Classification of Tree Leaf with Transfer Learning

COMPSCI 9637B Intro to Data Science

Neha Mary Thomas
University of Western Ontario
London, Canada
nthoma54@uwo.ca

Manideep Dara
University of Western Ontario
London, Canada
mdara@uwo.ca

Pavani Madhur
University of Western Ontario
London, Canada
pmadhur@uwo.ca

Oluwole Badmus
University of Western Ontario
London, Canada
Obadmus@uwo.ca

***Abstract*— The immense deforestation happening in the recent years led to the extinction of many tree species. It is thus important to preserve details of the trees. Leaves are easy to collect and can tell many details regarding the trees. We can use image classification to identify the leaves. The development of deep convolution network has been a major breakthrough in image classification. However, it is very difficult to develop and train a custom CNN, this is where transfer learning comes into the picture. Transfer Learning uses a pre trained model which was used to solve one problem be reused to solve another problem. In this study, we use pre trained models like 'ResNet 50' and 'VGG 16' along with classification algorithms like Decision Tree, Random Forest, Logistic Regression, KNN, Support Vector Machine (SVM) to classify images. Among all the models, SVM gave the best accuracy of 93% with VGG16 and Logistic Regression has given the best accuracy of 93.5% with ResNet 50.**

I. INTRODUCTION

The natural environment being severely damaged in many countries resulted in the reduction of forest area and the extinction of many species. To preserve the information about trees before extinction, the best and simple method is collecting leaves as they are readily available. This information can be useful for botanist, food engineers, industrialist. Plant image recognition and classification has enjoyed tremendous attention from wide section of interdisciplinary research areas as a reflection of the adaptation of computer science methods into various seemingly unrelated domains. This phenomenon has been enhanced by the growing sophistication and application of computational tools to augment, automate and speed up what was hitherto an exclusive domain of human intellect and segregated expertise in the recent past. Plant and/or leaf phenotyping is one of such areas that have generated significant interests lately Music can be classified by using lyrical analysis

The identification of plants can be based on their shape, color, textures which can be extracted through image processing and identified using machine learning algorithms. The development of deep convolution networks may take years and results in high computational cost due to the large number of parameters that needs to be trained. Model development also requires practitioners with deep learning expertise to make appropriate decisions such as what forms of regularization to apply, what architecture is appropriate, what type of pooling to incorporate, how to interpret training signal and debug appropriately (Neeraj Kumar, 2012). This led to the immense use of transfer learning especially in deep learning. **Transfer learning (TL)** is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem (Warnick, Sean).

In our study we use transfer learning as a solution to overcome the slow development of the model **and also using classification** algorithms results in much lesser parameters and smaller Hyperparameters. The pre-trained models used are ResNet 50 and VGG 16. The classification algorithms used are Support Vector Machine (SVM), KNN, Logistic Regression, Random Forest and Decision Trees. The rest of the paper is organized as follows: Section II presents the existing work done on the face recognition and song classification modules. Section III describes the data used in the project. Section IV explains the methods implemented as a part of this project. Section V presents the results of the project along with the proposed application. Section VI talks about further enhancements while section VII concludes the paper.

II. RELATED WORK

Chaudhury and Barron (2020) developed mechanisms for recognizing leaf's species even when such leaves are partially covered by other objects. What stands their work apart is the ability to go beyond shaping recognition to identify species to using contour dimensioning to generate a complete picture of the specie type. prior to that, Mattos, Herrman et al. (2014) had proposed a flower classification method that was eventually implemented in mobile phones. Their method involved classifying flower breeds based on comparison of color, shape and texture cues using metric learning for feature weighting. In a similar manner, Le, Tran and Hand (2014) had developed an automated leaf identification system using an improved version of the Kernel Descriptors algorithm to identify and classify Vietnamese medicinal plants. To overcome the recognition problem associated with within-specie plant image recognition, Feitoza, Silva and Calumny (2019) used a comparative study of Deep Neural Networks to evaluate the best vectorized deep features to overcome this problem. By combining a range of supervised learning methods with two different cross-validated CNN models, they suggested that linear kernel SVM classifier and One-vs-Rest decomposition can provide a satisfactory classification method to overcoming this problem. Building on this important discovery, we conducted a study that compared which machine learning method provide the best transfer learning in identifying leaf species data generated from two CNN models I.e., VGG16 and ResNet50

III. DATA

Our dataset is a Leafsnap (Neeraj Kumar, 2012), a collection of images used for the Leafsnap application which is used for leaf recognition. This dataset is developed by researchers from Columbia University, the University of Maryland, and the Smithsonian Institution. This dataset contains various tree leaves from the United States of America and Canada.

The dataset consists of 30866 leaf images, 23147 of which are high-quality lab images and 7719 of which are lower-quality field images. The lab images feature flattened leaves and are taken under the controlled back and front lighting. On the other hand, the field images feature unpressed leaves, taken by mobile devices outdoors. Each image is labeled with the species of the tree associated with the leaf, and there are 185 different classes of leaf present in the dataset.

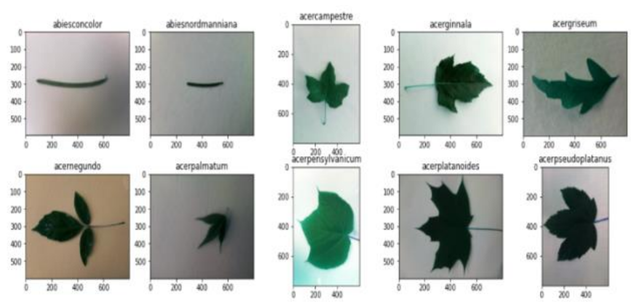


Fig 1 Sample images from the dataset

When we tried to experiment with the whole dataset the feature vector from the CNN model has 1500000 columns which is technically impossible to run in our systems. So due to this computation constraint, we have conducted the experiments on the 30 classes of field images.

For each class, we have number of images respectively. From the below table we can say that every class has a fair number of images and our dataset is balanced.

For the train and test split, we have taken 70% of the dataset as a training set and the remaining 30% as a test set.

TABLE I
Leaf Snap DATASET

Class Name	Image Count
abiesconcolor	51
abiesnordmanniana	35
acercampestre	36
acerginnala	31
acergriseum	46
acernegundo	33
acerpalmatum	92
acerpensylvanicum	35
acerplatanoides	20
acerpseudoplatanus	15
acerrubrum	45
acersaccharinum	39
acersaccharum	24
aesculusflava	15
aesculusglabra	44
aesculushippocastamon	19
aesculuspavi	104
ailanthusaltissima	16
albiziajulibrissin	28
amelanchierarborea	43
amelanchiercanadensis	35
amelanchierlaevis	31
asiminatriloba	49
betulaalleghaniensis	33
betulajacqemontii	22
betulalenta	11
betulanigra	20
betulapopulifolia	26
broussonettiapapyrifera	54
carpinusbetulus	15

IV ARCHITECTURE

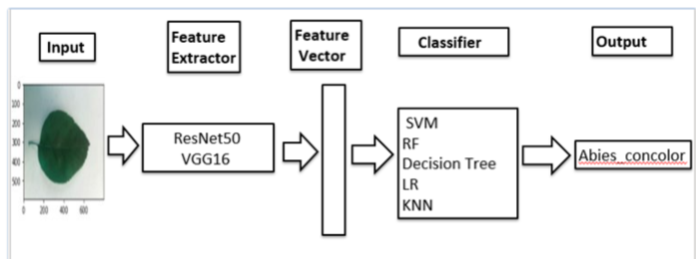


Fig 2 Model Architecture

The above picture is the visual representation of our model architecture. In general, the output of convolution layers is fed to the fully connected layers for classification purposes. Since we are experimenting with classification with transfer learning the topmost layers are removed. The output from the convolutions layers will be our feature extractors and the input to the classification algorithms. We are using pre-trained models Resnet50 and VGG16 with imagenet weights.

In our dataset, some images have different sizes. So to keep every image in a similar size we have resized each image to 64*64. In classification algorithms, to decrease computation time and to keep every feature in same scale we have used the standard scalar.

IV METHODS

A. Transfer Learning Algorithms

Resnet-50

ResNet50 is a variant of the ResNet model which has 48 Convolution layers along with 1 Ma Pool and 1 Average Pool layer. As we know that Deep Convolutional neural networks are great at identifying low, mid, and high-level features from the images, and stacking more layers gives us better accuracy. But at some point, the accuracy gets saturated and then degrades rapidly. To overcome this problem authors (Kaiming He, 2015) have introduced a deep residual learning framework. So, for this, they introduced shortcut connections that simply perform identity mappings. The benefit of these shortcut identity mapping was that there were no additional parameters added to the model and the computational time was kept in check.

The image below shows the Resnet-34 architecture which is initially developed and later researchers developed the Resnet-50 by stacking 50 layers.

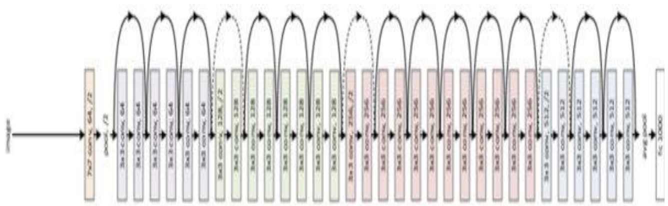


Fig. 3 : Resnet-34 Architecture

VGG-16

It is a Convolution Neural Network which trained over ImageNet which is a dataset with over 14 million images and 1000 classes. This model is an improvement over AlexNet by replacing large kernel-sized filters with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU's (Muneeb Ul Hassan). There are 16 layers in this architecture.

The input to a VGG architecture is an RGB image of 224*224. The image is passed through a stack of convolution layers with a filter size of 3. There are 3 fully connected layers of which the first 2 layers have 4096 channels each and the third one is a 1000 channel corresponding to the 1000 classes in the ImageNet dataset.

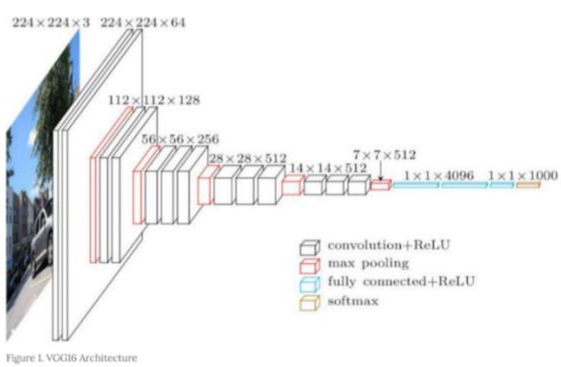


Fig 4: VGG Architecture

B. Hyper Parameter Tuning Methods

Hyperparameter of an algorithm controls the capacity of a model. It includes flexibility, degrees of freedom it has in fitting the data. Proper control of model capacity can prevent overfitting, which happens when the model is too flexible, and the training process adapts too much to the training data losing predictive accuracy on new test data. The different methods for hyperparameter tuning include:

- GridSearchCV - This function comes under Scikit-learn's model_selection package. It tries all combinations of values passed in the dictionary and using cross-validation evaluates the model and the hyperparameter with the maximum accuracy is returned.
- RandomizedSearchCV- This function comes under Scikit-learn's model_selection package. It works similar to GridSearchCV except that not all combinations are tried out but a fixed number of parameter settings are sampled from the dictionary hence it is quicker than GridSearchCV

The hyperparameter tuning used for this project is GridSearchCV as RandomizedSearchCV returns different hyperparameter for every execution as the combinations taken for hyperparameter tuning may vary with each execution and only a limited number of combinations can be tried for every execution.

C. Classification Algorithms

Decision Trees

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The topmost decision node in a tree which corresponds to the best predictor called root node. Information gain is a statistical property that measures how well a given attribute separates the training examples according to their target classification. The commonly used information gain is Entropy or Gini.

The hyperparameters used to get the best model are criterion, splitter, max_depth, min_samples_split, min_samples_leaf, max_features.

Random Forest

It is a supervised learning algorithm which is used for classification and regression tasks. It was designed to overcome the high variance in individual decision trees. It utilizes multiple decision tree and the output of every decision tree is cumulated to predict the final result of the data point. The hyperparameters used for this project includes:

a) n_estimators – Denotes the number of trees in the forest

b) max_depth - Maximum number of levels in each tree

K-nearest Neighbors Algorithm (KNN)

This supervised learning algorithm classifies data points according to how its neighbours are classified. It is used for classification and regression problems. The most common hyperparameter is n_neighbors (k) which is the number of neighbours that needs to be considered while taking a majority vote to classify the data point. For regression problems, the Euclidian distance between the data point to be classified and already classified data points are calculated and depending on the number of neighbours we need to rank the distance in ascending order and fetch the mean of first k Euclidian distance.

KNN with VGG-16 feature Extractor:

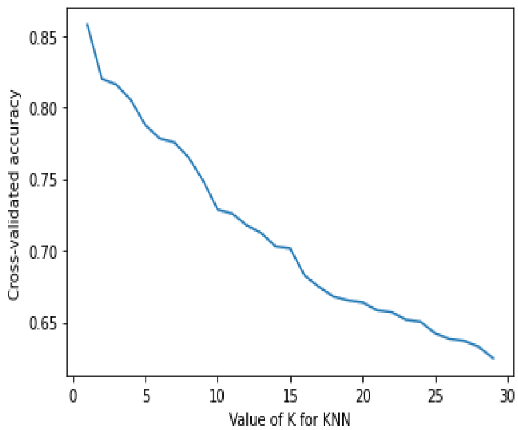


Fig. 5: VGG - KNN- Hyperparameter Tuning

KNN with ResNet-50 feature Extractor:

```
Text(0, 0.5, 'Cross-validated accuracy')
```

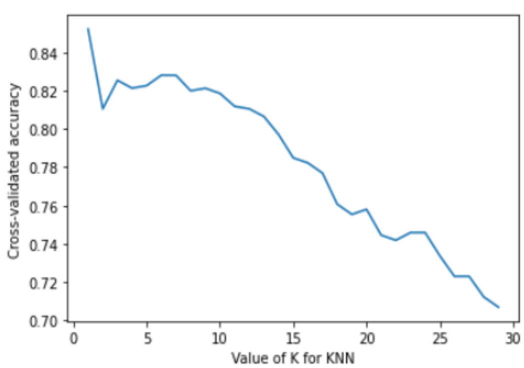


Fig. 6: Resnet - KNN- Hyperparameter Tuning

Support Vector Machine (SVM)

This supervised learning algorithm works on the principle of finding a hyperplane in an N-dimensional space where N is the number of features that can distinctly classify the data points (RohitGandhi,2018). The hyperparameters used in SVM are ‘C’ which controls the amount of regularization in data, gamma which controls the distance of influence of a single training point. Finding an optimal hyperplane is relatively easy for linearly separable data but for non-linear data this algorithm makes use of kernel. The main idea is that the data points are mapped on to a different dimensional space in which they are linearly separable. A kernel function shows a similarity measure between the data points of the original and new dimensional space. We have used Linear Kernel in our project

Linear Kernel –The hyperparameter for Linear Kernel is ‘C’. Training an SVM with Linear Kernel is faster than other kernels.

SVM with VGG-16 feature Extractor:

```
Text(0, 0.5, 'Mean Test Score of training data')
```

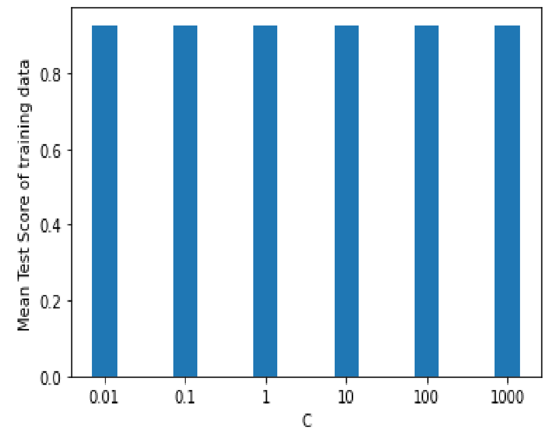


Fig 7. VGG- SVM Hyperparameter tuning for Linear Kernel

SVM with ResNet-50 feature Extractor:

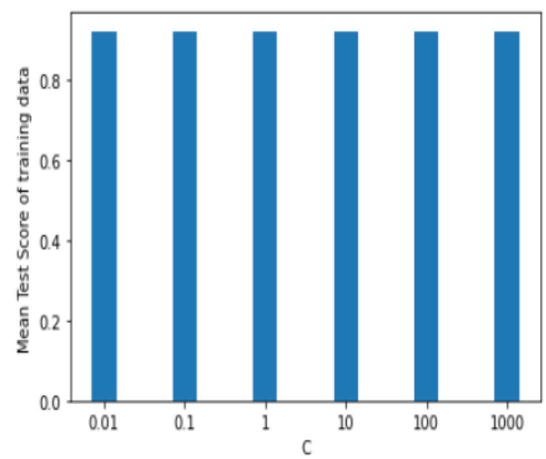


Fig 8. VGG- SVM Hyperparameter tuning for Linear Kernel

Logistic Regression

Logistic regression is a supervised learning algorithm that is useful for classifying outcomes into this or that group (i.e., binary LR) or multiple cases, known as multinomial logistic regression. Beyond the ordinary predictive power that simple regression can do, an advantage of logistic regression is that it allows the evaluation of multiple explanatory variables by extension of the basic principles into explaining the relationship between one dependent binary variable and one or more categorical or continuous independent variables. While the logistic regression is computationally cheap, it does however suffer accuracy problems (Chang, 2020)

TABLE II
Hyperparameter Tuning

Feature Extractor	Algorithm	HyperParameter	Best Hyperparameter
VGG-16	Random Forest	n-estimator	900
		max-depth	15
	Decision Trees	criterion	'entropy'
		splitter	best
		Max_depth	9
		min_samples_split	2
		min_samples_leaf	1
		max_features	None
	Logistic Regression	C	0.367
		penalty	L2
	KNN	n_neighbors	1
	SVM	C	0.01
ResNet-50	Random Forest	n-estimator	900
		max-depth	15
	Decision Trees	criterion	'entropy'
		splitter	best
		Max_depth	7
		min_samples_split	2
		min_samples_leaf	2
		max_features	None
	Logistic Regression	C	0.367
		penalty	L2
	KNN	n_neighbors	1
	SVM	C	0.01

V. RESULTS

A. *VGG-16 Feature Extraction*

Decision Trees, Random Forest, Logistic Regression, KNN, SVM have been applied on the features which are extracted from the VGG-16 feature extractor and the following accuracies were obtained.

TABLE III
Accuracy Table using VGG-16

Feature Extractor	Classification Model	Accuracy
VGG-16	Decision Tree	58.13
	Random Forest	86.6
	Logistic Regression	92
	KNN	85
	SVM	93

Decision Trees

	precision	recall	f1-score	support
abiesconcolor	0.83	1.00	0.91	15
abiesnordmanniana	0.92	1.00	0.96	11
acercampestre	0.75	0.55	0.63	11
acerginnala	0.20	0.11	0.14	9
acergriseum	0.62	0.57	0.59	14
acernegundo	0.18	0.20	0.19	10
acerpalmatum	0.73	0.79	0.76	28
acerpensylvanicum	0.55	0.55	0.55	11
acerplatanoides	0.33	0.17	0.22	6
acerpseudoplatanus	0.40	0.50	0.44	4
acerrubrum	0.43	0.43	0.43	14
acersaccharinum	0.73	0.67	0.70	12
acersaccharum	0.50	0.43	0.46	7
aesculusflava	0.00	0.00	0.00	4
aesculusglabra	0.50	0.85	0.63	13
aesculushippocastamon	0.14	0.17	0.15	6
aesculuspavi	0.79	0.73	0.76	30
ailanthusaltissima	0.00	0.00	0.00	5
albiziajulibrissin	0.62	0.62	0.62	8
amelanchierarborea	0.22	0.15	0.18	13
amelanchiercanadensis	0.50	0.55	0.52	11
amelanchierlaevis	0.67	0.67	0.67	9
asiminatriloba	0.83	0.67	0.74	15
betulaalleghaniensis	0.50	0.60	0.55	10
betulajacqemontii	0.50	0.86	0.63	7
betulalenta	0.12	0.33	0.18	3
betulanigra	0.40	0.33	0.36	6
betulapopulifolia	0.80	0.50	0.62	8
broussonettiapapyrifera	0.87	0.81	0.84	16
carpinusbetulus	0.00	0.00	0.00	4
accuracy			0.58	320
macro avg	0.49	0.49	0.48	320
weighted avg	0.58	0.58	0.57	320

Fig 9: Classification report for Decision Trees using VGG-16 extractor

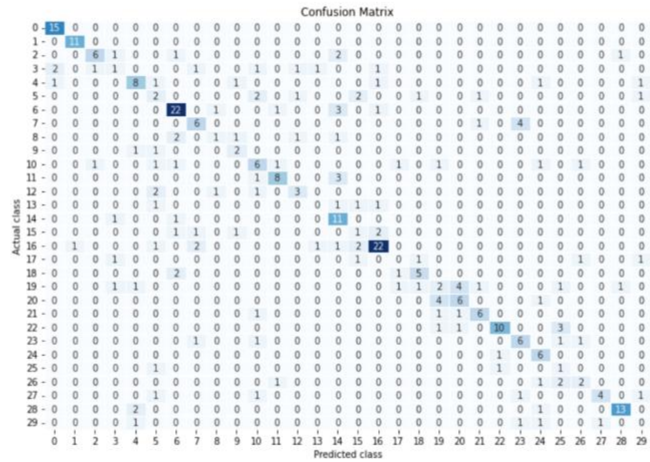


Fig 10: Confusion Matrix for Decision Trees using VGG-16 extractor

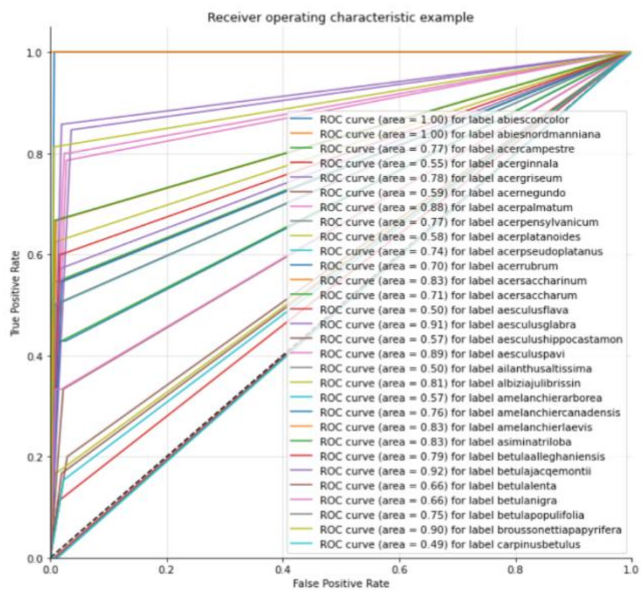


Fig 11: ROC curves for all the labels of Decision Trees using VGG-16 extractor

Random Forest

	precision	recall	f1-score	support
abiesconcolor	0.900	1.000	0.947	18
abiesnordmanniana	0.875	0.778	0.824	9
acercampestre	0.833	1.000	0.909	5
acerginnala	1.000	1.000	1.000	7
acergriseum	0.846	1.000	0.917	11
acernegundo	0.833	0.714	0.769	7
acerpalmatum	0.889	0.970	0.928	33
acerpensylvanicum	0.857	0.857	0.857	7
acerplatanoides	1.000	1.000	1.000	8
acerpseudoplatanus	0.667	0.500	0.571	4
acerrubrum	0.842	0.941	0.889	17
acersaccharinum	0.867	1.000	0.929	13
acersaccharum	1.000	0.889	0.941	9
aesculusflava	1.000	0.333	0.500	3
aesculusglabra	1.000	0.769	0.870	13
aesculushippocastamon	1.000	0.333	0.500	3
aesculuspavi	0.854	0.972	0.909	36
ailanthusaltissima	1.000	0.429	0.600	7
albiziajulibrissin	0.889	1.000	0.941	8
amelanchierarborea	0.444	0.667	0.533	12
amelanchiercanadensis	0.583	0.636	0.609	11
amelanchierlaevis	0.875	0.778	0.824	9
asiminatriloba	1.000	0.933	0.966	15
betulaalleghaniensis	1.000	0.750	0.857	8
betulajacqemontii	0.833	0.833	0.833	6
betulalenta	1.000	0.333	0.500	3
betulanigra	0.800	0.800	0.800	5
betulapopulifolia	1.000	0.875	0.933	8
broussonettiapapyrifera	1.000	0.857	0.923	21
carpinusbetulus	1.000	1.000	1.000	4
accuracy			0.866	320
macro avg	0.890	0.798	0.819	320
weighted avg	0.883	0.866	0.862	320

Fig 12: Classification report for Random Forest using VGG-16 extractor

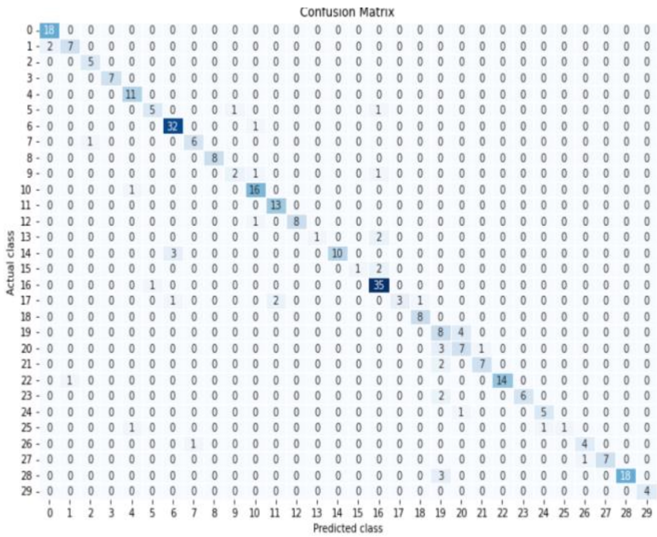


Fig 13: Confusion Matrix for Random Forest using VGG-16 extractor

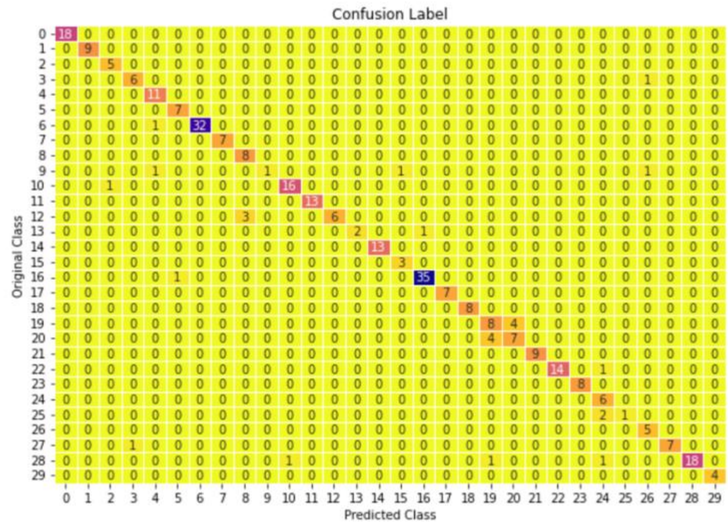


Fig 16: Confusion Matrix for Logistic Regression using VGG-16 extractor

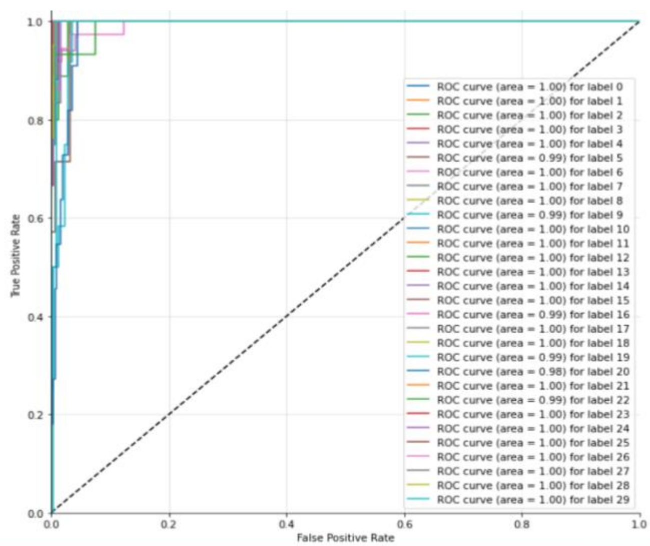
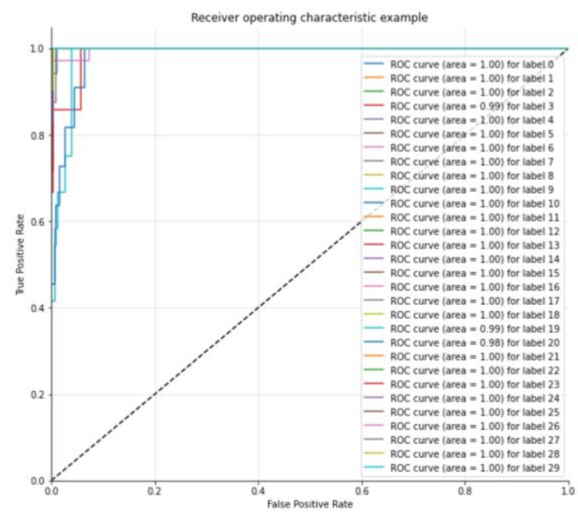


Fig 14: ROC curves for all the labels of Random Forest using VGG-16 extractor



<Figure size 432x288 with 0 Axes>

Fig 17: ROC curves for all the labels of Logistic Regression using VGG-16 extractor

Logistic Regression:

	precision	recall	f1-score	support
abiesconcolor	1.00	1.00	1.00	18
abiesnordmanniana	1.00	1.00	1.00	9
acercampestre	0.83	1.00	0.91	5
acerginnala	0.67	0.86	0.75	7
acergriseum	0.85	1.00	0.92	11
acernegundo	0.88	1.00	0.93	7
acerpalmatum	1.00	0.94	0.97	33
acerpensylvanicum	1.00	1.00	1.00	7
acerplatanoides	0.73	1.00	0.84	8
acerpseudoplatanus	1.00	0.25	0.40	4
acerrubrum	1.00	0.94	0.97	17
acersaccharinum	1.00	1.00	1.00	13
acersaccharum	1.00	0.67	0.80	9
aesculusflava	1.00	0.67	0.80	3
aesculusglabra	1.00	1.00	1.00	13
aesculushippocastamon	0.75	1.00	0.86	3
aesculuspavi	0.97	0.97	0.97	36
ailanthusaltissima	1.00	1.00	1.00	7
albiziajulibrissin	1.00	1.00	1.00	8
amelanchierarborea	0.67	0.67	0.67	12
amelanchiercanadensis	0.64	0.64	0.64	11
amelanchierlaevis	1.00	1.00	1.00	9
asiminatriloba	1.00	0.93	0.97	15
betulaalleghaniensis	1.00	1.00	1.00	8
betulajacqemontii	0.60	1.00	0.75	6
betulalenta	1.00	0.33	0.50	3
betulanigra	0.71	1.00	0.83	5
betulapopulifolia	1.00	0.88	0.93	8
broussonettiapapyrifera	1.00	0.90	0.95	21
carpinusbetulus	1.00	1.00	1.00	4
accuracy			0.92	320
macro avg	0.91	0.89	0.88	320
weighted avg	0.93	0.92	0.92	320

Fig 15: Classification report of Logistic Regression using VGG-16 extractor

KNN

	precision	recall	f1-score	support
abiesconcolor	1.00	1.00	1.00	18
abiesnordmanniana	1.00	1.00	1.00	9
acercampestre	0.67	0.80	0.73	5
acerginnala	0.83	0.71	0.77	7
acergriseum	0.85	1.00	0.92	11
acernegundo	0.83	0.71	0.77	7
acerpalmatum	1.00	0.94	0.97	33
acerpensylvanicum	1.00	1.00	1.00	7
acerplatanoides	0.73	1.00	0.84	8
acerpseudoplatanus	1.00	0.25	0.40	4
acerrubrum	0.78	0.82	0.80	17
acersaccharinum	0.76	1.00	0.87	13
acersaccharum	0.60	0.33	0.43	9
aesculusflava	1.00	0.67	0.80	3
aesculusglabra	0.72	1.00	0.84	13
aesculushippocastamon	0.60	1.00	0.75	3
aesculuspavi	0.97	0.94	0.96	36
ailanthusaltissima	1.00	0.71	0.83	7
albiziajulibrissin	1.00	1.00	1.00	8
amelanchierarborea	0.60	0.75	0.67	12
amelanchiercanadensis	0.75	0.55	0.63	11
amelanchierlaevis	0.70	0.78	0.74	9
asiminatriloba	0.93	0.93	0.93	15
betulaalleghaniensis	1.00	0.88	0.93	8
betulajacqemontii	0.55	1.00	0.71	6
betulalenta	1.00	0.33	0.50	3
betulanigra	0.57	0.80	0.67	5
betulapopulifolia	1.00	0.75	0.86	8
broussonettiapapyrifera	1.00	0.76	0.86	21
carpinusbetulus	1.00	0.75	0.86	4
accuracy			0.85	320
macro avg	0.85	0.81	0.80	320
weighted avg	0.87	0.85	0.85	320

Fig18: Classification report for KNN using VGG-16 extractor

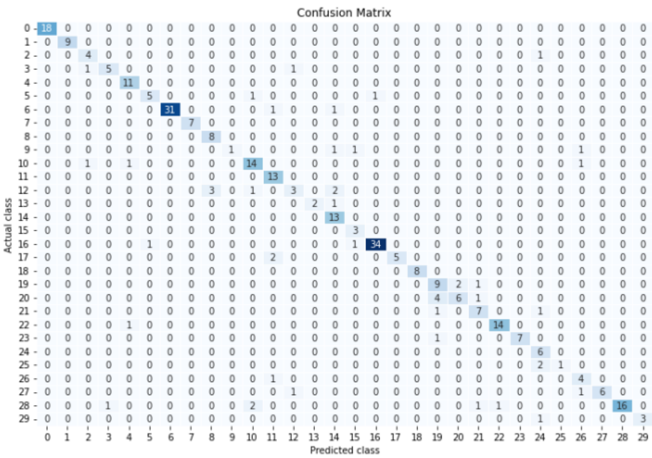


Fig 19: Confusion Matrix for KNN using VGG-16 extractor

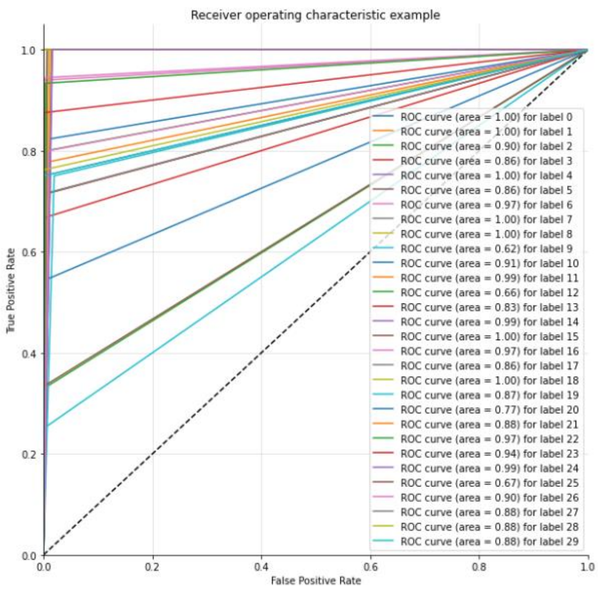


Fig 20: ROC curves for all the labels of KNN using VGG-16

Support Vector Machine

	precision	recall	f1-score	support
abiesconcolor	1.00	1.00	1.00	18
abiesnordmanniana	1.00	1.00	1.00	9
acercampestre	0.71	1.00	0.83	5
acerginnala	0.75	0.86	0.80	7
acergriseum	0.85	1.00	0.92	11
acernegundo	0.86	0.86	0.86	7
acerpalmatum	1.00	0.94	0.97	33
acerpensylvanicum	1.00	1.00	1.00	7
acerplatanoides	0.73	1.00	0.84	8
acerpseudoplatanus	1.00	0.50	0.67	4
acerrubrum	1.00	0.88	0.94	17
acersaccharinum	1.00	1.00	1.00	13
acersaccharum	1.00	0.67	0.80	9
aesculusflava	1.00	1.00	1.00	3
aesculusglabra	0.93	1.00	0.96	13
aesculushippocastamon	1.00	1.00	1.00	3
aesculuspavi	0.97	0.97	0.97	36
ailanthusaltissima	1.00	1.00	1.00	7
albiziajulibrissin	1.00	1.00	1.00	8
amelanchierarborea	0.71	0.83	0.77	12
amelanchiercanadensis	0.80	0.73	0.76	11
amelanchierlaevis	1.00	1.00	1.00	9
asiminatriloba	1.00	0.93	0.97	15
betulaalleghaniensis	1.00	1.00	1.00	8
betulajacqemontii	0.67	1.00	0.80	6
betulalenta	1.00	0.33	0.50	3
betulanigra	0.83	1.00	0.91	5
betulapopulifolia	1.00	0.88	0.93	8
broussonettiapapyrifera	1.00	0.95	0.98	21
carpinusbetulus	1.00	1.00	1.00	4
accuracy			0.93	320
macro avg	0.93	0.91	0.91	320
weighted avg	0.94	0.93	0.93	320

Fig 21: Classification report for SVM using VGG-16 extractor

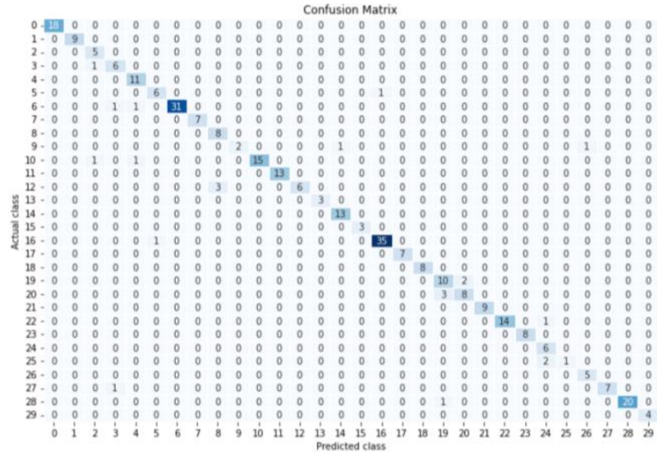


Fig 22: Confusion Matrix for SVM using VGG-16 extractor

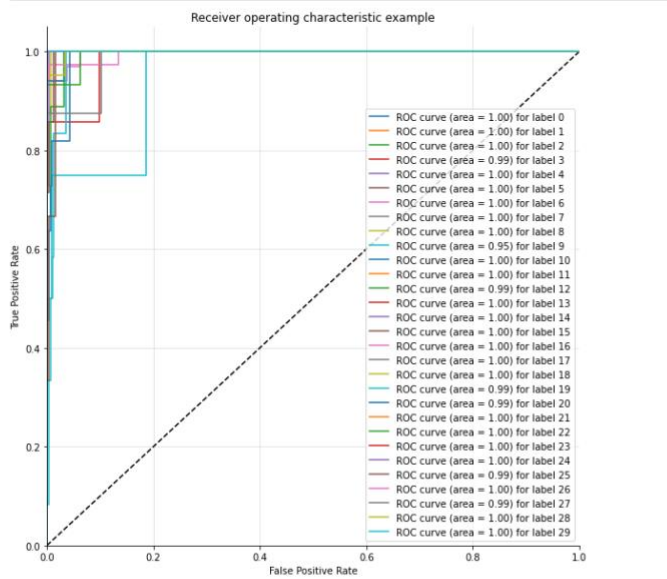


Fig 23: ROC curves for all the labels of SVM using VGG-16 extractor

B. ResNet-50 Feature Extraction

Decision Trees, Random Forest, Logistic Regression, KNN, SVM have been applied on the features which are extracted from the Resnet-50 feature extractor and the following accuracies were obtained.

TABLE IV

Accuracy Table using ResNet-50

Feature Extractor	Classification Model	Accuracy
Resnet-50	Decision Tree	54.37
	Random Forest	81.9
	Logistic Regression	93.5
	KNN	84
	SVM	90.93

Decision Trees

	precision	recall	f1-score	support
abiesconcolor	0.88	1.00	0.94	15
abiesnordmanniana	0.75	0.82	0.78	11
acercampestre	0.60	0.55	0.57	11
acerginnala	0.50	0.56	0.53	9
acergriseum	0.44	0.57	0.50	14
acernegundo	0.50	0.30	0.37	10
acerpalmatum	0.76	0.68	0.72	28
acerpensylvanicum	0.50	0.45	0.48	11
acerplatanoides	0.60	0.50	0.55	6
acerpseudoplatanus	0.40	0.50	0.44	4
acerrubrum	0.38	0.43	0.40	14
acersaccharinum	0.64	0.75	0.69	12
acersaccharum	0.30	0.43	0.35	7
aesculusflava	0.14	0.25	0.18	4
aesculusglabra	0.60	0.69	0.64	13
aesculushippocastamon	0.00	0.00	0.00	6
aesculuspavi	0.69	0.67	0.68	30
ailanthusaltissima	0.40	0.40	0.40	5
albiziajulibrissin	1.00	0.62	0.77	8
amelanchierarborea	0.38	0.38	0.38	13
amelanchiercanadensis	0.46	0.55	0.50	11
amelanchierlaevis	0.38	0.56	0.45	9
asiminatriloba	0.64	0.47	0.54	15
betulaalleghaniensis	0.44	0.40	0.42	10
betulajacqemontii	0.67	0.86	0.75	7
betulalenta	0.00	0.00	0.00	3
betulanigra	0.25	0.17	0.20	6
betulapopulifolia	0.60	0.38	0.46	8
broussonettiapapyrifera	0.29	0.31	0.30	16
carpinusbetulus	0.50	0.50	0.50	4
accuracy			0.54	320
macro avg	0.49	0.49	0.48	320
weighted avg	0.55	0.54	0.54	320

Fig 23: Classification report for Decision Trees using ResNet-50 extractor

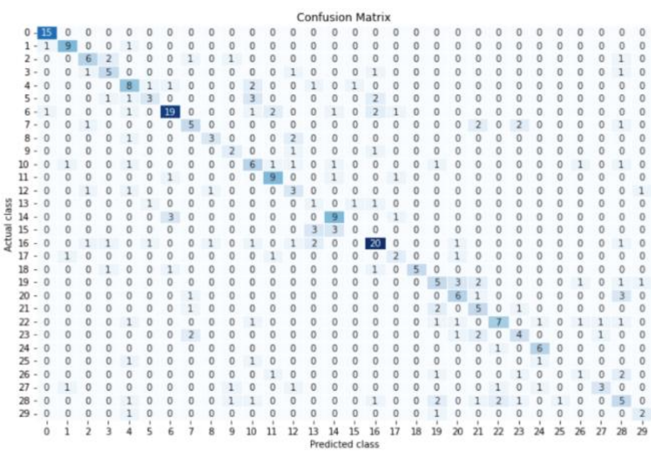


Fig 24: Confusion Matrix for Decision Trees using ResNet-50 extractor

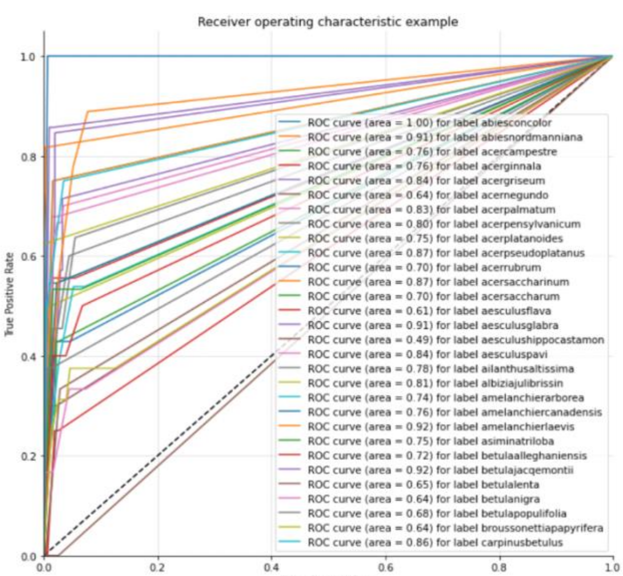


Fig 25: ROC curves for all the labels of Decision Trees using ResNet-50 extractor

Random Forest

	precision	recall	f1-score	support
abiesconcolor	0.882	1.000	0.938	15
abiesnordmanniana	1.000	0.909	0.952	11
acercampestre	0.769	0.909	0.833	11
acerginnala	1.000	0.778	0.875	9
acergriseum	0.591	0.929	0.722	14
acernegundo	1.000	0.900	0.947	10
acerpalmatum	0.867	0.929	0.897	28
acerpensylvanicum	0.900	0.818	0.857	11
acerplatanoides	1.000	0.667	0.800	6
acerpseudoplatanus	0.000	0.000	0.000	4
acerrubrum	0.625	0.714	0.667	14
acersaccharinum	0.917	0.917	0.917	12
acersaccharum	0.667	0.571	0.615	7
aesculusflava	0.000	0.000	0.000	4
aesculusglabra	0.688	0.846	0.759	13
aesculushippocastamon	0.500	0.333	0.400	6
aesculuspavi	0.857	1.000	0.923	30
ailanthusaltissima	1.000	0.200	0.333	5
albiziajulibrissin	1.000	1.000	1.000	8
amelanchierarborea	0.588	0.769	0.667	13
amelanchiercanadensis	0.667	0.545	0.600	11
amelanchierlaevis	1.000	0.889	0.941	9
asiminatriloba	0.933	0.933	0.933	15
betulaalleghaniensis	0.727	0.800	0.762	10
betulajacqemontii	1.000	0.857	0.923	7
betulalenta	1.000	0.333	0.500	3
betulanigra	1.000	1.000	1.000	6
betulapopulifolia	1.000	0.750	0.857	8
broussonettiapapyrifera	0.833	0.938	0.882	16
carpinusbetulus	1.000	0.500	0.667	4
accuracy			0.819	320
macro avg	0.800	0.724	0.739	320
weighted avg	0.818	0.819	0.805	320

Fig 26: Classification report for Random Forest using ResNet-50 extractor

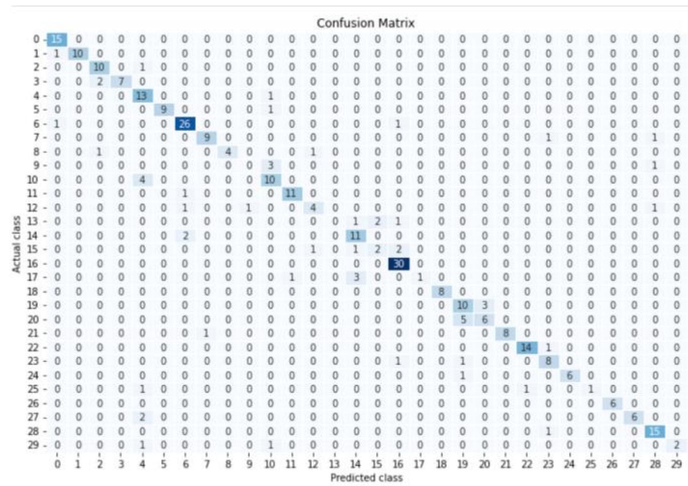


Fig 27: Confusion Matrix for Random Forest using ResNet-50 extractor

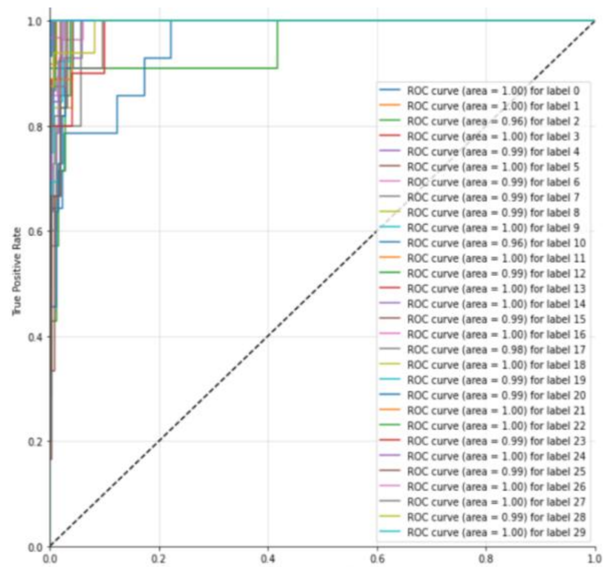


Fig 28: ROC curves for all the labels of Random Forest using ResNet-50 extractor

Logistic Regression:

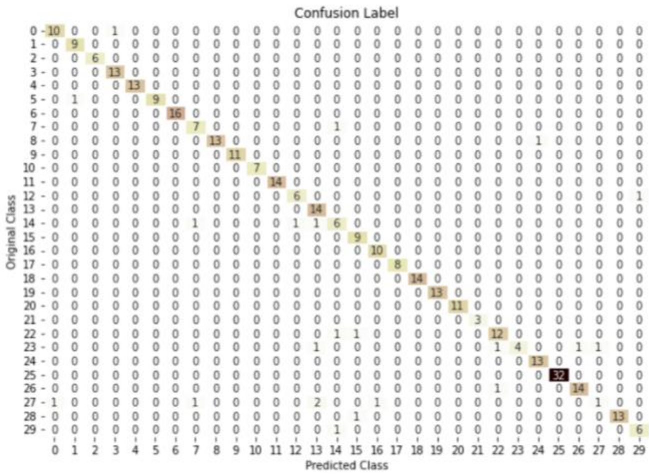


Fig 29: Confusion Matrix for Logistic Regression using ResNet-50

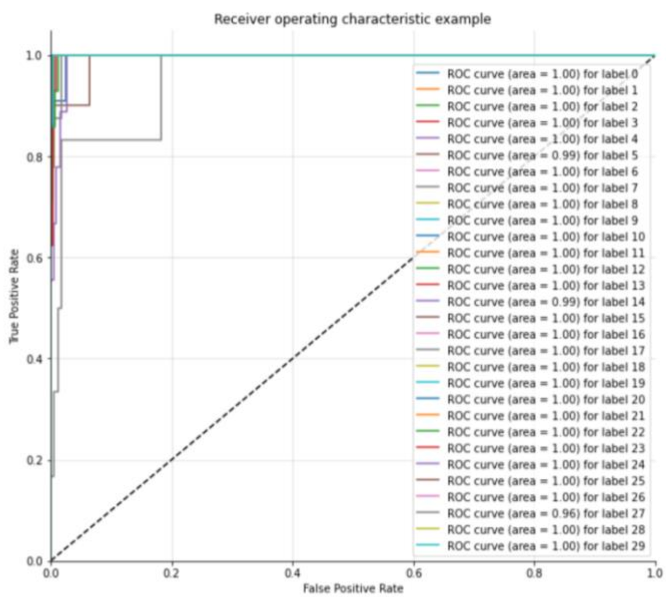


Fig 30: ROC curves for all the labels of Logistic Regression using ResNet-50 extractor

KNN

	precision	recall	f1-score	support
abiesconcolor	1.00	1.00	1.00	15
abiesnordmanniana	1.00	1.00	1.00	11
acercampestre	0.77	0.91	0.83	11
acerginnala	0.64	1.00	0.78	9
acergriseum	0.85	0.79	0.81	14
acernegundo	0.88	0.70	0.78	10
acerpalmatum	0.96	0.93	0.95	28
acerpensylvanicum	1.00	0.73	0.84	11
acerplatanoides	1.00	0.50	0.67	6
acerpseudoplatanus	0.57	1.00	0.73	4
acerrubrum	0.73	0.79	0.76	14
acersaccharinum	0.85	0.92	0.88	12
acersaccharum	0.83	0.71	0.77	7
aesculusflava	0.75	0.75	0.75	4
aesculusglabra	0.85	0.85	0.85	13
aesculushippocastamon	1.00	1.00	1.00	6
aesculuspavi	0.91	0.97	0.94	30
ailanthusaltissima	0.67	0.40	0.50	5
albiziajulibrissin	1.00	0.88	0.93	8
amelanchierarborea	0.50	0.54	0.52	13
amelanchiercanadensis	0.50	0.45	0.48	11
amelanchierlaevis	0.80	0.89	0.84	9
asiminatriloba	1.00	0.93	0.97	15
betulaalleghaniensis	0.64	0.90	0.75	10
betulajacqemontii	1.00	0.86	0.92	7
betulalenta	0.75	1.00	0.86	3
betulanigra	1.00	1.00	1.00	6
betulapopulifolia	0.88	0.88	0.88	8
broussonettiapapyrifera	1.00	0.69	0.81	16
carpinusbetulus	0.80	1.00	0.89	4
accuracy			0.84	320
macro avg	0.84	0.83	0.82	320
weighted avg	0.86	0.84	0.84	320

Fig 31: Classification report for KNN using ResNet-50 extractor

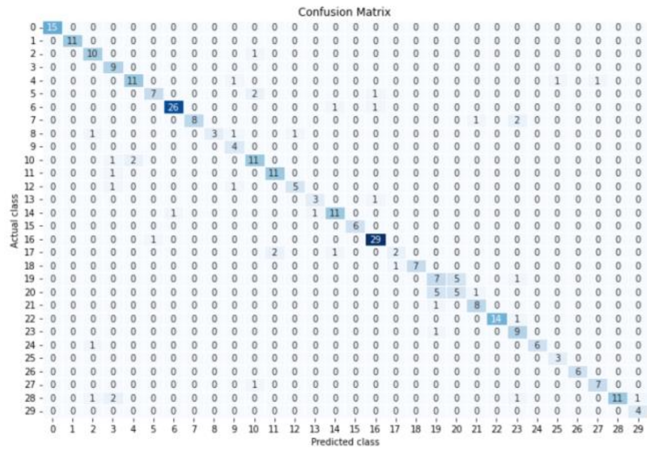


Fig 31: Confusion Matrix for KNN using ResNet-50 extractor

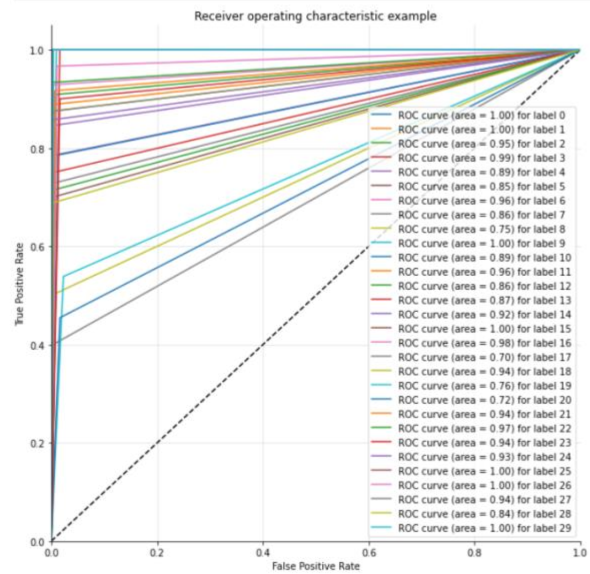


Fig 32: ROC curves for all the labels of KNN using ResNet-50

Support Vector Machine

	precision	recall	f1-score	support
abiesconcolor	1.00	1.00	1.00	15
abiesnordmanniana	1.00	1.00	1.00	11
acercampestre	0.83	0.91	0.87	11
acerginnala	0.89	0.89	0.89	9
acergriseum	0.81	0.93	0.87	14
acernegundo	0.83	1.00	0.91	10
acerpalmatum	0.93	0.96	0.95	28
acerpensylvanicum	0.91	0.91	0.91	11
acerplatanoides	1.00	0.67	0.80	6
acerpseudoplatanus	0.80	1.00	0.89	4
acerrubrum	0.86	0.86	0.86	14
acersaccharinum	0.85	0.92	0.88	12
acersaccharum	1.00	0.86	0.92	7
aesculusflava	1.00	0.50	0.67	4
aesculusglabra	0.86	0.92	0.89	13
aesculushippocastamon	0.75	1.00	0.86	6
aesculuspavi	1.00	0.97	0.98	30
ailanthusaltissima	1.00	0.40	0.57	5
albiziajulibrissin	1.00	1.00	1.00	8
amelanchierarborea	0.73	0.85	0.79	13
amelanchiercanadensis	0.82	0.82	0.82	11
amelanchierlaevis	1.00	0.89	0.94	9
asiminatriloba	1.00	1.00	1.00	15
betulaalleghaniensis	0.73	0.80	0.76	10
betulajacqemontii	1.00	0.86	0.92	7
betulalenta	1.00	1.00	1.00	3
betulanigra	1.00	1.00	1.00	6
betulapopulifolia	1.00	0.88	0.93	8
broussonettiapapyrifera	1.00	0.88	0.93	16
carpinusbetulus	1.00	1.00	1.00	4
accuracy			0.91	320
macro avg	0.92	0.89	0.89	320
weighted avg	0.92	0.91	0.91	320

Fig 33: Classification report for SVM using ResNet-50 extractor

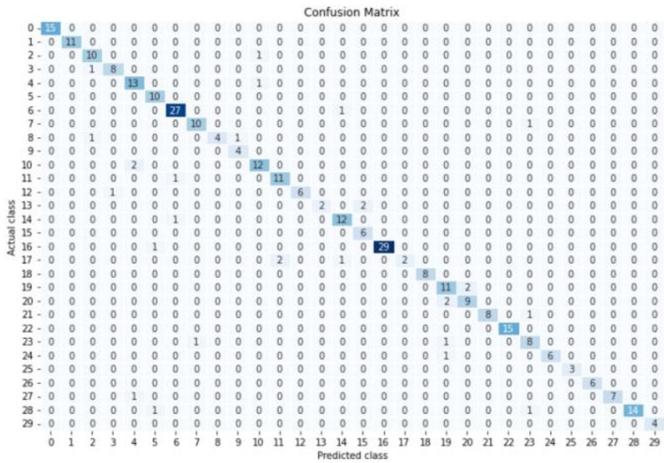


Fig 34: Confusion Matrix for SVM using ResNet-50 extractor

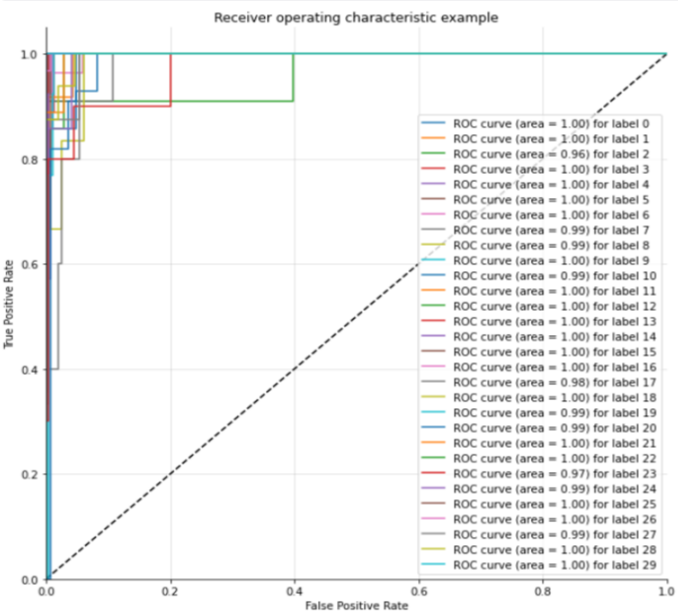


Fig 35: ROC curves for all the labels of SVM using ResNet-50 extractor

VII. CONCLUSION AND FUTURE WORK

Our study shows that image classification of images can be achieved by machine learning models using the state of art algorithms with transfer learning. From our experiments, Logistic Regression with Resnet50 feature extractor and SVM with the VGG-16 feature extractor have outperformed the other algorithms and are able to classify the majority of the images in test set. We noticed that the accuracies of all the models with both feature extractors are nearly equal, may be the embeddings from both models are nearly same.

In future we are planning to experiment with the whole dataset but instead of transfer learning planning to use CNN models for classification and will try to experiment with multiple datasets. Planning to implement the image recognition in other domains as well like detecting Covid severity by chest X-rays, detection of structural damage, etc…

REFERENCES

Neeraj Kumar, P. N. (2012). Leafsnap: A Computer Vision System for Automatic Plant Species Identification Proceedings of the 12th European Conference on Computer Vision (ECCV).

West, Jeremy; Ventura, Dan; Warnick, Sean (2007). "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer". Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on 2007-08-01. Retrieved 2007-08-05.

Muneeb Ul Hassan (2018). https://neurohive.io/en/popular-networks/vgg16/

Andréa Britto Mattos, Ricardo Guimarães Herrmann, Kelly Kiyumi Shigeno, and Rogerio Schmidt Feris. 2014. Flower Classification for a Citizen Science Mobile App. In Proceedings of International Conference on Multimedia Retrieval (ICMR '14). Association for Computing Machinery, New York, NY, USA, 532–533.DOI: https://doi-org.proxy1.lib.uwo.ca/10.1145/2578726.2582620

Marcondes Coelho Feitoza, Wanderson Bezerra da Silva, and Rodrigo Tripodi Calumby. 2019. Exploring Deep Features and Transfer Learning for Plant Species Recognition. In Proceedings of the XV Brazilian Symposium on Information Systems (SBSI'19). Association for Computing Machinery, New York, NY, USA, Article 54, 1–8. DOI: https://doi-org.proxy1.lib.uwo.ca/10.1145/3330204.3330264

Ayan Chaudhury and John L. Barron. 2020. Plant Species Identification from Occluded Leaf Images. IEEE/ACM Trans. Comput. Biol. Bioinformatics 17, 3 (May-June 2020), 1042–1055. DOI: https://doi-org.proxy1.lib.uwo.ca/10.1109/TCBB.2018.2873611.

Chang, A. C. (2020). Chapter 5 - Machine and Deep Learning. In A. C. Chang (Ed.), *Intelligence-Based Medicine* (pp. 67–140). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-823337-5.00005-6

Kaiming He, X. Z. (2015). Deep Residual Learning for Image Recognition. *ImageNet Large Scale Visual Recognition Challenge*