PINTEREST IMAGE TAGGING USING LIME ANALYSIS

PROJECT REPORT

DS - 8013, DEEP LEARNING

Master of Science (MSc)
in
Data Science & Analytics

Winter 2020

By
Ankit Dhall - 500942040

Yeates School of Graduate Studies

Ryerson University

Toronto, ON - M5B 2K3

INDEX

LIST OF FIGURES

LIST OF TABLES

1. DESCRIPTION OF THE PROBLEM

Social Media refers to the platform or technologies that facilitate the creation or sharing of information, ideas, career interests, and other forms of expression via virtual communities and networks. Some famous examples may include Facebook, Instagram, Pinterest, Reddit, LinkedIn and more. Social media content may have no bounds and can range from just a few lines of text, high-resolution images, and videos, or even audio tracks. This data is directly linked to the people who upload, share, and interact with them. This information, used with the right conduct can be of immense use to a number of industries and can help in increasing brand awareness, increase website traffic, boosting sales, etc.

Artificial Intelligence, driven by Deep Learning is at the pinnacle of technology. They have helped us investigate and solve many real-world problems that were previously too hard to tackle or too time-consuming for us to hard code solutions for. One such industry, that implements these technologies at their core, is the social media industry.

With the social media being loaded with millions of images, and hundreds and thousands more being uploaded every single day, tagging these images could be of pivotal importance right as they are being uploaded. This would help us achieve a much more organized data system and can also help in many applications such as user recommendations, virtual profile building, sentiment analysis, customer and audience engagement, and general analytics.

Fig 1. Pinterest Logo

Pinterest is a visual discovery engine for finding ideas like recipes, home and style inspiration, and more. Users are able to upload and save either their own images or images that have been uploaded by other users on the network. The platform allows users to create ‘boards’ where they are able to pin ideas that they might come across. Pins are ideas that people on Pinterest create, find, and save from around the web.

When a user uploads or adds images to their boards, they can organize their pins among subcategories. Our project aims to automatically classify these images into

relevant categories and recommend these categories to users to enable the easy and efficient organization of their ideas.
For example, if a user wants to collect ideas for a dinner party, they might pin images of different cuisines, say Indian food and Italian food. While they may pin them under separate boards, they essentially belong to the same category, i.e. food. By recognizing this, we can suggest to the user the most suitable category so as to streamline and categorize pins and boards in a cleaner way.

The data would be collected directly from Pinterest using Pinterest API and will automatically follow the robots exclusion protocol.
For the scope of this project, we would be focusing mainly on 8 of the most popular sub-categories found on Pinterest and will be trying to build a suitable model around this data.

A variety of Image Classification algorithms would be used to try and achieve promising results on the collected data and will involve the use of Deep Learning using Convolutional Neural Networks and Transfer Learning. These approaches would then be compared to traditional primitive approaches to solving image classification tasks like Artificial Neural Networks (without convolutional layers) as well as machine learning algorithms like Support Vector Machines (SVMs)

The results for the analysis and testing conducted would be compared and a suitable model will be developed. The primary metric that would be used in the project would be ‘Accuracy’ and this would be the primary metric that we will be looking to maximize.

In addition to developing the models, we will also try to look into what a particular model has learned using model explanation techniques such as LIME (Local Interpretable Model-agnostic Explanations). These techniques will help us to gain an explanation as to “why” our model is working the way it is and whether it is actually able to detect features that a model is expected to. In turn, this will enable us to build trust in the model before being deployed in a real-world scenario.

2. DATA SOURCES

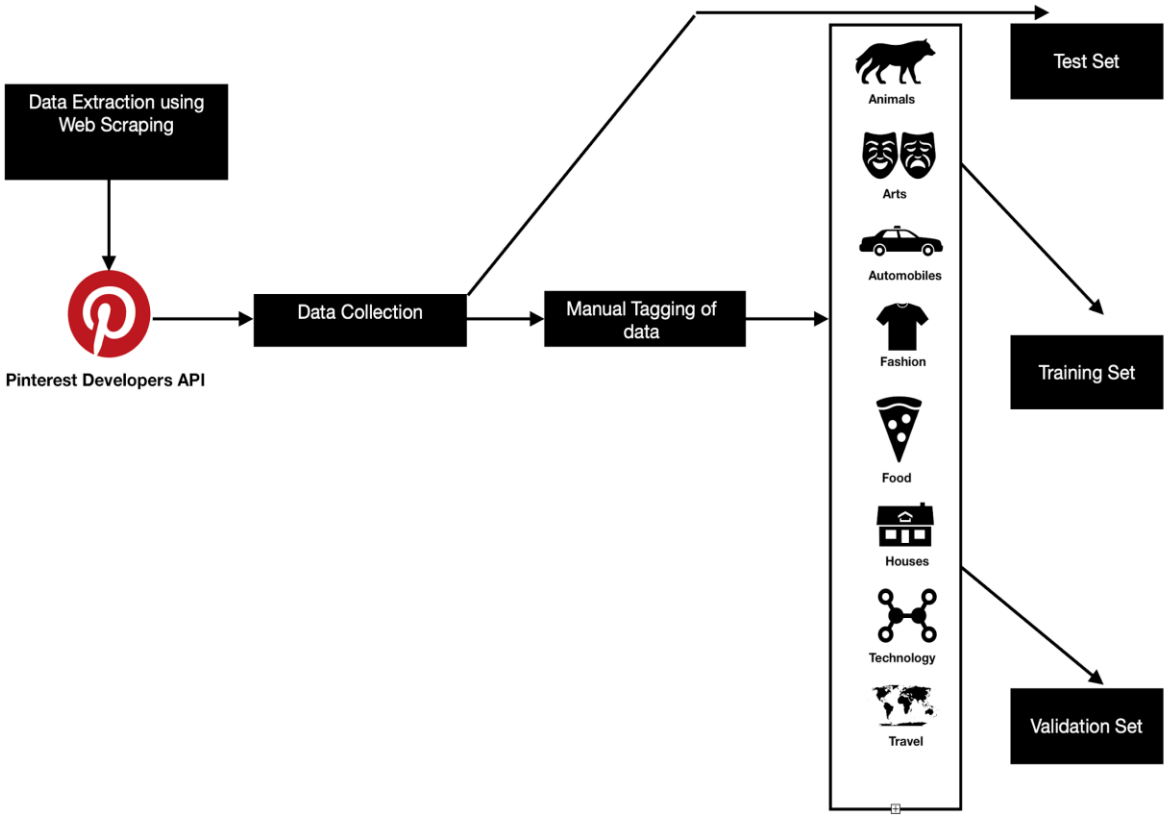


Fig 2. Showing Data Sources and Data Collection from Pinterest Developer API

Images of 8 of the most popular categories were targeted and collected using the Pinterest Developers API.

Fig 3. Data Organization Structure

This was then followed by a manual tagging process and was organized in a file structure (shown below in Fig. 2) which is supported by the Keras 'ImageDataGenerator' class. This dataset sourced the training set as well as the validation set. The 'ImageDataGenerator' class enables us to load data into our model while training more efficiently without us manually specifying which image belongs to which class.

Parallelly, data belonging to the same 8 classes was randomly extracted using the API to make an unseen distribution for the test set.

3. LITERATURE REVIEW

The ability to share photos, opinions, events, etc. in real-time has transformed the way we live and, also, the way we do business. Social Media is the new home for people where they feel very accommodated and welcomed with the respectively connected people. Platforms such as Facebook, twitter, YouTube, WhatsApp creates a humongous amount of data every day. All these big social media giants try their best to provide smooth customer experience and to accompany these platforms, 10 years back, Pinterest, a popular social curation service where people collect, organize, and share content (pins in Pinterest), has gained great attention.

Pinterest is a pinboard-style content sharing platform that allows users to exhibit collections of images, posters, and videos. Since image content is predominant in Pinterest which can lead to great research prospects with image classification problems.

This section describes various recent literature related to Pinterest and image classification using deep neural networks.

Paper 1:
Collecting, Organizing, and Sharing Pins in Pinterest: Interest-Driven or Social-Driven?

Over a period of 10 years, Pinterest has become such a huge success with currently having 335 million + users with a market value of $13.7 billion [2]. Han, J. and his team tried to analyze this huge network of Pinterest with their interesting research questions [1]. In this process of analyzing the Pinterest system, they came up with interesting research questions and intricate analysis which laid a strong foundation of research on Pinterest.

In this paper, they have tried to study how people collect and manage pins by their tastes in Pinterest? What factors do mainly drive people to share their pins on Pinterest? How do the characteristics of users (e.g., gender, popularity, country) or properties of pins (e.g., category, topic) play roles in propagating pins in Pinterest?
According to this research, there are some basic properties associated with Pinterest which are as follows:

- Pinterest is different from all other social networking platforms as there is no direct communication between users. (e.g., private messages on Facebook or Twitter)
- Pinterest basic function is to let users collect, organize and share pins according to their interests.
- Each image is called a pin

- Each pin can be collected (mostly images) that the user finds interesting. Users can create their own boards(categories) and place their pins in these boards.

In terms of analysis, they introduced some complex research questions and answered with comprehensive analysis.

Q 1. How do people collect and manage pins by their tastes? How many interests do people usually have?

Started with a dataset of 3.5 million pins shared by 150 million users. They found that the majority of the users on Pinterest are females which accounts 85% of the total users. The United States is one of the top 5 countries which has the highest number of users. The most riveting thought explained by them in the paper, If a user has 2 boards such as a basketball board and baseball board, both of which belongs to a sports category (as a board can be interpreted as a user defined topic). This thought of organizing user's boards automatically or recommending users what the best category can be for a particular pin, has defined our problem statement.

Q 2. How do pins propagate in Pinterest?

Pins propagated through the process of repining where a user shares or re-pins a particular pin. According to this paper, a small portion of pins have received great attention. They have also analyzed the spreading time of the pin and on average a new pin propagates within 6 hours.

Q 3. Which factors affect pin propagation?

The basic factors which affect pin propagation are pinner's influence and pin's influence itself.

They found that "food & drink" exhibits the shortest inter-re-pin time, which implies that the "food & drink" category is the most active category since pins are posted and spreading quickly.

In summarization of comprehensive analysis on Pinterest, this study demystified the various aspects of Pinterest such as how people collect, manage and share pins in Pinterest. In terms of pin propagation, it solely depends upon the pin's properties like its topic or content not by the user's characteristics like her number of followers. In terms of Pinterest categories, they have defined 32 independent categories for the users although the user's board category is user defined and it depends on how a user interprets a pin or interest of users. Their empirically grounded simulation demonstrated that the properties of pins are more important factors than those of users for accurately predicting pin consumption patterns in Pinterest.

Blog 1: Image classification: A comparison of DNN, CNN and Transfer Learning Approach

This article explains the various architectures used in deep learning literature to classify the images or image tagging with multiple class problems. It compares the basic features involved in all the configurations and describes how we can initialize these models in python using TensorFlow.

Model	Configurations	Val_accuracy	Training Time	Findings
DNN: Deep Neural Networks	3 layers with number of nodes as [128,64,1] and used “ReLu” in hidden layers and tanh in output layers	50%	6.3 hrs.	-Slow in convergence -Low validation accuracy -Input features are not fully accommodated, need to reduce dimensions
CNN: Convolutional Neural Networks	3 convolution layers with no of filters as [32,32,64] with filter size of 3*3 in each layer.	80%	5 hrs.	-Fast in Convergence -High validation accuracy -Considers the complex image features
Transfer Learning	Used VGG 16	92%	12 mins	-Considerable decrease in execution time -Fits the problem really well

Table 1. Comparative Study of CNN Architectures

The article also explains the use of transfer learning can significantly decrease the difficulty level of the problem. According to the author, Transfer learning is a method of reusing the already acquired knowledge. The idea is to use a state of the art model which is already trained on a larger dataset for a long time and proven to work well in related tasks [3]. We can use transfer learning in two conditions.

1. **Direct Application:** when you can directly relate your problem with the given transfer learning architecture and need to preprocess the input according to model and then feed it to model to get results.
2. **Representation Learning:** Assess that the pretrained model may not be directly applicable to our problem. however, we can use it to get a useful representation of our input data or use the pretrained layers of convolution to get the intermediate results and then apply your own model and integrate with existing models.

Blog 2: CNN Architectures: VGG, ResNet, Inception + TL

To gain more deeper insights into the famous CNN architectures, This article talks about CNN architectures such as VGG 16, VGG 19, Inception Net, ResNet, and Xception Nets. It also illustrates Image feature extraction and transfer learning.

VGG 16

- One of the simplest networks used in ImageNet competitions which was published in 2014.
- Includes 16 layers with 13 layers of convolution and 3 Dense layers for classification.
- max pooling layers applied at different steps in the architecture
- In terms of disadvantages, it is slow to train and produces a model of very large size
- Author has also explained the use of VGG 16 with gold mine Image Net data set.

VGG 19

- VGG19 is a similar model architecture as VGG16 with three additional convolutional layers, it consists of a total of 16 Convolution layers and 3 dense layers.
- the use of 3 x 3 convolutions with stride 1 gives an effective receptive field equivalent to 7 * 7. This means there are fewer parameters to train.

Inception Nets

- Fundamental blocks of Inception Nets is Inception modules which is a kind of network topology. The idea of the inception module is to use different filter sizes to achieve multi-level feature extraction.
- To reduce the dimensionality 1*1 filter is used in the inception module.
- With the output layer to predict the class of a given input. It also has auxiliary output layers corresponding to the inception module. Since it is a very large network so users can see the intermediate outputs using these auxiliary nodes.

Xception Nets

- Xception is an extension of the Inception architecture which replaces the standard Inception modules with depth wise separable convolutions.

Transfer learning

The author has tried to explain how we can do image feature extraction and predictions through transfer learning. Load the weights of a pretrained model and do not use the last layer of pre trained model so that you can design your own machine learning classifier. After extracting features, flatten the features and feed into a machine learning or deep learning classifier (MLP) to classify the input data.

Paper - 2: “Why Should I Trust You?” Explaining the Predictions of Any Classifier

The process of machine learning is to learn and improve from the experience without being explicitly programmed. It focuses on the development of computer programs that can access data and use it to learn the meaningful patterns for future predictions. Despite the power of artificial intelligence and machine learning, it is still considered a black box. Nobody tries to explain how these powerful algorithms work and give these predictions. The problem of trust also comes into picture because if a user does not trust a model or a prediction and does not know about the basis on which the prediction comes out, he will not be going to use the model.
In 2016 Marco and his team came up with this novel approach of explaining the predictions of any classifier [5].

This research introduces two concepts which are as follows:

1. **LIME - (Local Interpretable Model-Agnostic Explanations)** : an algorithm that can explain the predictions of any classifier and regressor in a faithful way.
2. **SP-LIME - (Submodular Pick for Explaining Models)** : a method that selects a set of instances which can answer the question of “why this prediction?”. It basically gives you extra parameters which can be used to make a judgement on prediction.

According to this paper, there is a strict need of evaluate the model as a whole before deploying into “in the wild”. Currently, models are evaluated using scientific metrics such as F1_scores and accuracy on validation data sets. However, real world data is different and further the evaluation metrics might not be the appropriate indicator of predictions. In these cases, it is good to have supporting instances or some representations with predictions which can show users some support for a particular prediction.

In order to provide a definition of “explaining a prediction”, This paper shows that one can explain a prediction by presenting some textual or visual artifacts that shows a qualitative understanding of the relationship between input and prediction. For instance, A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient’s history that led to the prediction. **“Sneeze”** and **“headache”** are portrayed as contributing to the “flu” prediction, while **“no fatigue”** is evidence against it. With these, a doctor can make an informed decision about whether to trust the model’s prediction.

To make good explainers, essential traits can be interpretation and local fidelity. An explainer should be interpretable, i.e., provide a clear understanding between the input and prediction. However interpretability is highly dependent on target audience because a ML expert can understand weights, Bayesian networks or mathematical equations but user can only understand some qualitative features which should be in a human readable language.

Another essential criterion is local fidelity which explains about how the model behaves in the locality of the instance being predicted. To achieve global fidelity for an explainer is a difficult task however it can show some local features which are faithful to the local instances.

Working of LIME with Respect to Image Classification:

This research touches all the aspects of an explaining system and how different classifiers and regressors can implement LIME to give more credibility to the models. However, for this project we are focusing mainly on image classification models. For an image classification system, an interpretable representation can be a binary vector indicating the presence or absence of a contiguous patch of similar pixels.

In this paper, they have used sparse linear explanations for image classifiers. Fig 3. Shows the working of lime analysis on inception neural networks. The image can be classified into these 3 categories with the explanations highlighted corresponding to each class. For example, why acoustic guitar was predicted to be electric because of fretboard presence in the image.

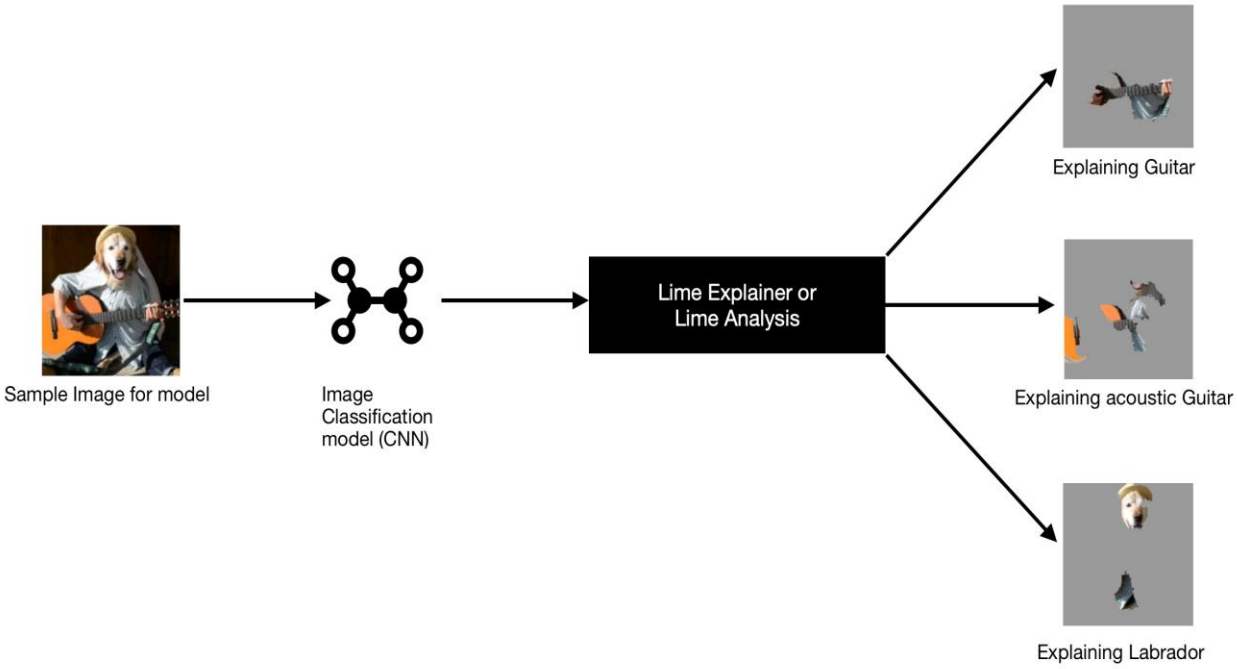


Fig 4. Sample Image and Explainer's Results on the Image

With these kinds of explainers, they have also claimed that feature engineering and active learning can be significantly improved as explanations can aid in the process of presenting important features, particularly for removing features that the users feel do not generalize.
We can also do the active learning on complicated input instances if there is a contradiction between user and model’s prediction.

In this project, our aim was to tag the Pinterest images for users so that it can help them in organizing their personal boards in a more standardized way. In order to build the trust of users and get insights into the prediction, we have also applied LIME analysis in the project and detailed discussion on this process provided in the Implementation section.

4. TECHNOLOGIES USED + CHALLENGES FACED

TECHNOLOGIES USED

The problem was tackled using 3 major technological fronts:

- Machine Learning (SVM)
- Artificial Neural Networks
- Convolutional Neural Networks

As a part of the convolutional neural networks approach, the **"Transfer Learning"** approach was used in addition to a basic CNN model built from scratch.
All of the transfer learning approaches were chosen and compared with respect to their performance on the basis of the model accuracy as well as their speed of prediction and compute efficiency.

Transfer Learning Approaches Used:

- MobileNet
- VGG – 16
- ResNet – 50
- Inception V3

Parameter tuning was performed on all four of the transfer learning approaches to be able to find the best possible model using this approach.

The use of primitive techniques for image related tasks, like SVM and Artificial Neural Networks posed several challenges in terms of their performance and training times.

In addition, certain transfer learning models were difficult to train due to the restrictions in compute resources available at hand. Image related tasks require a large amount of compute power, including GPUs and high amounts of RAM to be able to simultaneously load, preprocess, and feed training data into the model.

Training of all the models were carried out on an average laptop with an NVIDIA 940MX GPU, and 8GB of RAM. Even with the availability of a GPU, these specifications are very limited in terms of their compute capabilities for a heavy image classification task.
Hence, the use of the transfer learning approach also drastically decreased the amount of resources and training time required to achieve good results.

CHALLENGES FACED:

Model Architecture Used	Challenges Faced
Machine Learning (SVM)	● Not able to feed original dimensions of images ● RGB images took more than 6 hrs. to train with reduced dimensions. ● Converted all RGB images into grayscale which results in loss of color related features. ● Radial kernels were also not able to classify the images properly. ● Average running time with gray scale images was very long (approximately 50 minutes) ● Used 32*32 images as an input to SVM classifier which results in major feature loss in terms of image characteristics.
Artificial Neural Network	● With an input image size of 160 x 160, and 5 following hidden layers, the total trainable parameter count exceeded 1 billion and was close to impossible to train on local machines. ● Even with a reduced input image size of 56 x 56, and a total parameter count of about 33 million, training took about 40 minutes even by using TensorFlow GPU. ● Due to the lack of spatial information being captured by the network, the maximum accuracy obtained on the test set was about 44.75%
Basic CNN Model	● While the total number of trainable parameters were about 6.5 million and training time was only 12 minutes, the model overfit the training data obtaining an accuracy of 94.4% and failed to achieve a good generalization accuracy with an accuracy of only 69.1% on the test set. ● The same was noticed even after adding a few regularization techniques like dropout layers.
Inception V3 (Transfer Learning)	● While performing parameter tuning on the Inception model, we constantly received the Out of Memory error on our system while producing the Keras scratch graph with a batch size of 32.

Table 2. Challenges Faced with Different Architectures

5. IMPLEMENTATION

This section will explain about the detailed process of implementation. For the comprehensive analysis we have divided our projects into different phases which can be seen from the fig 4.

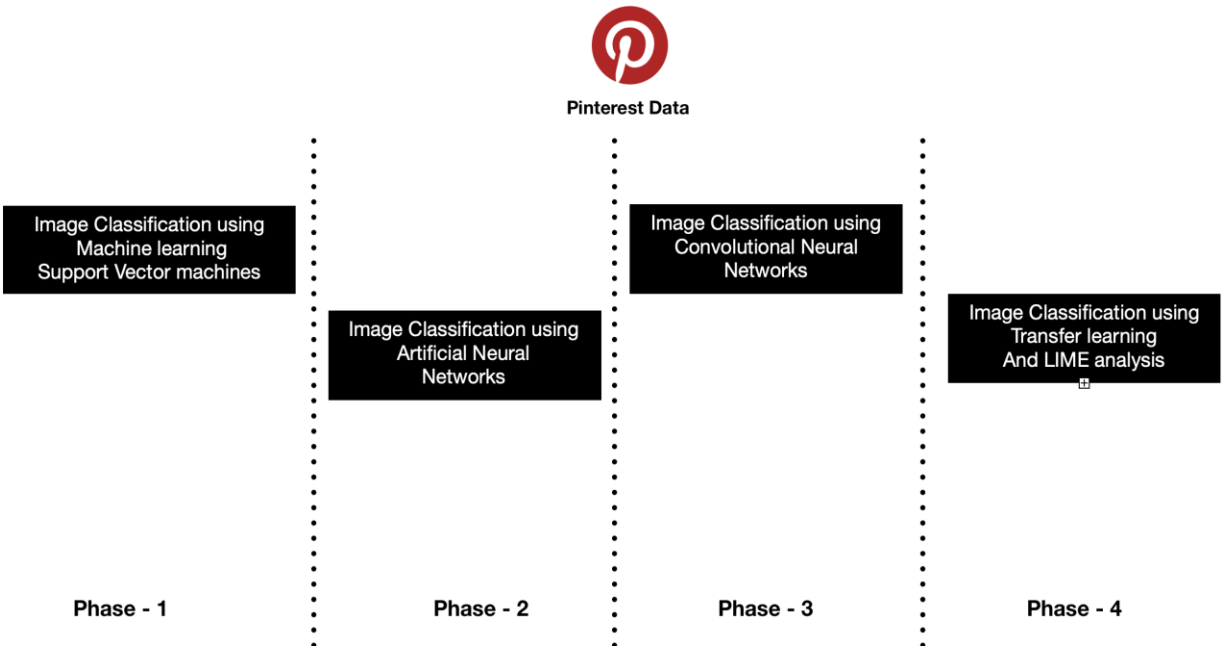


Fig 5. Different Implementation Phases

DATA PRE-PROCESSING

1. **Image Classification using ML: Support Vector Machines**
   - Transformed input image to 32 x 32 to be able to fit in the support vector machine algorithm
   - Converted each image from RGB to grayscale to reduce the execution time
   - Flatten the gray scale images to vectors

2. **Image Classification using: ANN**
   - Transformed input image size to 56 x 56 to be able to train neural network with a manageable number of parameters
   - Flattened colour images from 56 x 56 x 3 to vectors of shape 9,408
   - Normalized input vector (Divided pixel values by 255)

3. **Image Classification using: CNN**
   - Transformed input image size to 56 x 56
   - Normalized input images (Divided pixel values by 255)

4. Transfer Learning and LIME Analysis

Transfer learning models that were implemented using the Keras Applications have their pre-defined preprocessing functions depending on the particular transfer learning model being used.
These preprocessing transformations are ones that have proven to be most useful in the training of the transfer learning models.

ARCHITECTURE & DESIGN

This section will describe all the architectures used in the project to make the predictions. Since this project involves many techniques and architectures, each architecture is shown with the help of flow diagrams in detail with their corresponding layers.

1. Support Vector Machines:

Fig 6. Support Vector Machine Architecture

2. Artificial Neural Networks:

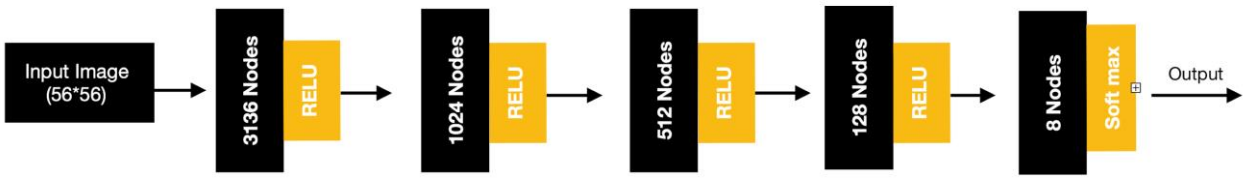


Fig 7. Artificial Neural Network Architecture

3. Convolutional Neural Networks:

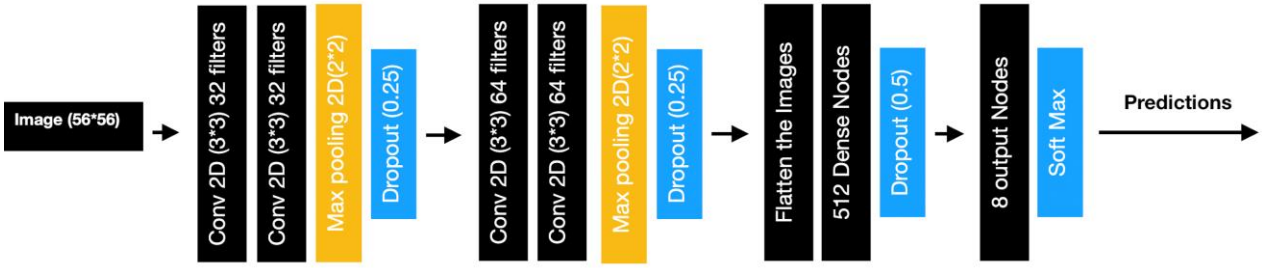


Fig 8. Basic Convolutional Neural Network Architecture

4. Transfer Learning: (MobileNet, VGG - 16, ResNet - 50, Inception V3)

For the “Transfer Learning” approach, we decided to use the pre-trained weights for the ‘ImageNet’ dataset as a starting point for training on our dataset.

Since our image dataset was comparatively much smaller and very different from the ImageNet dataset that the models have been originally trained on, we decided to freeze training of the layers in the base model.
On top of this base model, we added a few extra fully connected layers of our own as well as an output layer containing eight output nodes, each representing one of the eight classes in our image classification task.

Only the weights of the newly added fully connected layers and output layers were updated in the training process whereas the weights of the base layer were kept frozen throughout the training process.

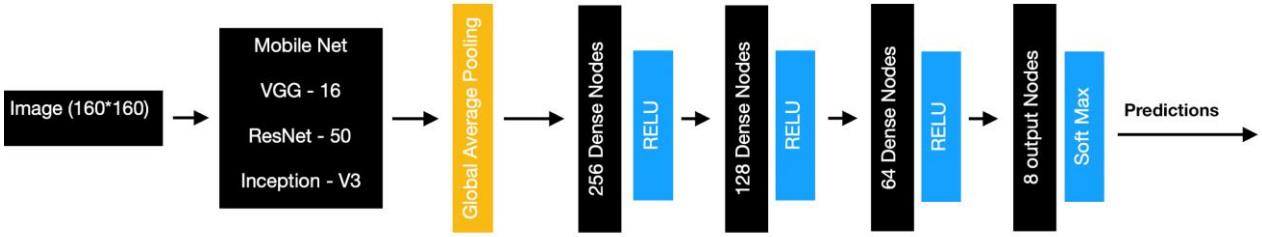


Fig 9. Transfer Learning Model Architectures

5. LIME Analysis:

Once our models are trained and tested on the test set, we can analyze the working of our model using LIME Analysis. LIME stands for Local Interpretable Model-agnostic Explanations and helps us understand what our model is learning and helps us

develop trust in our models. It increases the explainability in an otherwise black-box approach to image classification.
LIME can help us in comprehending our model better by developing masks on the image for the most important features that the model learns. These masks can then help us make an informed decision on whether we think our model is recognizing the most important features in a particular image.

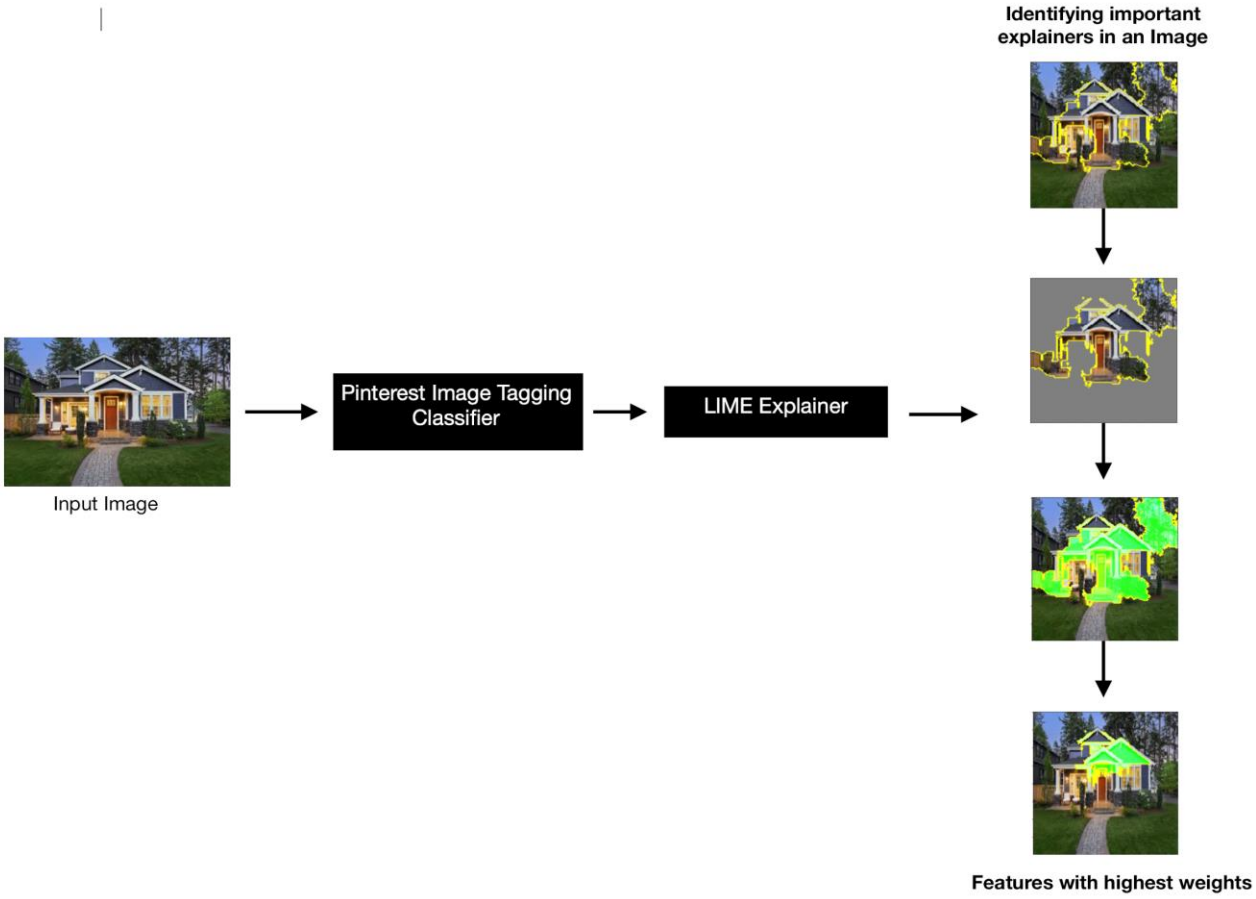


Fig 10. LIME Analysis Architecture

6. RESULTS

This section will aim to explain the experimental study, results obtained, and, the comparative study of different architectures used in the project.

EXPERIMENTAL STUDY

In this subsection, for each model, the accuracy obtained on the validation and training sets using line charts has been shown. Further, the hyper parameter tuning for transfer learning models has been presented in a tabular format.

The following were the observed Loss/Accuracy graphs obtained for the Artificial Neural Network & Basic CNN Architecture approaches:

Artificial Neural Network:

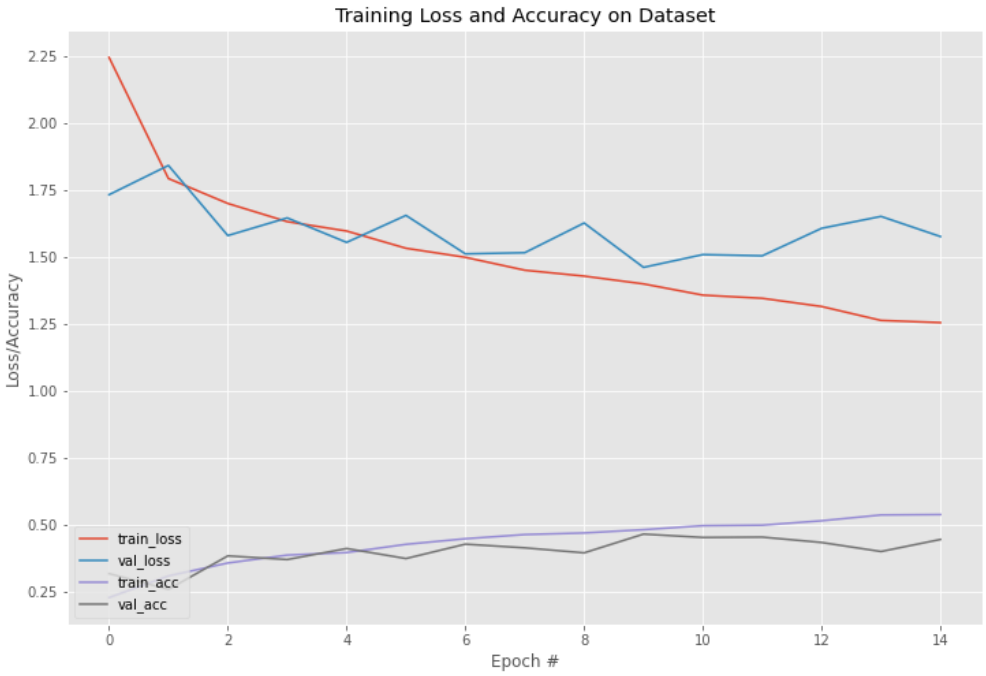


Fig 11. ANN Loss & Accuracy Graph

Final Training Accuracy: **54.05%**
Final Validation Accuracy: **44.75%**

Basic Convolutional Neural Network:

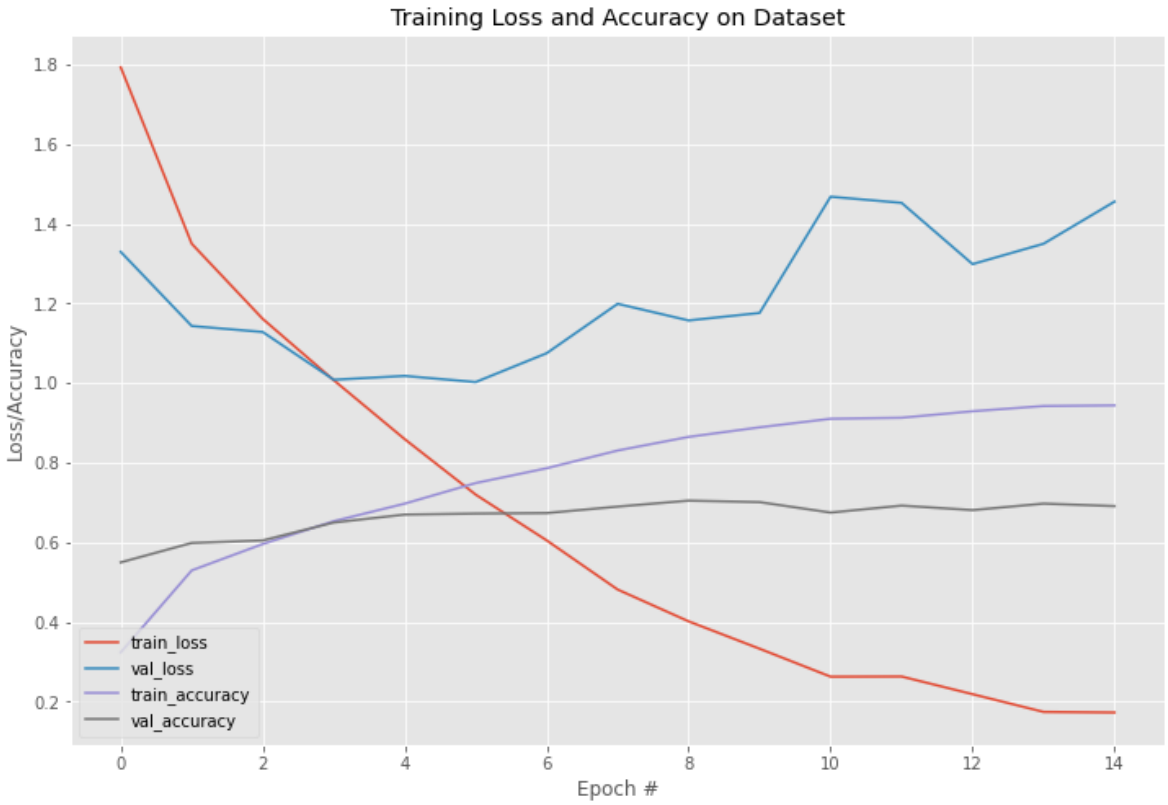


Fig 12. Basic CNN Loss & Accuracy Graph

Final Training Accuracy: **94.41%**
Final Validation Accuracy: **69.13%**

HYPERPARAMETER TUNING (Support Vector Machine)

For the machine learning approach (Support Vector Machine), we decided to experiment with hyper parameter tuning and the best configuration of parameters are as follows:

- **Kernel function: radial (as it is a multi-class problem)**
- **Best value of C = 2, gamma = 0.01**

S. No	C	gamma	Accuracy
1.	1	0.1	37.20%
2.	1	0.01	39.80%
3.	**2**	**0.01**	**41.53%**

Table 3. SVM Hyperparameter Tuning

HYPERPARAMETER TUNING (Transfer Learning Models)

Since our main approach towards tackling the “Pinterest Image Tagging” problem is the “Transfer Learning” approach, we decided to carry out basic hyperparameter tuning for the 4 base models we considered for the project. This included trying to play with parameters such as learning rate and batch size.

Since training of such deep learning models can be quite expensive in terms of computer resources and time taken, we decided to use certain parameter choices that are known to give the best results on such image classification tasks.

Fixed Parameter Choices:

- **Optimizer: Adam, beta_1 = 0.9, beta_2 = 0.999**
- **Activation Functions for Added Hidden Units: ReLu**
- **Activation Function for Output Layer: Softmax**
- **Loss Function: Categorical Cross-Entropy**

The following is the detailed analysis of the hyperparameter tuning for each of the 4 transfer learning models that was carried out. Each of the resultant models were then tested on the test set.

Further, the best configurations for each of the 4 transfer learning models were then taken ahead for a further detailed analysis in terms of their class-wise prediction accuracies and average prediction time per image.

MobileNet:

S. No	Batch Size	Learning Rate	Best Validation Accuracy	Test Set Accuracy
1	16	0.001	90.84%	90.25%
2	16	0.0001	90.75%	89.25%
3	32	0.001	90.25%	90.375%
4	**32**	**0.0001**	**91.37%**	**92.0%**

Table 4. MobileNet Hyperparameter Tuning

Loss & Accuracy for Best MobileNet Model

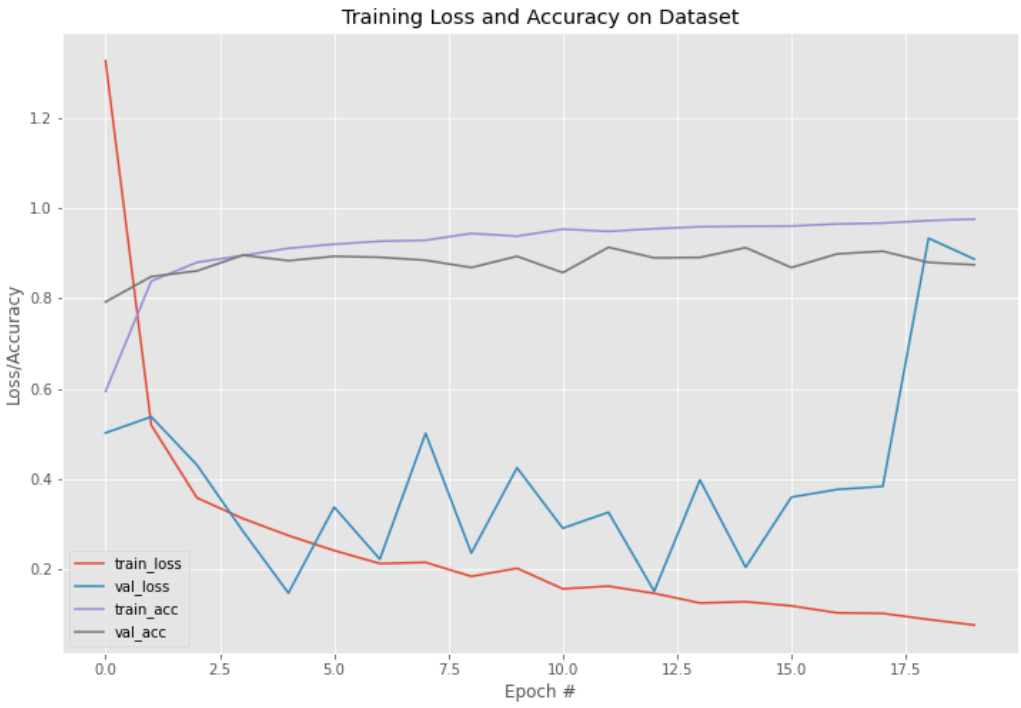


Fig 13. MobileNet Loss & Accuracy Graph

VGG-16:

S. No	Batch Size	Learning Rate	Best Validation Accuracy	Test Set Accuracy
1	**16**	**0.001**	**92.25%**	**91.5%**
2	16	0.0001	89.31%	89.875%
3	32	0.001	89.88%	90.375%
4	32	0.0001	87.52%	89.625%

Table 5. VGG-16 Hyperparameter Tuning

Loss & Accuracy for Best VGG-16 Model

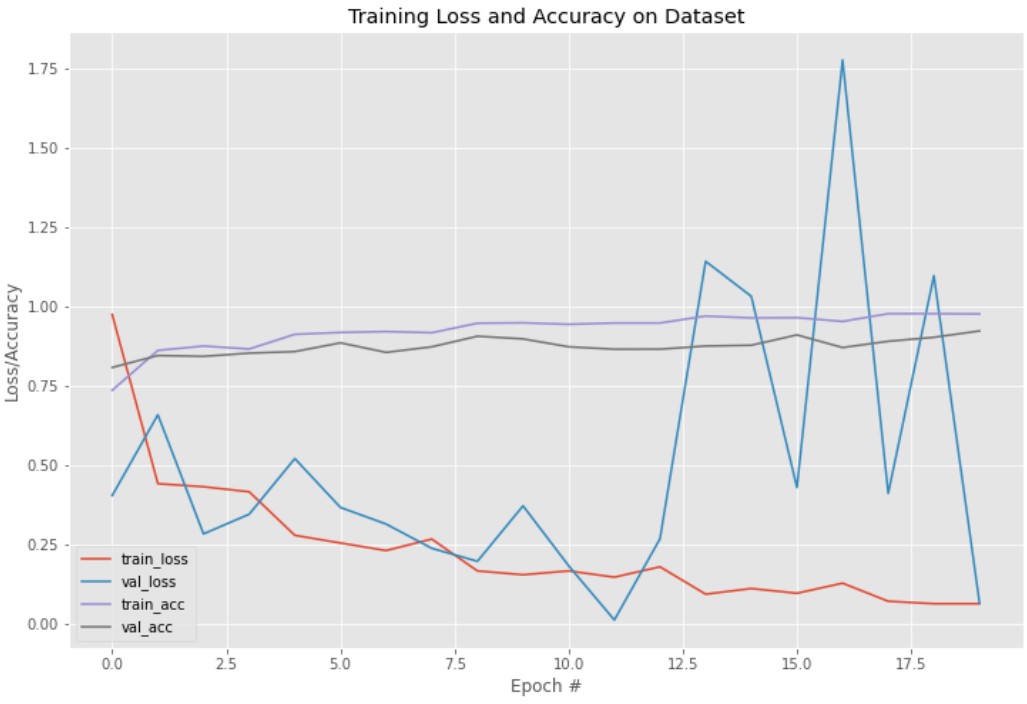


Fig 14. VGG-16 Loss & Accuracy Graph

ResNet – 50

S. No	Batch Size	Learning Rate	Best Validation Accuracy	Test Set Accuracy
1	**16**	**0.001**	**92.50%**	**91.375%**
2	16	0.0001	91.75%	90.375%
3	32	0.001	89.75%	90.5%
4	32	0.0001	91.55%	91.25%

Table 6. ResNet-50 Hyperparameter Tuning

Loss & Accuracy for Best ResNet-50 Model

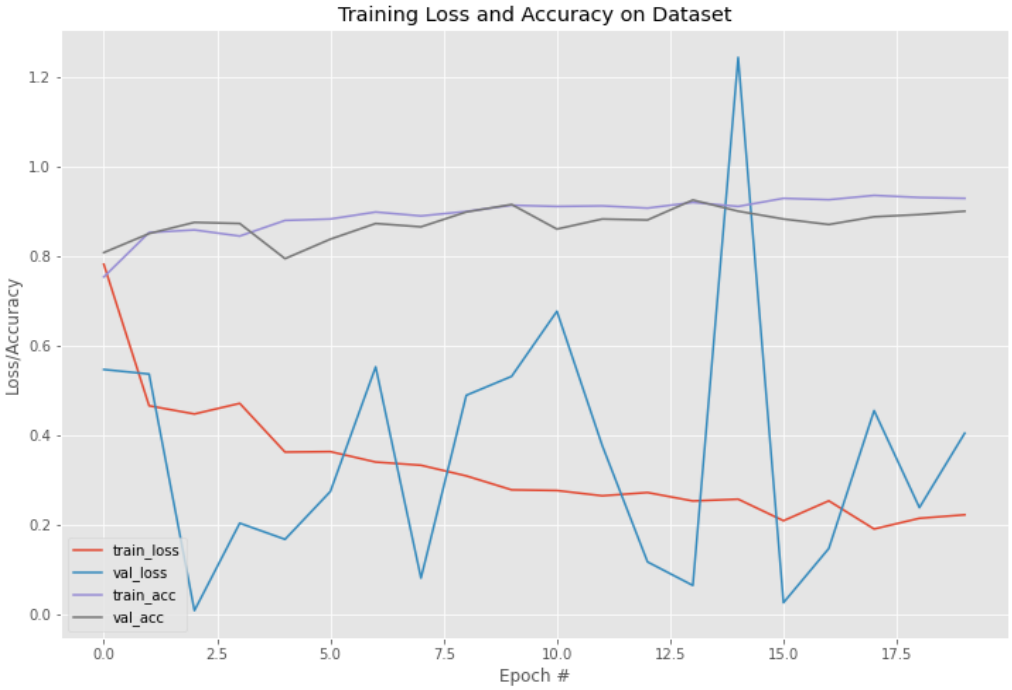


Fig 15. ResNet-50 Loss & Accuracy Graph

Inception-V3

S. No	Batch Size	Learning Rate	Best Validation Accuracy	Test Set Accuracy
1	16	0.001	83.97%	79.25%
2	**16**	**0.0001**	**84.00%**	**80.625%**
3	32	0.001	OOM Error	N/A
4	32	0.0001	OOM Error	N/A

Table 7. Inception-V3 Hyperparameter Tuning

Loss & Accuracy for Best Inception-V3 Model

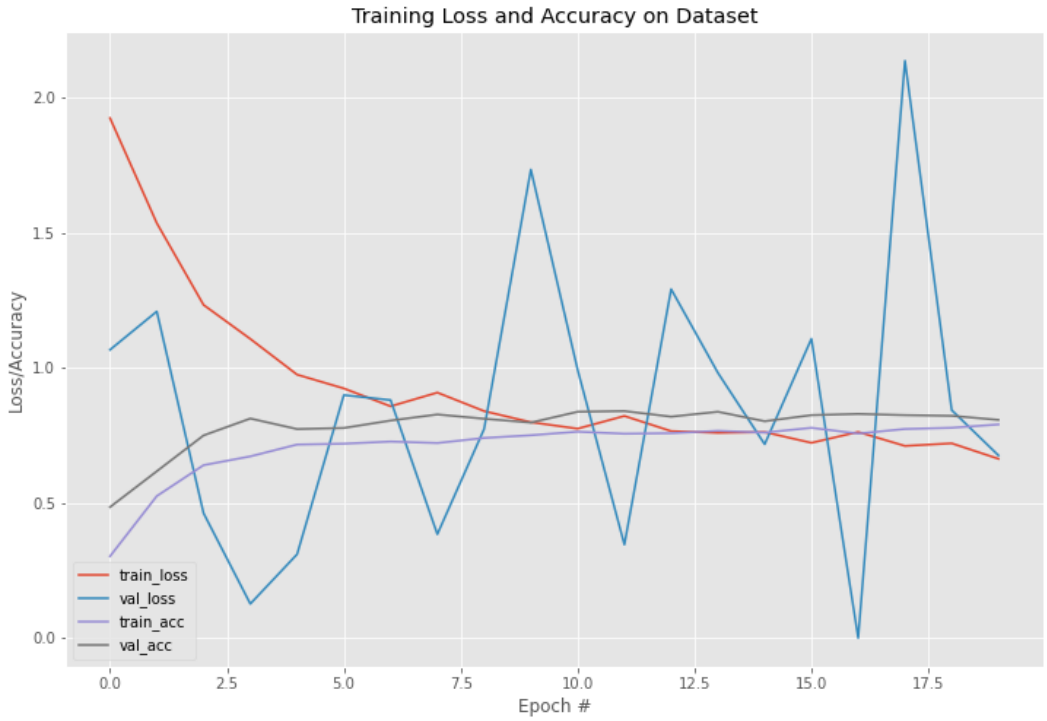


Fig 16. Inception-V3 Loss & Accuracy Graph

LIME ANALYSIS (Transfer Learning Models):

To understand the working of the trained models and gauging the explainability factor, we analyze a few images using LIME and the following are the observed results:

Original Image: (Houses)

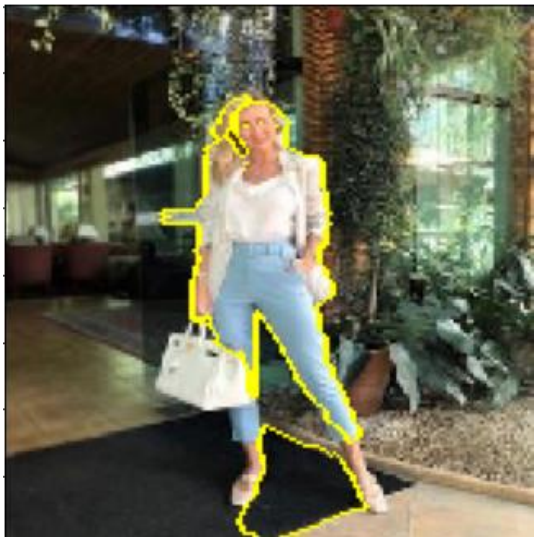
Mask for 5 Most Important Features Detected:

Fig 17. LIME Working on ‘Houses’ Class

We can clearly see that our model detects features of the image like the arched roofs and the front entrance that are features we as humans would also pay the most attention to.

Original Image: (Fashion)

Mask for 5 Most Important Features Detected:



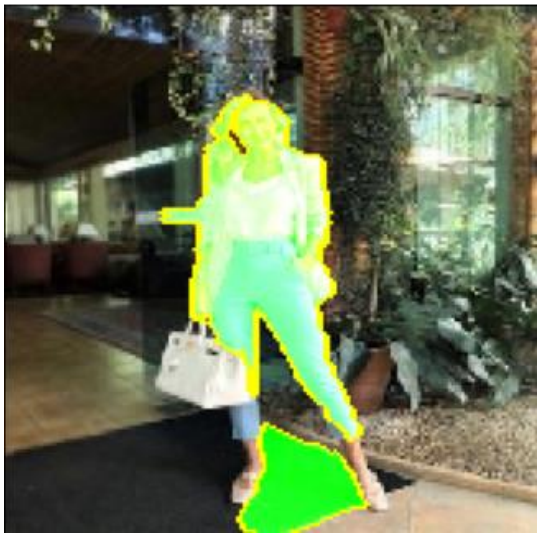


Fig 18. LIME Working on ‘Fashion’ Class

In the above image we can see that the entire outfit is being masked and they are thus the most important features being recognized by the model, thus classifying the picture as ‘fashion’.

CLASS-WISE ACCURACY:

This section describes the overall performance of each model and technique using a confusion matrix of accuracy for each class.

Machine Learning Approach - Support Vector Machines:

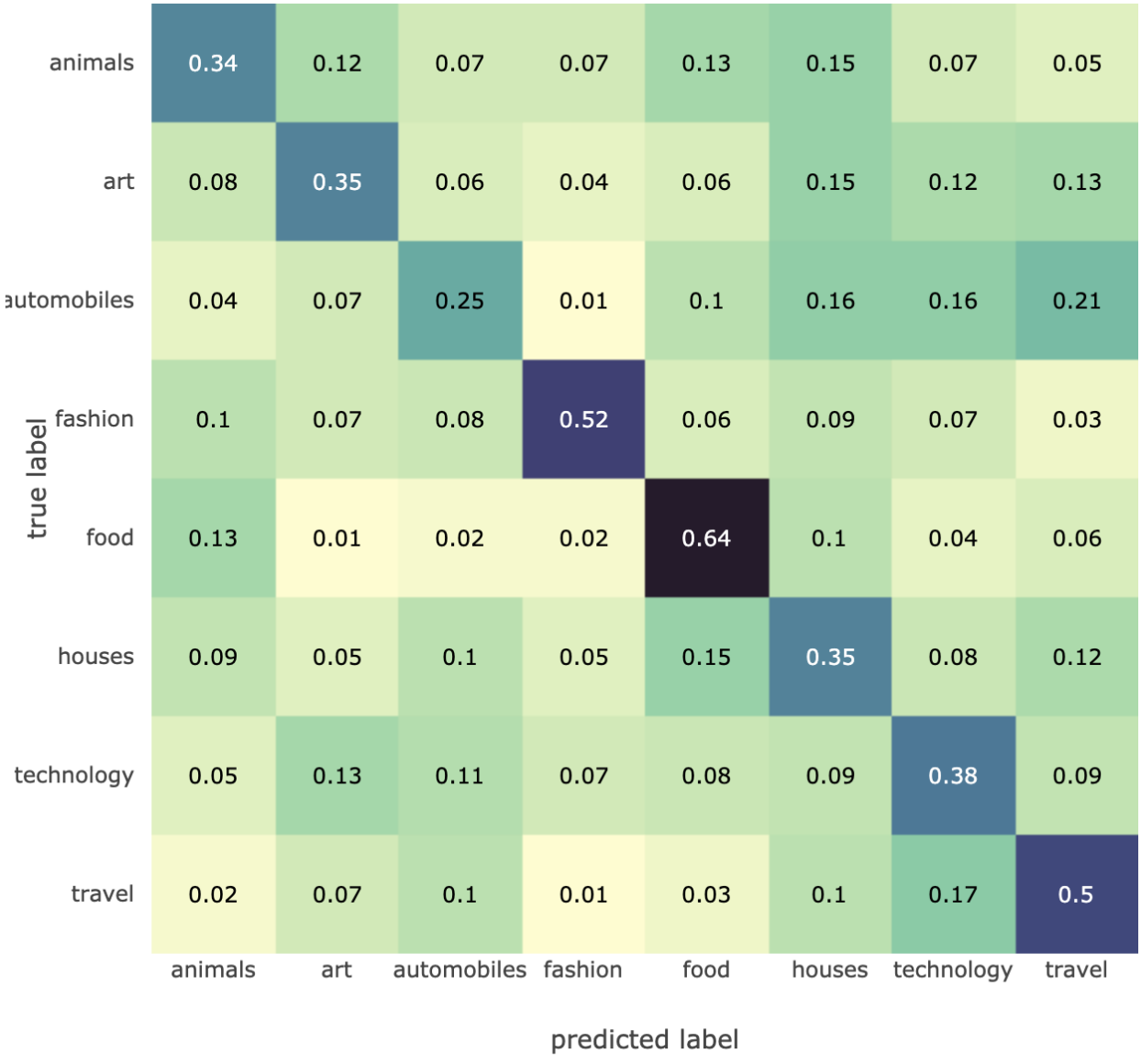


Fig 19. SVM Confusion Matrix

Average SVM Model Accuracy on Test Set = 41.625%

Artificial Neural Network:

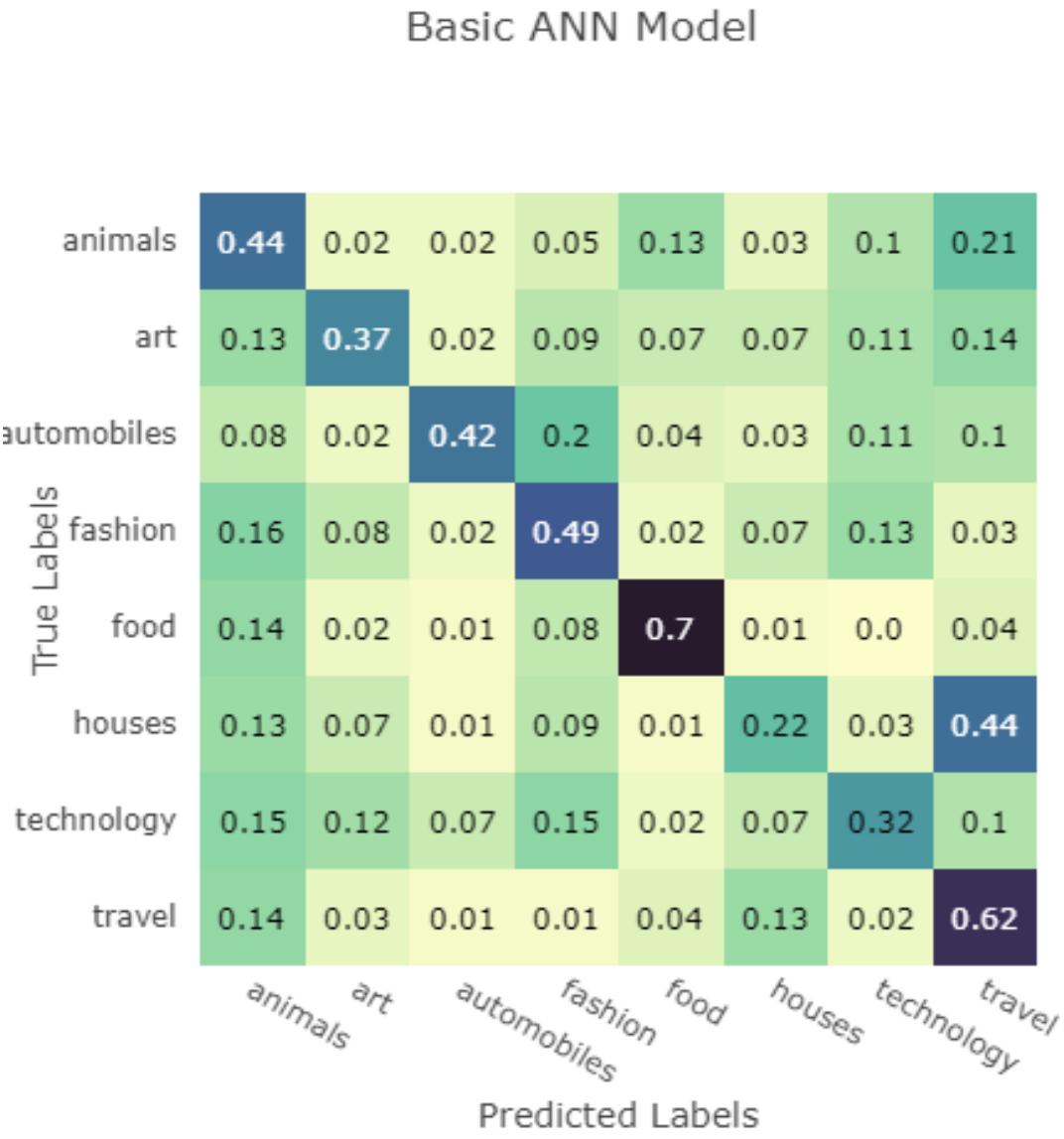


Fig 20. ANN Confusion Matrix

Average ANN Model Accuracy on Test Set = 44.75%

Convolutional Neural Network:

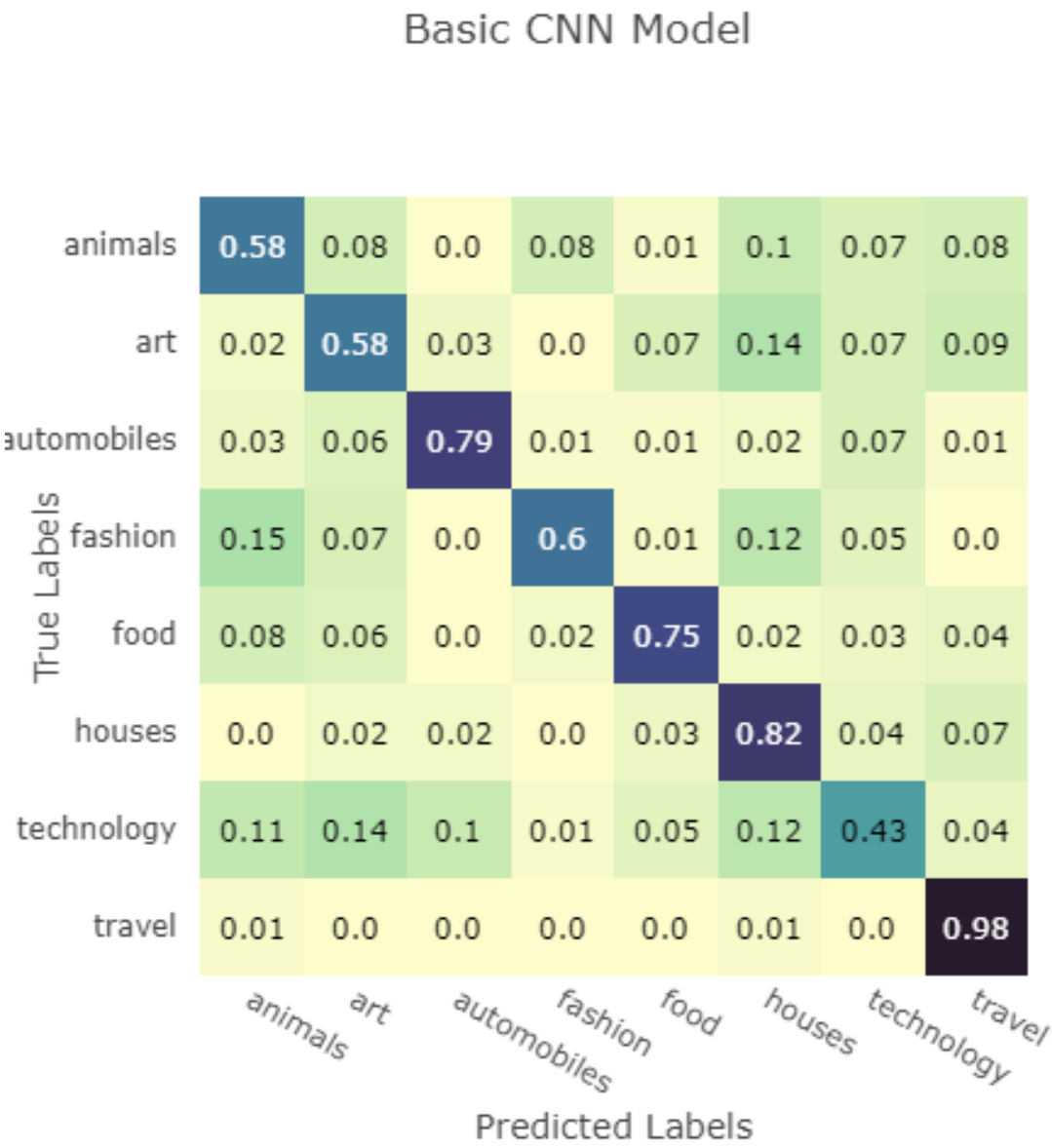


Fig 21. Basic CNN Confusion Matrix

Average CNN Model Accuracy on Test Set = 69.13%

The 4 of the best performing transfer learning models were tested on our test set and the resultant confusion matrices were printed out. The following were the results of the analysis:

MobileNet:

Fig 22. MobileNet Confusion Matrix

Average MobileNet Model Accuracy on Test Set = 92.0%

VGG-16:

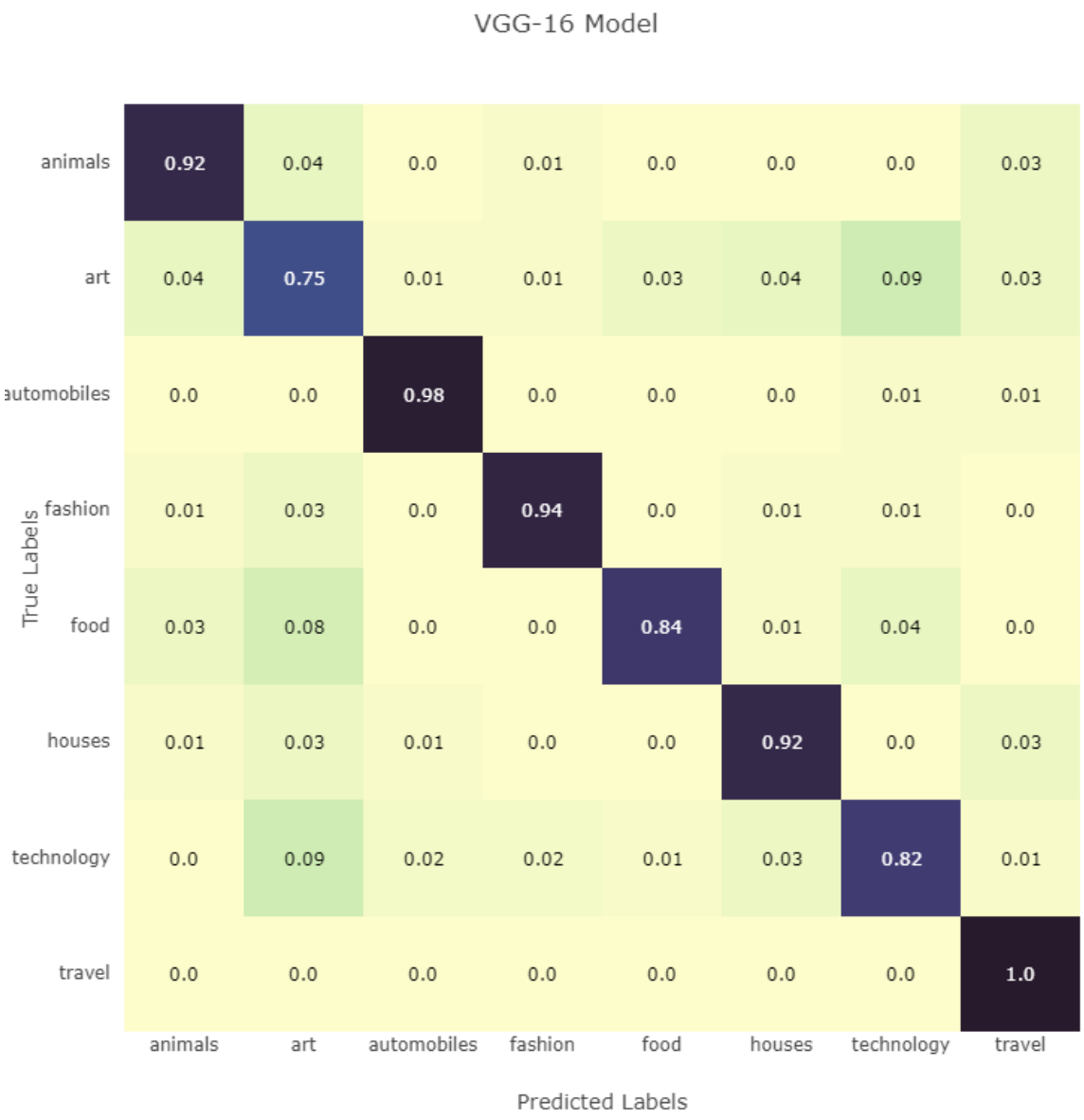


Fig 23. VGG-16 Confusion Matrix

Average VGG-16 Model Accuracy on Test Set = 89.625%

ResNet-50:

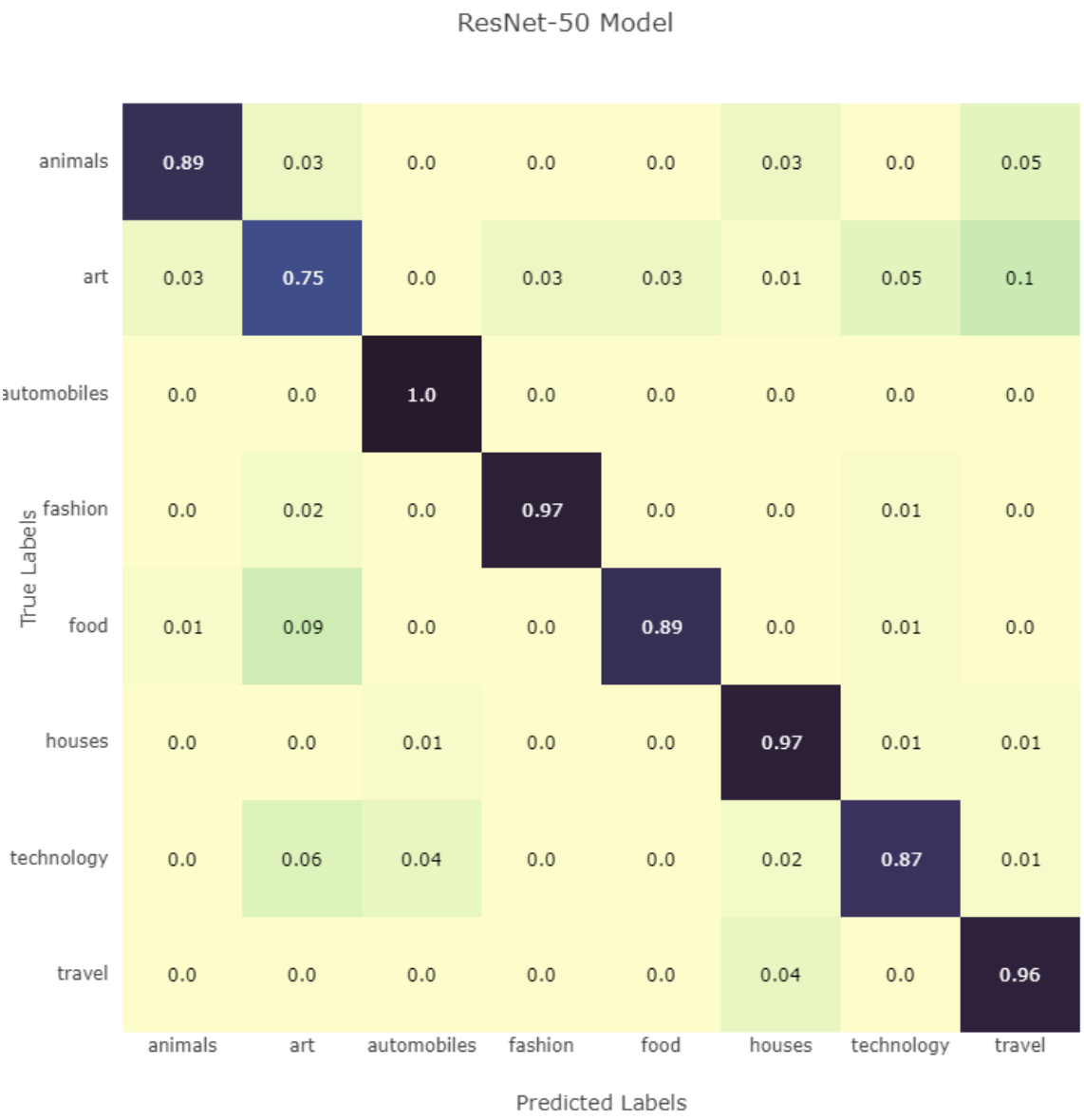


Fig 24. ResNet-50 Confusion Matrix

Average ResNet-50 Model Accuracy on Test Set = 91.25%

Inception-V3:

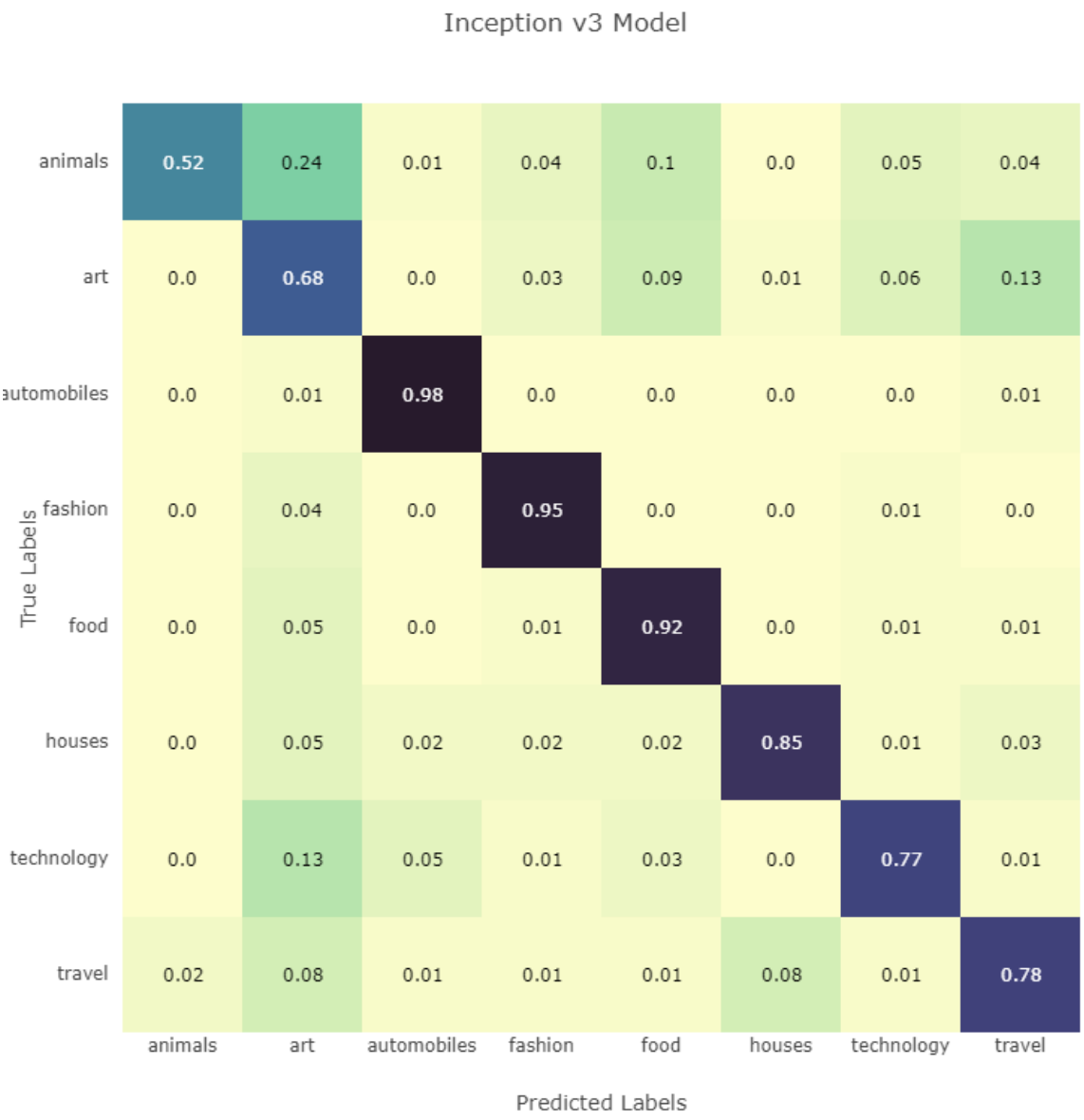


Fig 25. Inception-V3 Confusion Matrix

Average Inception-V3 Model Accuracy on Test Set = 80.625%

K-FOLD CROSS VALIDATION:

The best performing model on the test set turned out to be the MobileNet Transfer Learning Model. The model was further evaluated using 10-fold Cross Validation. The following average confusion matrix and accuracies were achieved:

Fig 26. MobileNet 10-Fold Cross Validation Confusion Matrix

Average MobileNet Model Accuracy using 10-Fold Cross Validation = 91.05%

RUNTIME ANALYSIS

In addition to general testing, we also decided to gauge the average time taken by each of the transfer learning models to output a prediction on any given image. This kind of analysis plays a crucial factor in deciding what kind of model can be deployed. In a real-world scenario, a conscious trade-off might have to be made between the prediction time of a model and the prediction accuracy of the model.

Transfer Learning Model	Average Prediction Time per Image
MobileNet	**0.025 s**
VGG-16	0.077 s
ResNet-50	0.097 s
Inception-V3	0.077 s

Table 8. Transfer Learning Model Runtime Analysis

COMPARATIVE STUDY OF DIFFERENT MODELS

Parameters	SVM	ANN	CNN	Transfer Learning
Accuracy	41.53%	44.75%	69.13%	92.0%
Training Time	~1 hr	~40 minutes	~13 minutes	~25 minutes
Performance on Each Class	Not significant in each class as only fashion and food are the classes which is giving accuracy of more than 50 %	Poor performance on almost all classes except for the food and travel classes. Almost similar performance to that of SVM due to lack of spatial orientation information	Better performance as compared to SVM and ANN. However, the model overfit the training data and received a poor generalizatio n accuracy.	High training and test accuracy. Extremely fast in making predictions. Achieved above 90% test accuracy on 5 out of the 8 classes.

Table 9. Comparative Study of Different Models

BEST ARCHITECTURE

Looking at the detailed analysis of the multiple models used to tackle the task at hand, we were able to make an informed decision on the best architecture for the ‘Pinterest Image Tagging’ problem.

It is evident that the **Transfer Learning** approach is much more useful in an advanced image classification problem. In general, applying this technique helped us achieve a high accuracy despite our small dataset of images.

Since previously proven architectures are used as base models in transfer learning, we are able to save a huge amount of time in designing a new architecture. Moreover, using pre-trained weights as a starting point for our problem greatly helps in reducing training time by enabling pre-compute for all the layers in the base model.

Specifically, the **MobileNet** model was the most successful for us in order to successfully classify images. It received an average accuracy of **92%** on our test set and was able to achieve an accuracy of **91.05%** using 10-fold cross validation.

Moreover, due to the lightweight, compressed, and efficient nature of the MobileNet architecture, we were able to complete training in just about **25 minutes**.

In addition, the MobileNet model was able to compute predictions in only about **0.025 seconds** per image. The lightweight and fast nature of the MobileNet model plays a crucial role in being able to work well on mobile devices with limited hardware capabilities which is indeed a crucial consideration point for a mobile-driven social media platform like Pinterest.

7. LESSONS LEARNED

The project presented an exciting opportunity to learn and understand few of the key concepts of the Deep Learning field. The roadblocks we faced in the process helped us dig deeper and find better solutions and techniques to solve the problem at hand.

To start, the process of data collection was the first challenge we faced, in terms of how cumbersome it was to manually tag these images. In response we learned to use the Keras “ImageDataGenerator” class which helped us sort and feed data into our models much quickly and easily, both, in terms of building the pre-processing pipeline and tagging the images for training.

Next, we realized soon enough how hard it was to use simple machine learning techniques to develop image classification models due to the sheer number of dimensions(pixels) a single image contains. It was evident that to be able to train such models, we would have to reduce the number of pixels we could feed into the model. This point was further strengthened when we were only able to train grayscale images in an acceptable time frame.

Even the use of an Artificial Neural Network was not of much use due to the lack of spatial information being captured by the individual nodes of the neural network. In addition, the size of the image again had to be reduced to about 1/3rd of the original image size to able to obtain a manageable number of parameters to train the network.

Even with the use of the Transfer Learning models, we faced a few issues in terms of required hardware resources to train heavy architectures like the Inception-V3 architecture with a batch size of 32. We noticed that we had a limited amount of memory in order to be able to train such a heavy model on our local computer.
Also, the amount of time taken to produce a prediction on an image also became an interesting point of consideration and comparison and we realized that it would be the best for us to have as much of an efficient, lightweight model as possible even if that would mean sacrificing a bit on the model performance.

To round up the project, we wanted to be able to create a model that was explainable as possible. The use of LIME analysis helped us understand and gain trust in our deep learning models.

8. CONCLUSION + FUTURE SCOPE

Social media is ubiquitous and user engagement with social media dominates the world. Pinterest is also one of the parts where users share, collect and manage the pins on their personal boards.

Since the majority of the content on Pinterest is images and visual graphics, it is necessary for users to order the content and structure their boards under relevant categories.

The aim of this project is to suggest appropriate categories respective to their collective pins so that users can easily put their pins to the boards with pertinent categories. In order to provide the effectiveness of the model, we have also proposed a LIME analysis where each prediction is explained through the most suitable features. It tries to identify the features that explain a particular prediction in a more interpretable way.

With LIME explainers, we try to build the trust among the user and classifier so that users can easily believe in the predictions of the classifier. In this process of building an image tagging classifier, transfer learning proved to be the best approach.

We have also analyzed our classifier with different experimental setup such as hyper parameter tuning and k-fold cross validation. MobileNet has shown a promising accuracy over the Pinterest data set.

We have also created the confusion matrices to analyze the class wise accuracy for the dataset and have reached the conclusion that, using the Transfer Learning approach with MobileNet as a base model outperforms all the other approaches.

FUTURE SCOPE

- Coarse grain classification of images can be improved with fine-grained classification with multi label classification. As we have only considered 8 popular categories however these labels can be extended with more fine categories.
- Currently, the classification is based on purely images however we can also consider the hashtags and text content related to images. Comments of users can also be collected for a particular image to classify the instances more correctly.

9. REFERENCES

[1] Han, J., Choi, D., Chun, B. G., Kwon, T., Kim, H. C., & Choi, Y. (2014). Collecting, organizing, and sharing pins in Pinterest: interest-driven or social-driven?. *ACM SIGMETRICS Performance Evaluation Review*, *42*(1), 15-27.
[2] https://www.omnicoreagency.com/pinterest-statistics/
[3]https://medium.com/analytics-vidhya/image-classification-a-comparison-of-dnn-cnn-and-transfer-learning-approach-704535beca25
[4] https://www.kaggle.com/shivamb/cnn-architectures-vgg-resnet-inception-tl
[5] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why Should I Trust You?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).
[6]https://github.com/marcotcr/lime/blob/master/doc/notebooks/Tutorial%20-%20Image%20Classification%20Keras.ipynb
[7] https://keras.io/applications/
[8] https://keras.io/preprocessing/image/