**Object Classification**

Zeeshan Ahmed Pachodiwale, Sai Harshith Reddy Gaddam, Tanmai Bhat

*Computer Engineering Department, College of Engineering*

*San Jose State University, San Jose, CA 94303*

**Abstract**

*Object Classification is a concept of Image Classification that finds a solution to the task of classifying different objects for which the model is trained. This is grown to be one of the more challenging tasks in machine learning. The objective is to make a machine learn how to detect and classify an object in a given image. The paper tries different approaches to try and answer this problem.*

**1. Introduction**

Object classification has already been the significant research direction and the focus in computer vision which finds its applications in areas like driverless vehicles, robotics, video surveillance and pedestrian detection etc. This paper describes the key concepts involved in successfully building a classification model that can successfully differentiate between 14 different classes given in the dataset.

**2. Dataset**

The dataset used for the object classification task was traffic images collected from traffic cameras all over the country.

**2.1. Dataset Overview**

The dataset contains images of 14 different classes including the following,

TABLE 1: Classes in the dataset

|  |  |
| --- | --- |
| Cars | Busses |
| Motocycles | Bicycles |
| Van | Trucks |
| Large Trucks | Medium Trucks |
| Traffic Lights | Pedestrians |
| SUVs | People |



**Fig. 1 – Different Classes in the Dataset**

**2.2. Observations**

The dataset was found to be highly imbalanced with some classes like cars, having over 15000 images and others like bicycles having only 900 images. This imbalance in the dataset can cause the model to perform poorly on the minority classes.

**3. Model Selection and Approach**

For object classification task we have tried many machine learning algorithms. The aim was to compare the performance of many of these algorithms for the given dataset. Using different machine learning algorithms helped us in understanding various data preprocessing techniques and dimensionality reduction techniques. And also helped us understand the need for cleaner data to build models with better accuracy and precision. The ML algorithms used are as follows,

**3.1. Support Vector Machine**

Support vector machine is a classifier defined by a hyperplane. In supervised learning, SVM outputs an ideal hyperplane which classifies the new data. SVM is effective in tasks such as classification, numerical prediction, and pattern recognition tasks.



**Fig. 2 – Support Vector Machine**

**3.2. Random Forest**

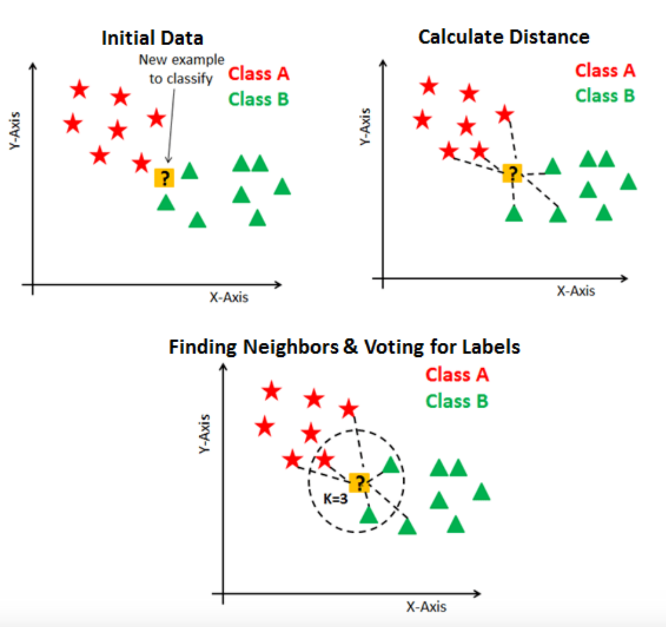
Random forest is just a bunch of decision trees bundled together. A decision tree is a supervised learning algorithm. Random forest is an ensemble of various decision trees (weak learners) to produce a strong learner. It is known as one of the bagging methods.



**Fig 3: Random Forest**

**3.3. KNN Classifier**

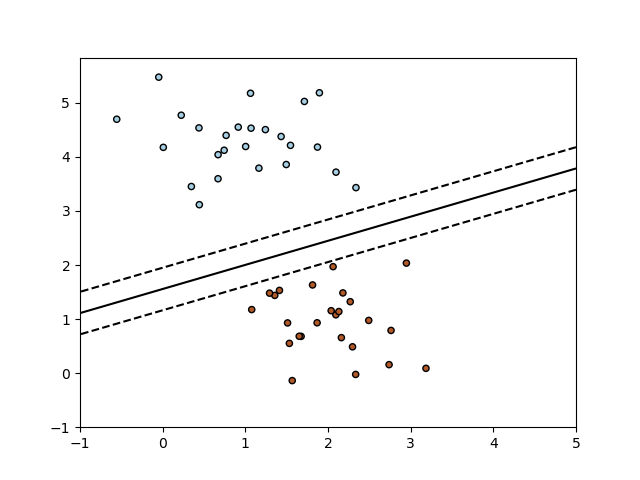
K-nearest neighbor is one of essential classification machine learning algorithm. KNN is a supervised algorithm widely used in pattern recognition, and data mining. Benefits of KNN algorithm are that it is easy to interpret the output and has low calculation time. The distance between two data points can calculated by Euclidean, or Manhattan or Minkowski functions. In another approach instead of computing the distancing, the KNN classifier can also compute the similarities between any given data points too. Both the approaches work the same way.



**Fig. 4 – K-nearest Neighbor Classifier**

**3.4. Stochastic Gradient Descent Classifier**

This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning, see the partial\_fit method. For best results using the default learning rate schedule, the data should have zero mean and unit variance.



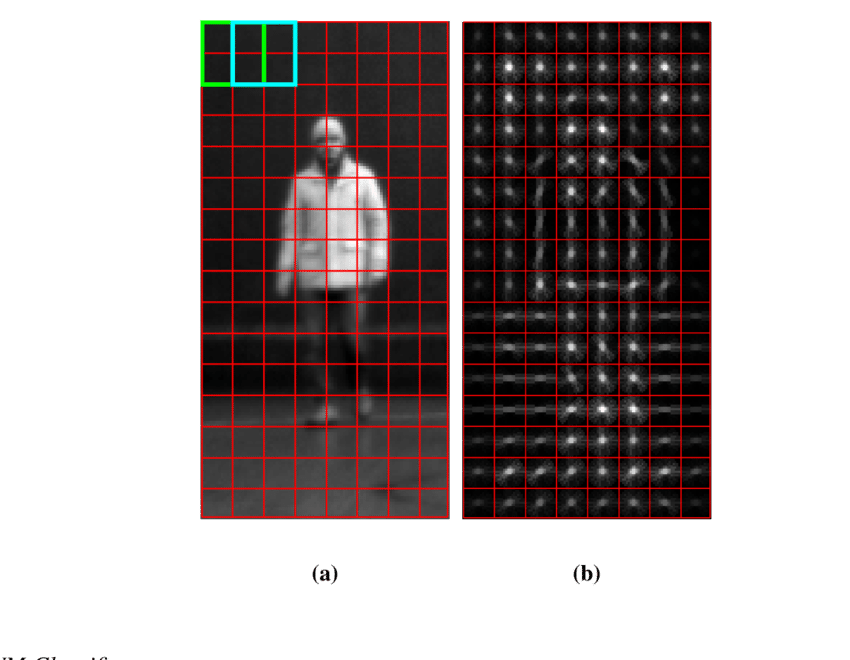
**Fig 5: SGD Classifier**

**4. Dimensionality Reduction**

Dimensionality reduction techniques are used to reduce the features without losing much information and improve the performance of the model. The advantages of using dimensionality reduction are decrease in computation time because of less features, and visualization of data. The various dimensionality reduction techniques are as follows:

**4.1. Histogram of Gradients**

Mostly used in computer vision and image processing for object detection. Frequency of gradient descents is counted in localized portions of an image. This method is like that of edge orientation histograms but is slightly different in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

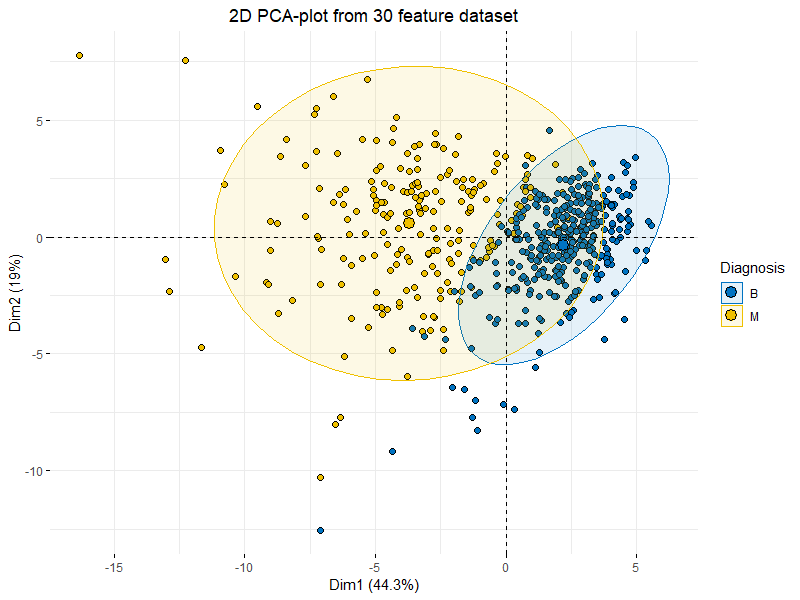


**Fig 6: HOG**

**4.2. Principal Component Analysis (PCA)**

Principal Component Analysis (PCA) is a dimension-reduction technique that is used to decrease a large number of feature to a smaller group that would still contain most of the information in the larger set.

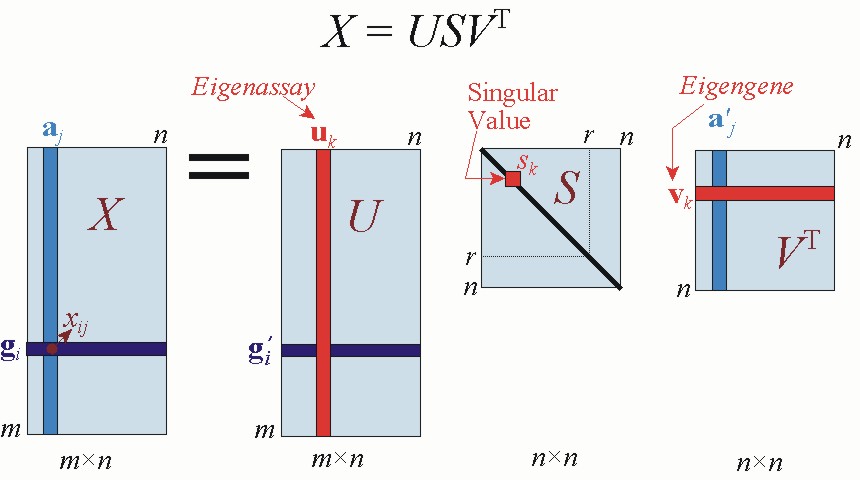
Principal component analysis (PCA) is a mathematical procedure that converts many correlated features into a small set of uncorrelated features called principal components. The first principal component shows the highest variance in data and the variance decreases we go down the Principal components.



**Fig 7: PCA**

**4.3. Singular Valued Decomposition (SVD)**

SVD is another dimensionality reduction technique in which we use matrix factorization to calculate the singular values from the matrix and using those singular values we can reduce the dimensions.



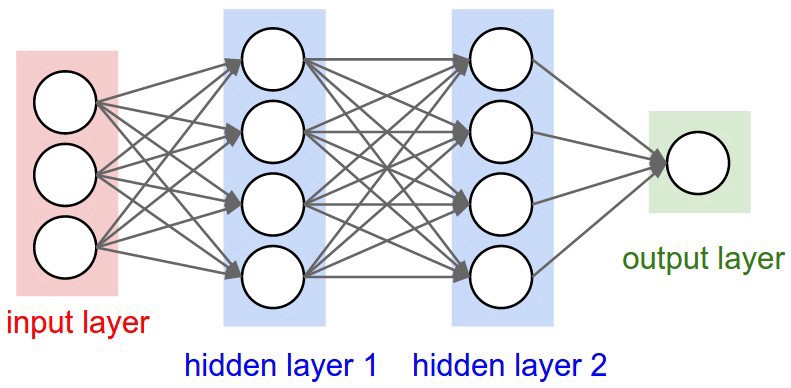
**Fig 8: SVD**

**5. Deep Learning**

Image classification task has achieved state-of-the-art results using deep learning models. The concepts of deep learning are:

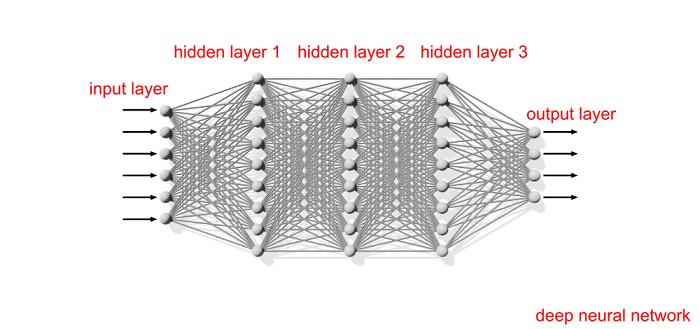
**5.1. Neural Network**

A neural network concept is derived from a human brain’s neural network. Likewise, neural network is a collection of neurons which process the information when they receive an input and produce output signals. A neural network consists of an input layer, hidden layer, and an output layer



**Fig 9: Neural Network (Sample Image)**

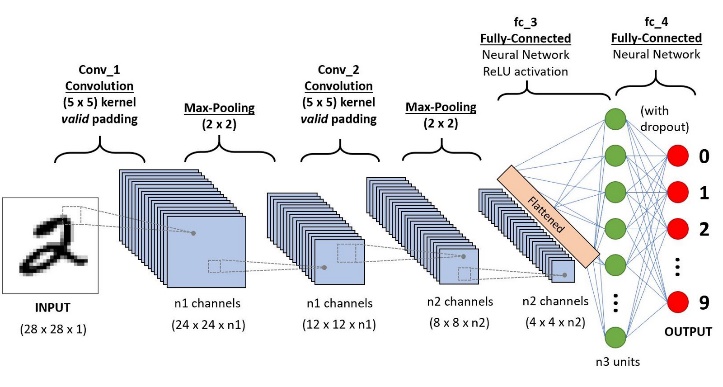
The main factors that distinguish the various types of neural networks: number of hidden layers, and how the neurons/nodes are connected. A layer is said to be fully connected if it receives input from every neuron/node of the previous layer. A neural network is said to be a deep neural network when it has multiple hidden layers. The neural network with one or zero hidden layer is said to be shallow neural network.



**Fig 10: Deep Neural Network (Sample Image)**

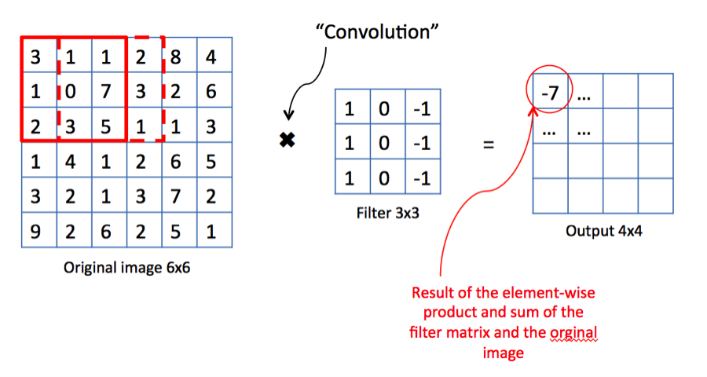
**5.2. Convolutional Neural Network (CNN)**

CNN is a type of a neural network which has convolution as one of the main building blocks. In a deep neural network some hidden layers are called convolutional layers. The term convolution means a function which performs element wise multiplication followed by summation. CNNs are most widely used for analyzing images. In a convolutional layer there exists a n x n matrix which is called a filter or a kernel. The kernel/filter is used to identify patterns from the images.

****

**Fig 11: CNN**

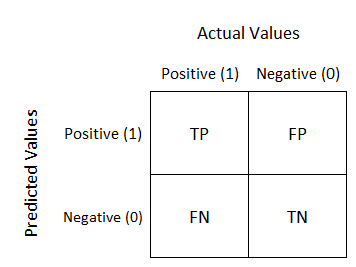
The number and size of the filter are known as the hyper-parameters. Hyper-parameters are values which are tuned during the training process to achieve better results during the training process of the neural network.



**Fig 12: Convolutional Operation by a 3x3 filter**

**6. Evaluation Metrics**

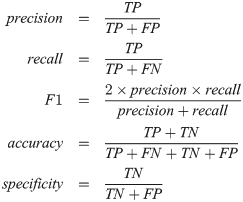
An evaluation metric is used to check for the performance of a neural model. A metric should be chosen carefully based on the problem we are trying to solve. For object classification tasks, commonly used metric is F1 score, and mean average precision (mAP). It is the mean of the precision calculated over all the classes.



**Fig 13: Confusion Matrix**

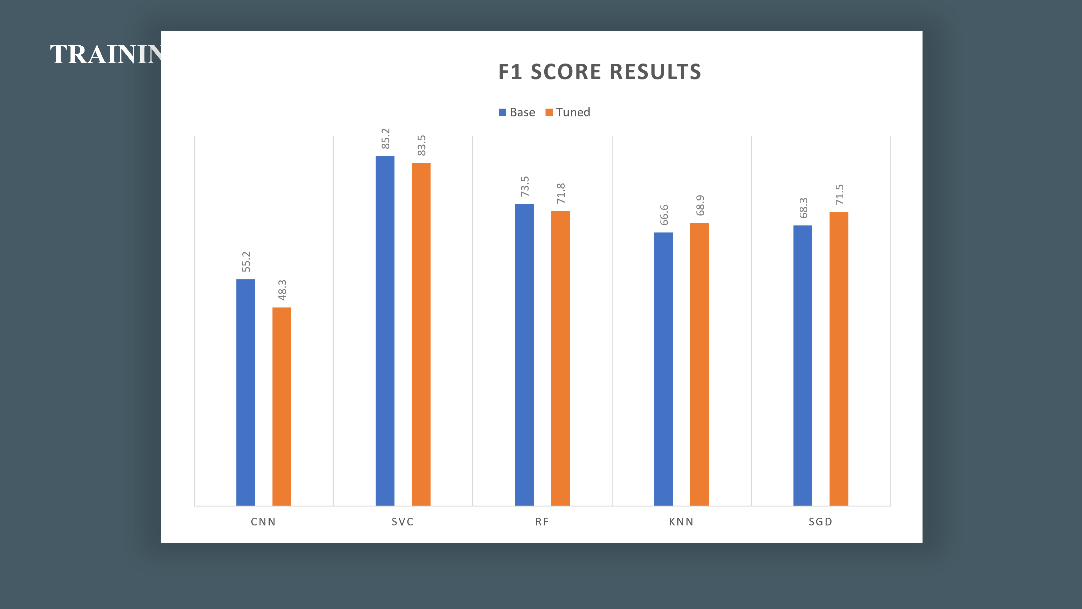
**6.1. Formulae**

This section details the list of formulae that is used for performance measurement of a model.



**7. Results**

We got the following F1 scores for different algorithms.



**Fig 14: Performance Results**

**8. Conclusion**

We analyzed the dataset and found that it was imbalanced. Various feature extraction techniques were successfully employed to reduce the features and easily fit our algorithms. Histogram of gradients was chosen to build the model. We built an model with an accuracy of around 86% and integrated that model with a android application for real time object detection.

**9. References**

[1] Dr. Simon Shim, *Lecture Notes of CMPE 257*, Computer Engineering Department.