64
18	AF-RCNN	24-categories pest	85.14
21	Faster RCNN	Sweet peppers pest	89
	Mask RCNN		81
36	YOLO	Hazelnut pests	94.5
6	BOF-SVM	Southern vegetable pests	91.56
2	SVM	Leaf pests	98.4
5	Inception-ResNet-V3	Citrus pests	98.73
14	Inception-V3, Xception, and MobileNet	D0	98.81
		SMALL	95.15
		IP102	67.13
32	Xception	Tomato pests	77.9
	Inceptionv3		77.19
	VGG16		71.74
	CNN		24.28
16	VGG19	Kinds of insects	97.39
35	DNCC	Rice crop pests	95.012
26	DensNet161	Tomato crop	95.65
	DensNet121		94.93
	VGG16		90.58
10	Inceptionv4	71 types of pests	97.3
	SVM		89.26
11	Inceptionv3*	Pests in maize	97
	Inceptionv3		94.8
	AlexNet		96.3
29	ResNet34	Plant and Leaf diseases	88
	ResNet50		88
	Inception-V3		88
28	VGG16	Fruit 360	99
	ResNet50		95
4	CaffeNet	Paddy pests	87
30	ResNet50	Hot pepper diseases	95.38–98.42
	VGG16		85.6
	VGG19		93.62
	CNN		14.86
13	ResNet50-SVM	Turkey Pest	97.86
12	CNN*	NBAIR	96.75
		Xie1	97.47
		Xie2	95.97

Figure 2.1: *Summary of Related Works*

Chapter 3

Methodology

The purpose of this study is to demonstrate how different models perform on the same dataset. Thus, a selected dataset had to be preprocessed and trained accordingly to each implemented model of SVM, Inceptionv3, and Xception. We collected about 1500 images of five types of pests that the paddy crops are commonly infested with. Following the appropriate preprocessing and augmentation, each model was trained with the clean training dataset and tested on with a separate clean testing dataset, and the results were then examined to determine their performance.

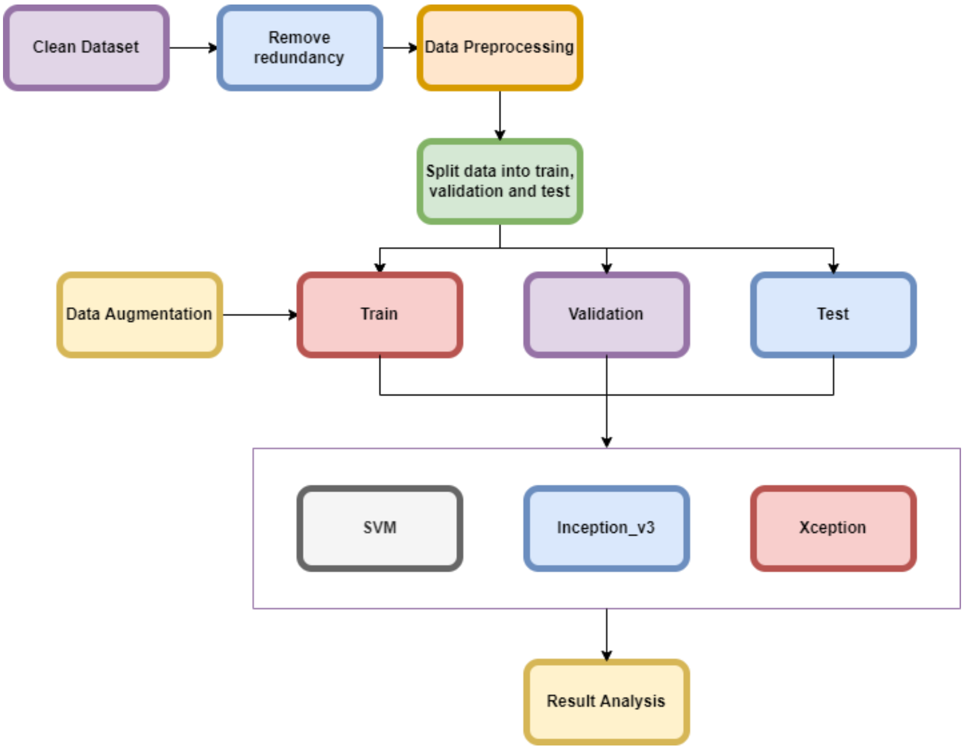

Figure 3.1: *Workflow Diagram*

3.1 Support Vector Machine

Small patterns in large datasets can be detected using SVM (Support Vector Machine), which is a reinforcement technique. Vapnik was the first to propose it, back in 1995. Essentially, it is a linear model that can be used to tackle challenges related to classification and regression. It is capable of dealing with both linear and nonlinear problems, and it is suitable for a wide range of application areas.

SVM separates data into two classes by drawing a line through it, which is the main idea of the approach. It is, thus, generally used for binary classification. However, over recent years, it has been possible to get multi-class classification by implementing numerous binary SVMs together. There are a variety of approaches that may be used to complete this task, but the ultimate goal is to identify the line that has the greatest distances between the classes so that if a new data point needs to be categorized, later on, it can be done fast and efficiently.

3.1.1 Mechanism

SVM modeling has two stages: first, the dataset is trained and a model is plotted, and then the model is used for prediction from a test data set.

One of the goals of the SVM method is to determine the best fit line or decision boundary for sorting n-dimensional space into specific classes in order to make it easier to place more data points in the proper section later on. It can be illustrated by Figure 3.2.

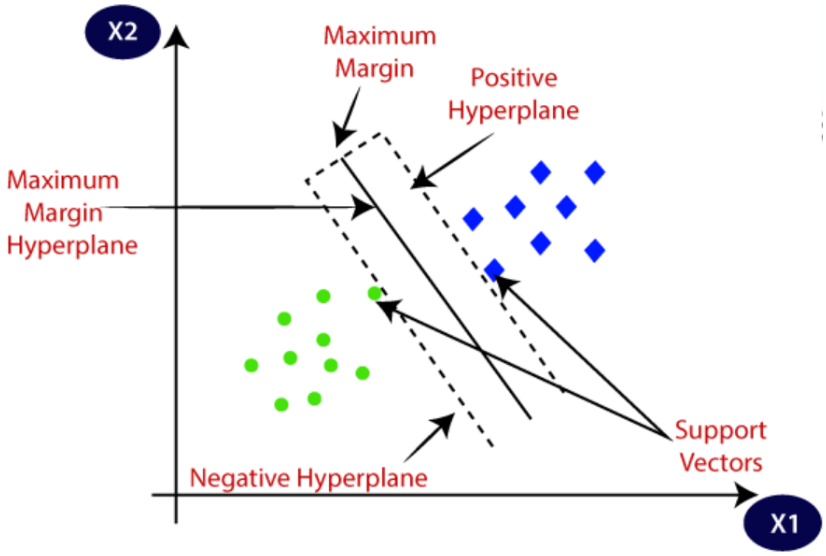

Figure 3.2: *Working principle of SVM*

Here, in the binary classification, the hyperplane is the optimal choice boundary. It gets its dimensions based on the number of features in the provided dataset. Since, there are two features in Figure 3.2, the hyperplane is a straight line. Had there been three of them, then we would have gotten a 2-dimensional line as the hyperplane and so on.

The hyperplane is generally created in such a way so that the maximum margin is obtained. The highest distance between the two data points is known as the margin.

Lastly, the extreme points that are at the most nearest positions to the hyperplane are known as the support vectors, which, in fact, help form the hyperplane by determining its position. The entire algorithm is, therefore, named after this.

3.1.2 Implementation

The Support Vector Machine algorithm uses a kernel function to compute the results. To take data as input and transform it into the format required for processing by the model, a kernel function is used. It uses a set of mathematical functions to provide a window through which the data can be changed. Accordingly, the kernel function modifies the training data in a manner that an otherwise nonlinear decision surface can be turned into a linear equation in an increased number of dimension spaces. On the most basic level, it returns the inner product of two points in a standard feature area. As a result, the SVM becomes more effective, adaptive, and exact in its operations.

The standard kernel function equation is:

$$\begin{aligned} &\bar{K}(\bar{x}) = 1, \text{ if } \|\bar{x}\| <= 1 \\ &\bar{K}(\bar{x}) = 0, \text{ Otherwise} \end{aligned} \tag{3.1}$$

The equation for Linear Kernel, which is the dot product of any two observations, is:

$$\mathrm{K(x, xi) = K(x \times xi)} \tag{3.2}$$

The equation for Polynomial Kernel, which is generally used in image processing, is:

$$k\left(\mathbf{x_i}, \mathbf{x_j}\right) = \left(\mathbf{x_i} \cdot \mathbf{x_j} + 1\right)^d \tag{3.3}$$

where, d is the degree of the polynomial

3.2 Inception V3

Convolutional Neural Networks (CNN) are used in the Inception V3 deep learning model for image classification. It's a more upgraded version of Inception V1's core model, which was constructed by a Google team and first released as GoogleNet in 2014. To transmit label information deeper down the network, it employs Label Smoothing, Factorized 7 x 7 convolutions, and an auxiliary classifier.

3.2.1 Mechanism

The Inception V3 model has 42 layers in total and a decreased error rate than previous models. The model includes symmetric and asymmetric foundation which consists of convolutions, concatenations, max pooling, dropouts, average pooling and fully linked layers. The activation inputs are batch normalized, which is used extensively throughout the model. In addition, the loss is calculated using Softmax, which is an optimizer for the classification.

The extensive dimension reduction was one of the main advantages of the Inception V1 model. The model's larger convolutions were factored into smaller convolutions to increase its effectiveness. It has a 5x5 convolutional layer, which is expensive to compute. To save time and cost, the 5x5 convolutional layer was replaced with two 3x3 convolutional layers. The computational expenses are lowered in turn due to the reduced number of parameters and a quantitative gain of 28% was achieved by factoring bigger convolutions into smaller convolutions.

Despite having larger convolutions factored into smaller ones, asymmetric convolutions i.e. convolutions in the form nx1, were a superior option for making the model more efficient. Thus, a 1x3 convolution followed by a 3x1 convolution is used in lieu of the 3x3 convolutions. It is the equivalent of putting in a two-layer network having an identical receptive field as a 3x3 convolution. If the number of input and output filters is kept equal, then the two-layer approach is 33% less expensive for the same number of output filters.

Using an auxiliary classifier to enhance the convergence of very deep neural networks is the goal. It is primarily employed to tackle the vanishing gradient problem. In the early stages of the training, the auxiliary classifiers made no difference. However, in the end, the network with auxiliary classifiers outperformed the network without them in terms of accuracy. As a result, in the Inception V3 model architecture, the auxiliary classifiers operate as regularizer.

To minimize the grid size of feature maps, max pooling and average pooling were conventionally utilized. But in the inception V3 model, the activation dimension of the network filters is enhanced in order to lower the grid size substantially.

A depiction of the final model for Inception v3 is as follows in Figure 3.3.

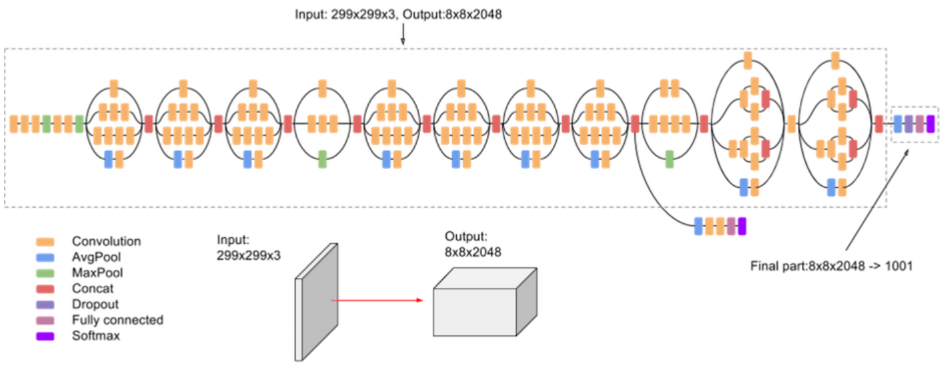

Figure 3.3: *Architecture of Inception V3*

It is evident that the Inception V3 model outperforms V1 and V2 even by having a cumulative number of layers of 42. The figure below represents the detailed components of the model.

TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Conv	3×3/2	299×299×3
Conv	3×3/1	149×149×32
Conv padded	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Conv	3×3/1	73×73×64
Conv	3×3/2	71×71×80
Conv	3×3/1	35×35×192
3 × Inception	Module 1	35×35×288
5 × Inception	Module 2	17×17×768
2 × Inception	Module 3	8×8×1280
Pool	8 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
Softmax	Classifier	1 × 1 × 1000

Figure 3.4: *Components of Inception V3*

3.3 Xception

Google describes inception modules in CNN as a base among both standard convolution and the depth-wise separable convolution operation. Given that, the latter can be compared to an Inception module with the greatest number of towers. As a result of this finding, they proposed Xception, a novel deep CNN architecture based on Inception, except, using depth-wise separable convolutions instead of Inception modules.

3.3.1 Mechanism

"Xception" is an abbreviation for "extreme Inception", as it is an extreme version of Inception. To shrink the original input in Inception, 1x1 convolutions were employed, and different kinds of filters were used to each of the depth spaces dependent on the input spaces. Xception, on the other hand, flips the action. Rather, the filters are applied to each depth map separately before the input space is compressed using 1x1 convolution across the depth. This approach is virtually identical to a depth-wise separable convolution, that has been utilized since 2014. The apparent lack of a non-linearity following the first process is another contrast between Inception and Xception. In the Inception model, both operations are followed by a ReLU non-linearity, whereas the Xception model does not incorporate any nonlinearity. The figure below shows the working principle of Xception.

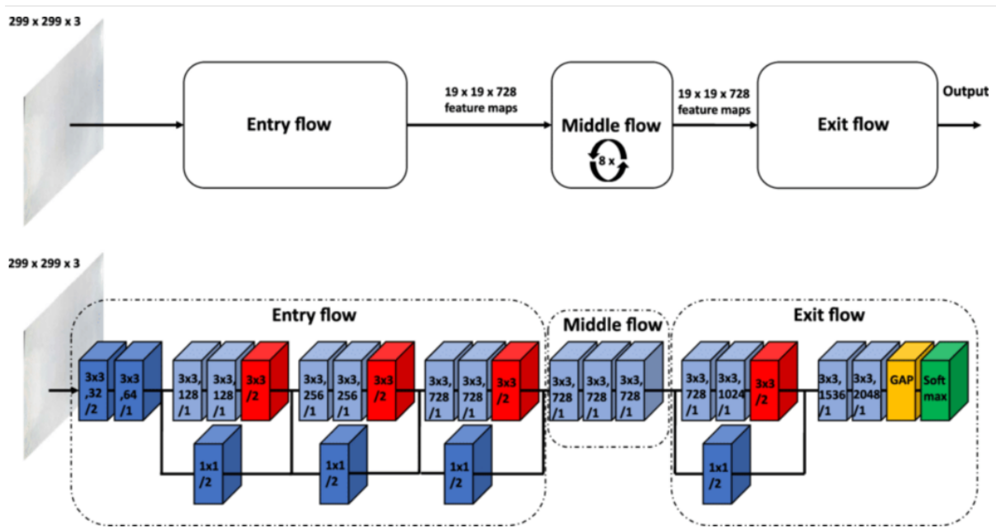

Figure 3.5: *Architecture of Xception*

Chapter 4

Dataset and Experimentation

4.1 Dataset

Our raw dataset consists of a total of 1500 images of pests. While going forth with our study with respect to Bangladesh, we maintained our dataset abiding by having five common pest types that infest the paddy crops, namely Brown Plant Hopper ("Badami Gaach Foring"), Rice Gall Midge("Golmacchi"), Rice Leaf Hopper("Paata Foring"), Rice Leaf Roller("Paata Morano Poka") and Stem Borer("Maajra Poka"). The images consist of 256 × 256 pixels in the RGB color space. This dataset is obtained from the website www.kaggle.com which is a source of thousands of reliable datasets. While this dataset is freely available, it is not primarily not processed properly and is present in its raw state. It, therefore, requires cleaning and processing in order to use it in the implementation of a model for pest detection research. Due to our limitation in obtaining a cleaner dataset, we opted to cleaning and pruning the images manually, removing any unusable and unwanted aspects from the images before commencing data preprocessing. The dataset was then labelled as five different categories for the five insects to be trained on. Then the entire dataset was divided into "Training" and "Testing" for SVM and into "Training", "Validation", and "Testing" for Inception V3 and Xception.

The raw dataset can be downloaded from: https://www.kaggle.com/kevinyawned/25insects

Name	**Data Size**
Brown Plant Hopper	300
Rice Gall Midge	300
Rice Leaf Hopper	300
Rice Leaf Roller	300
Stem Borer	300

Table 4.1: Labels of Dataset

4.2 Data Pre-Processing and Manipulation

As part of data preprocessing, we implemented the process of normalization. Data re-scaling is the process of projecting picture data pixels to a preset range (usually (0,1) or (-1, 1)) in this way. This is frequently used on a variety of data types, and we normalize them all so that the same procedures may be used for them. This helps to ensure that all images are treated equally. Scaling all photos to the same range of [0,1] or [-1,1], for instance, allows all images to contribute equally to the overall loss. Re-scaling also provides that all images have the same learning rate, as higher resolution images necessitate a low learning rate and vice versa.

We then applied the process of grayscaling on the images that turned the colored images to black and white. It is commonly used in machine learning techniques (used in SVM in this study) to reduce computing complexity. Since many images do not require shades to be identified, grayscale is the ideal option, as it reduces the number of pixels in an image, and hence the number of computations needed.

Another approach for scaling and preparing images so as to keep their heights and widths identical is standardization. It rescales the data to a 0 mean and 1 standard deviation (unit variance). This particular technique can help to enhance data quality and consistency.

Additionally, using SVM required us to implement Local Binary Pattern (LBP). It's a brilliantly simple texture extractor that identifies pixels in an image by thresholding each pixel's surrounding pixels and treating the result as a binary integer. In real-world applications, the LBP operator is resistance to monotonic grayscale changes produced, for example, by lighting variations is a critical feature. Another distinguishing aspect is its computational simplicity, which enables it to assess photos in challenging real-time circumstances.

Furthermore, we implemented HOG (Histogram of Oriented Gradients), which is a feature extraction technique from image data. The HOG descriptor is able to offer the edge direction by focusing on the geometry of an item. The gradient and orientation of the edges are extracted to do this. Each of these zones would get its own histogram from the HOG.

We implemented data augmentation in order to make slight changes to current data (training data) to, in turn, expand its diversity without having to collect new information. It can be said to be a method for increasing the size of a dataset. Horizontal and vertical flipping, rotation, cropping, shearing, and other data augmentation techniques are common. Data augmentation prevents a neural network from learning features that are not useful. As a result, the model's performance improves.

Our augmentation techniques included the ones as follows-

- Shifting - This is the method of moving picture pixels horizontally or vertically.
- Flipping - In either vertical or horizontal scenarios, this method swaps the rows or columns of pixels.
- Altering brightness - This method enhances or diminishes the image contrast.
- Cropping - To generate an excerpt of an image, which is subsequently scaled to the original image's dimensions is called cropping.
- Scaling - Scaling an image inward or outward is possible. When a picture is scaled outward, it becomes more meaningful than when scaled inward, and vice versa.
- Rotation - It is the process of rotating an image by a certain amount.

4.3 Simulator

We ran our experiment and machine learning algorithms in Google Collaboratory, a Python-based Website Integrated Development Environment (IDE). Python 3 was the programming language we utilized. We completed our experiments on the dataset and presented our results in Chapter 5 using all essential libraries and frameworks such as Numpy, Pandas, Matplotlib, Seaborn, Keras, TensorFlow, Sklearn, and others.

4.4 Confusion Matrix

A confusion matrix, or an error matrix, is a graphical representation of an algorithm's functional ability. The following factors play a significant influence in a confusion matrix that is formed as a result of a machine learning algorithm:

- True Positive (TP): When a model predicts a positive result and it turns out to be correct, it is referred to as true positive.
- True Negative (TN): When a model predicts the result to be negative and it turns out to be correct, it is referred to as true negative.
- False Positive (FP): When a model predicts the result to be positive and it turns out to be incorrect, it is referred to as false positive.
- False Negative (FN): When a model predicts the result to be negative and it turns out to be incorrect, it is referred to as false negative.

4.5 Precision, Recall and Accuracy

The key metrics of any pattern recognition technique that aids in the detection of a specific pattern in a given set of data are precision, recall, and accuracy.

- Accuracy: The best metric for comparing the results of a model simulation is accuracy. The ratio of the total number of right predictions divided by the sum of the number of forecasts is accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{4.1}$$

- Precision: Precision is the percentage of relevant examples among the recovered instances in pattern recognition and machine learning classification. Positive predictive value is another name for it. Mathematically it can be depicted as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

- Recall: In pattern recognition, classification, and information retrieval, recall is just as important as precision. It's the proportion of relevant retrieved occurrences that are relevant. It can be written mathematically as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

Chapter 5

Result Analysis and Discussion

5.1 Result Analysis

In order to get the most of each working model, we have manipulated the parameters and learning rates for both the ML and DL techniques and obtained the results.

5.1.1 SVM

For SVM, we altered the parameters by changing the kernel functions used and trained our model under two separate conditions. When the kernel was set to linear, the overall performance scores of the model was relatively lower than those obtained when the kernel was set to polynomial. Table 5.1 illustrates the said results.

Model	Parameter	Precision	F1 Score	Accuracy
SVM	Kernel=Linear	0.693	0.693	69%
SVM	Kernel=Polynomial	0.725	0.725	72.5%

Table 5.1: Manipulation of parameter for SVM

As seen from the results achieved above, it was evident that the results obtained from the polynomial was better so the further results analyzed are all from that of the model trained under polynomial kernel function.

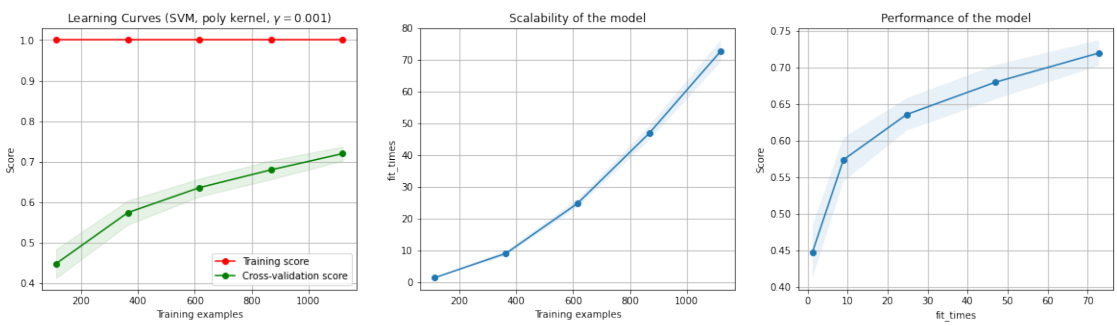

Figure 5.1: *SVM Graphs*

From Figure 5.1 we can see that the accuracy of our model of SVM is 72.5%. Scalability is a feature of a model that describes its capacity to cope with and perform well when the workload or scope is raised or expanded. Since the percentage of scalability is around 72%, we can also conclude that our model can work moderately well on larger dataset. Additionally, it can be depicted based on the overall performance of 71% our SVM model on our implemented dataset, that it is likely to outperform its current threshold if a cleaner dataset of image were to be used.

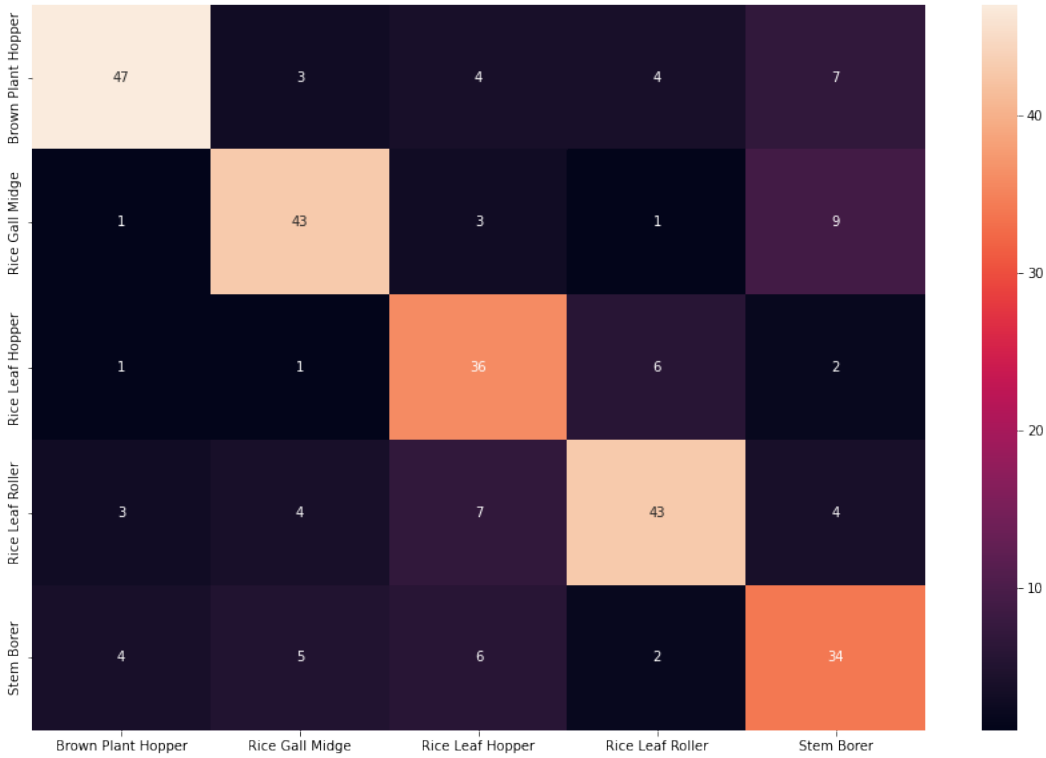

Figure 5.2: *SVM Confusion Matrix*

The confusion matrix has been plotted in order to observe the values obtained along the diagonal which depict the scores of the correctly labelled data points. Here, from Figure 5.2, we can see that our model has successfully detected 47 Brown Plant Hopper, 43 Rice Gall Midge, 36 Rice Leaf Hopper, 43 Rice Leaf Roller and 34 Stem Borer out of the 100 images in the test set for each kind.

	Precision	Recall	F1 score	Support
Brown Plant Hopper	0.75	0.75	0.75	56
Rice Gall Midge	0.73	0.77	0.75	56
Rice Leaf Hopper	0.73	0.77	0.75	56
Rice Leaf Roller	0.77	0.79	0.78	56
Stem Borer	0.65	0.57	0.61	56

Table 5.2: Classification report for SVM

It can be depicted from Table 5.2 that the average accuracy that the model could achieve in identifying the five different classes of pests was around 73%.

5.1.2 Transfer Learning

We have manipulated the learning rates for the Xception Transfer Learning model to figure how it affects the validation loss and accuracy. Lowering the learning rates significantly lowered the validation loss but it took longer for the model to train, whereas when we increased the learning rate, the model trained way faster but the loss increased drastically.

Model	Parameter	Time	Val-loss	Val-Accuracy
Xception	LR=0.0001	7	1.2502	0.843
Xception	LR=0.001	6	1.4502	0.82
Xception	LR=0.5	5	3.7646	0.812

Table 5.3: Manipulation of parameter for Xception

The same approach was implemented on Inception V3 but the results were rather identical to those of Xception's.

Further hyperparameter tunings were performed on the model by changing the dropout layer rate within the range of 0.2 to 0.8. It was seen that the optimum working dropout layer rate for the model was at 0.5 which gave a validation loss of 0.2134 as can be seen in Figure 5.3.

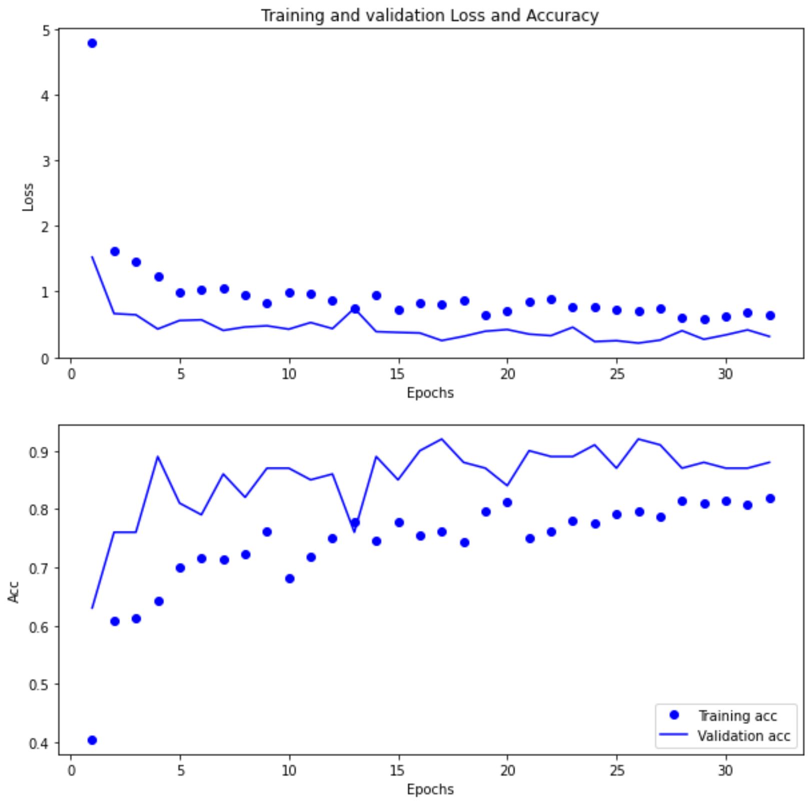

Figure 5.3: *Xception graphs*

Inception V3 model behaved in a similar manner when it was trained, except the dropout layer rate was set to 0.2. Thus, there happened a significant fall in the validation accuracy, as shown in Figure 5.4

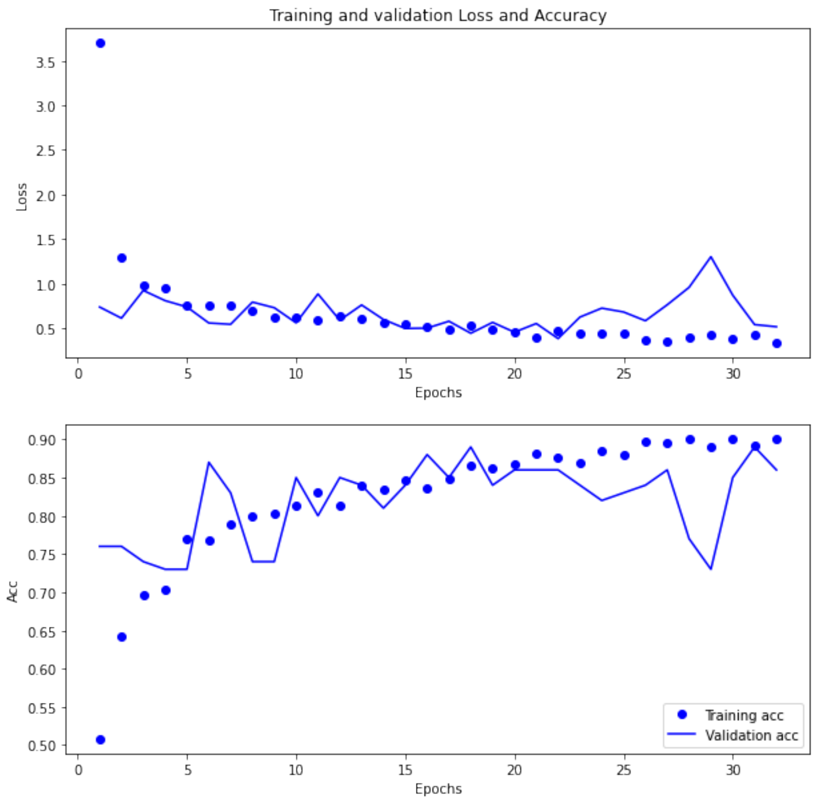

Figure 5.4: *Inception graphs*

	Precision	Recall	F1 score	Support
Brown Plant Hopper	0.40	0.40	0.40	20
Rice Gall Midge	0.40	0.40	0.40	20
Rice Leaf Hopper	0.60	0.60	0.60	20
Rice Leaf Roller	0.40	0.40	0.40	20
Stem Borer	0.80	0.80	0.80	20

Table 5.4: Classification report for Xception

The results achieved by the Xception model showed that although it had an average accuracy of 52%, it could identify the classes of Rice Leaf Hopper at 60% and Stem Borer at 80%. So, if the test set had a total of 100 images, the model successfully detected 80 Stem Borers and 60 Rice Leaf Hoppers.

	Precision	Recall	F1 score	Support
Brown Plant Hopper	0.40	0.40	0.40	20
Rice Gall Midge	0.60	0.60	0.60	20
Rice Leaf Hopper	0.60	0.60	0.60	20
Rice Leaf Roller	0.60	0.60	0.60	20
Stem Borer	0.40	0.40	0.40	20

Table 5.5: Classification report for Inception V3

The results achieved by the Inception V3 model showed that even though it had an average accuracy of 50%, it could identify the classes of Rice Gall Midge, Rice Leaf Hopper and Rice Leaf Roller all at 60%. For instance, if the test set had a total of 100 images, the model successfully detected 60 images of the said types.

Chapter 6

Conclusion and Future Work

Our goal was to examine alternative approaches to utilizing Machine Learning and Transfer Learning techniques to identify different types of pests. During our literature review, we came across multiple algorithms and tested our acquired dataset. Such Algorithms include: ResNet152, ResNet50, ResNet101, YOLOv5, Support Vector Machine(SVM), Xception and Inception V3. Among these, Support Vector Machine showed the highest accuracy and F1 score of about 0.725. Although transfer learning techniques are known to work better on a smaller dataset, they require a very high amount of hyper parameter optimization. Even though we have tried using Adam optimiser, SGD optimizer and RMSprop optimiser, at the end both transfer learning methods could capture 3 out of 5 classes with 40% accuracy. This leads us to believe we need to spend more time on data curation. Human involvement is needed to correctly acquire and identify different images for each label. Our dataset did not suit the transfer learning models but performed better with the machine learning model we chose. We want to try out hybrid approaches, maybe work with a binary model at first to identify whether there is a pest or not, and then using SVM (or any ML technique) to identify which kind of pest it is, to increase accuracy. A wider range of tranfer learning methods should be used further, to investigate which one would be better suited for the dataset we will choose, to increase the chances for our model to actually learn something from the data.

Bibliography

[1] P. Rajan, B. Radhakrishnan, and L. P. Suresh, "Detection and classification of pests from crop images using support vector machine," in *2016 international conference on emerging technological trends (ICETT)*, IEEE, 2016, pp. 1–6.

[2] R. U. Rani and P. Amsini, "Pest identification in leaf images using svm classifier," *International Journal of Computational Intelligence and Informatics*, vol. 6, no. 1, pp. 248–260, 2016.

[3] M. Ebrahimi, M.-H. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-based pest detection based on svm classification method," *Computers and Electronics in Agriculture*, vol. 137, pp. 52–58, 2017.

[4] A. A. Alfarisy, Q. Chen, and M. Guo, "Deep learning based classification for paddy pests & diseases recognition," in *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, 2018, pp. 21–25.

[5] M. Lee and S. Xing, "A study of tangerine pest recognition using advanced deep learning methods," 2018.

[6] D. Xiao, J. Feng, T. Lin, C. Pang, and Y. Ye, "Classification and recognition scheme for vegetable pests based on the bof-svm model," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 3, pp. 190–196, 2018.

[7] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, p. 939, 2019.

[8] W. Dawei, D. Limiao, N. Jiangong, G. Jiyue, Z. Hongfei, and H. Zhongzhi, "Recognition pest by image-based transfer learning," *Journal of the Science of Food and Agriculture*, vol. 99, no. 10, pp. 4524–4531, 2019.

[9] P. Kan-Rice, *Pests and pathogens place global burden on major food crops.* Agriculture and Natural Resources, University of California: Food Blog, 2019.

[10] Y. Song, X. Duan, Y. Ren, J. Xu, L. Luo, and D. Li, "Identification of the agricultural pests based on deep learning models," in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, IEEE, 2019, pp. 195–198.

[11] W. S. Souza, A. N. Alves, and D. L. Borges, "A deep learning model for recognition of pest insects in maize plantations," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, IEEE, 2019, pp. 2285–2290.

[12] K. Thenmozhi and U. S. Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Computers and Electronics in Agriculture*, vol. 164, p. 104 906, 2019.

[13] M. Türkoğlu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 3, pp. 1636–1651, 2019.

[14] E. Ayan, H. Erbay, and F. Varçın, "Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 179, p. 105 809, 2020.

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[16] X. Cao, Z. Wei, Y. Gao, and Y. Huo, "Recognition of common insect in field based on deep learning," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1634, 2020, p. 012 034.

[17] Y. Du, Y. Liu, and N. Li, "Insect detection research in natural environment based on faster-r-cnn model," in *Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence*, 2020, pp. 182–186.

[18] L. Jiao, S. Dong, S. Zhang, C. Xie, and H. Wang, "Af-rcnn: An anchor-free convolutional neural network for multi-categories agricultural pest detection," *Computers and Electronics in Agriculture*, vol. 174, p. 105 522, 2020.

[19] T. Kasinathan, D. Singaraju, and S. R. Uyyala, "Insect classification and detection in field crops using modern machine learning techniques," *Information Processing in Agriculture*, 2020.

[20] Y. Li, H. Wang, L. M. Dang, A. Sadeghi-Niaraki, and H. Moon, "Crop pest recognition in natural scenes using convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 169, p. 105 174, 2020.

[21] T.-L. Lin, H.-Y. Chang, and K.-H. Chen, "The pest and disease identification in the growth of sweet peppers using faster r-cnn and mask r-cnn," *Journal of Internet Technology*, vol. 21, no. 2, pp. 605–614, 2020.

[22] J. Liu and X. Wang, "Tomato diseases and pests detection based on improved yolo v3 convolutional neural network," *Frontiers in plant science*, vol. 11, p. 898, 2020.

[23] M. A. Mahmoud, P. Guo, and K. Wang, "Pseudoinverse learning autoencoder with dcgan for plant diseases classification," *Multimedia Tools and Applications*, vol. 79, no. 35, pp. 26 245–26 263, 2020.

[24] A. Morbekar, A. Parihar, and R. Jadhav, "Crop disease detection using yolo," in *2020 International Conference for Emerging Technology (INCET)*, IEEE, 2020, pp. 1–5.

[25] H. Nagar and R. Sharma, "A comprehensive survey on pest detection techniques using image processing," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2020, pp. 43–48.

[26] M. Ouhami, Y. Es-Saady, M. El Hajji, A. Hafiane, R. Canals, and M. El Yassa, "Deep transfer learning models for tomato disease detection," in *International Conference on Image and Signal Processing*, Springer, 2020, pp. 65–73.

[27] V. Ponnusamy, A. Coumaran, A. S. Shunmugam, K. Rajaram, and S. Senthilvelavan, "Smart glass: Real-time leaf disease detection using yolo transfer learning," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, IEEE, 2020, pp. 1150–1154.

[28] H. Tomar, "Multi-class image classification of fruits and vegetables using transfer learning techniques," Ph.D. dissertation, Dublin Business School, 2020.

[29] H. Wan, Z. Lu, W. Qi, and Y. Chen, "Plant disease classification using deep learning methods," in *Proceedings of the 4th International Conference on Machine Learning and Soft Computing*, 2020, pp. 5–9.

[30] H. Yin, Y. H. Gu, C.-J. Park, J.-H. Park, and S. J. Yoo, "Transfer learning-based search model for hot pepper diseases and pests," *Agriculture*, vol. 10, no. 10, p. 439, 2020.

[31] S. Cheeti, G. S. Kumar, J. S. Priyanka, G. Firdous, and P. R. Ranjeeva, "Pest detection and classification using yolo and cnn," *Annals of the Romanian Society for Cell Biology*, pp. 15 295–15 300, 2021.

[32] L. Cleetus, R. Sukumar, and N. Hemalatha, "Computational prediction of disease detection and insect identification using xception model," *bioRxiv*, 2021.

[33] B. A. L. Fatih and F. Kayaalp, "Review of machine learning and deep learning models in agriculture," *International Advanced Researches and Engineering Journal*, vol. 5, no. 2, pp. 309–323, 2021.

[34] M. Lippi, N. Bonucci, R. F. Carpio, M. Contarini, S. Speranza, and A. Gasparri, "A yolo-based pest detection system for precision agriculture," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2021, pp. 342–347.

[35] V. Malathi and M. Gopinath, "Classification of pest detection in paddy crop based on transfer learning approach," *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science*, pp. 1–8, 2021.

[36] M. P. Reddy and A. Deeksha, "Mulberry leaf disease detection using yolo," 2021.