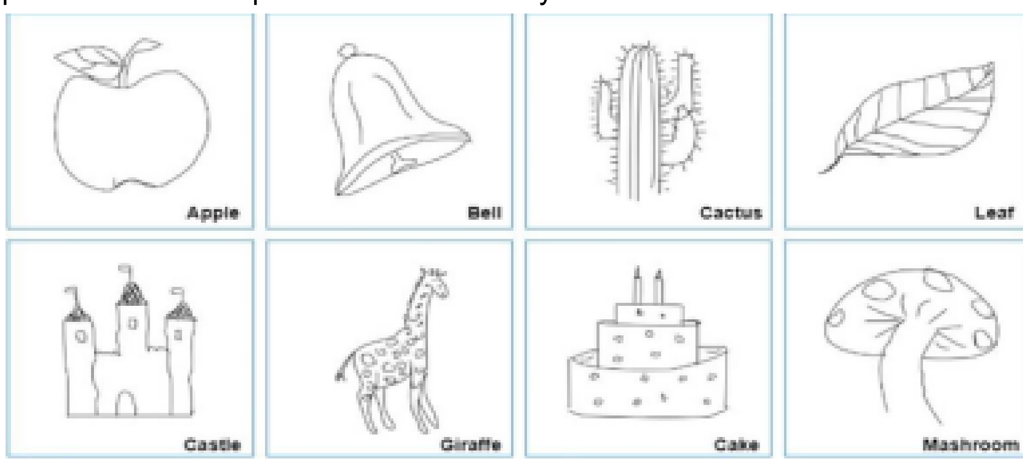
Hand-drawn Sketch recognition

Abstract— ***We were given the Sketchy database, the first large-scale collection of sketch of photographic objects sampled from 125 categories and acquire 75,471 sketches of 12,500 objects. The Sketchy database gives us fine-grained associations between particular photos and sketches, and we use this to train cross-domain convolutional networks which embed sketches and photographs in a common feature space. We use our database as a benchmark for fine-grained retrieval and show that our learned representation significantly outperforms both hand-crafted features as well as deep features trained for sketch or photo classification. Beyond image retrieval, we believe the Sketchy database opens up new opportunities for sketch and image understanding and synthesis.***

Keywords—VGG16,RandomForest Classifier , Naïve Bias , SVM Models

1.INTRODUCTION

Sketch recognition attempts to recognize the intent of the user while allowing the user to draw in an unconstrained manner. This allows for the user to not have to spend time being trained how to draw on the system, nor will the system need to be trained on how to recognize each users particular drawing style. Deciphering freehand sketches can be viewed under the lens of image category recognition, a well studied problem in the computer vision community.



In this paper ,we explore the use of Random Forest Classifier ,Naïve Bias ,SVM Models classifiers and Deep neural network(DNN) architectures for sketch recognition .

2. RELATED WORK

Reading the Dataset:

We read the dataset using tensorflow . Initially 125 classes were given out of which we decided to pick 10 classes because our aim was to study the effects of various machine learning techniques like cross

validation, data preprocessing(feature extraction) and Dimensionality reduction. A total of 5672 files are collected from 10 classes.
Now we make 2 arrays X and Y of size (5672, 224, 224, 3) ,(5672, 10) respectively. Here Y denotes which class the image belongs to.

Extracting features from images using pretrained VGG16 Model:

Now we have to convert array X into 1-D array because the classifiers only accept 1-d input. For this we use the VGG16 model which is a feature extractor model trained by google .It converts an array of size (224, 224, 3) to 1-D array of 4096 features..

Reducing Dimensionality of Dataset and Standarising dataset

Now due to its large size(4096) it is taking infinite time to process .So now we have to perform dimensionality reduction in order to merge some features and return an compact array. For this we used **Principal component analysis** (**PCA**) that converts X into an array of (5072,100) with 100 features per file.
Now we standarise the following dataset using Standard Scaler.

Splitting the Dataset into train and test dataset.

Split the following dataset using model selection into train_x, train_y, test_x and test_y which are further used in classification.

3.METHODS

- We splitted the dataset into training and testing.

- We have train and testes our model on 3 classifiers which we have studied in class:

1. Random Forest
2. Naive Bias with Cross Validation
3. SVM model

RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees. Random decision forests are correct for decision trees' habit of overfitting to their training set.Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance

By training the model and testing on the dataset we get the accuracy of

54% NAIVE BIAS WITH CROSS VALIDATION

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features. They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve higher accuracy levels.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers

By training the model and testing on the dataset we get the accuracy of 79%

SVM MODELS

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis, SVMs are one of the most robust prediction methods, being based on statistical learning frameworks. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier . SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

By training the model and testing on the dataset we get the accuracy of

85% It is the best accuracy that we get from all the three models.

4.Using neural networks for a larger dataset to get better results

In the above traditional methods on sketch recognition we followed the conventional image classification paradigm that is, extracting hand-crafted features from sketch images followed by feeding them to a classifier. But these features are sensitive to the view perspectives and some appearance cues of drawn sketches. Furthermore, the ability of learning algorithms to train the classification models are also influenced by the handcrafted features and the capacity of classifiers like SVM to memorize feature information.

Furthermore these traditional algorithms limit us on the number of classes we can use as well as in accuracy . In comparison to hand-crafted visual features, features learned by deep neural networks (DNNs) have recently been shown more capable of capturing abstract invariant features to various phenomena in the visual world .

Preprocessing

So, here we are gonna use the Neural network for 24 classes. The initial steps of reading the dataset and Extracting features from images using pretrained VGG16 Model are the same. Here X and Y are of size (6975, 224, 224, 3) and (6975, 24) respectively.

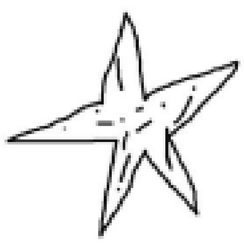
Now we split the dataset into train and test , here train_x, test_x, train_y, test_y have size (5580, 4096) (5580, 24) (1395, 4096), (1395, 24) respectively.

Creating the Dense neural network

Here first we initialized a sequential model ,then added three sequences of dense and dropout layers . A densely connected layer provides learning features from all the combinations of the features of the previous layer . Dropouts layers are used for regularization that is used to prevent overfitting in the model. Dropouts are added to randomly switching some percentage of neurons of the network. This is done to enhance the learning of the model.
Then we used Adam optimizer and categorical cross entropy to fit the model. After which the **accuracy came out to be 96.77%.** Which is a good improvement from our previous best model of SVM.

Predicting on custom images

Now we tried to test on some custom hand drawn images belonging to the above 24 classes and get the prediction



This hand drawn image is correctly classified as class 14 i.e. starfish.

4.Conclusion

We have calculated the performance of our proposed approach on the dataset which is a collection of hand-drawn drawings having about 125 different object classes. We have divided the entire data in 80-20 ratio for training and testing respectively. The approach of converting raw data to simplified forms has helped a lot in reducing the computation for training the models as now the models have only refined meaningful data to process and extract features. Each instance in the data has the coordinates of stroke segments that are drawn to make the object (belonging to the class of 345 different objects). We have given trained and tested our models on the basis of these classes. We have compared the models based on accuracy. Our work in the paper includes object recognition. But data dimensionality and size are the biggest hurdles in our progress. To overcome this problem, we have used the vgg-16 model to convert our raw data . This conversion proved to be very effective in refining our information and fastening the training of our model for the recognition task. The machine learning models we used – Random forest, naïve bias and SVM. We got the accuracy of 85 % with SVM which is maximum and with deep neural network 96.77%.