Object Classification

Abhinav Jindal
2016CSB1026
2016csb1026@iitrpr.ac.in

Chirag Khurana
2016CSB1037
2016csb1037@iitrpr.ac.in

Mattela Nithish
2016CSB1042
2016csb1042@iitrpr.ac.in

ABSTRACT

This report provides the details of our object classification project performed on PASCAL VOC dataset. Various classification techniques have been tried such as Neural Networks, SVM and Clustering. From selected regions, the classification accuracy as reached up to 90 % for some models.

Abhinav Jindal, Chirag Khurana, and Mattela Nithish. 2019. Object Classification.

1 INTRODUCTION

The motivation of this project is to achieve object classification using the techniques of Machine Learning on given image. Due to deficiency of resources, only a subset of PASCAL VOC has been used for training and testing. Various methods have been tried and the performance evaluation has been done. First the model is passed through CNN layers to get the feature vector of an image and then these feature vectors are used for classification of image in the classes chosen. Classification methods used include neural networks, SVM and KMeans clustering. .

2 METHODOLOGY

2.1 Preparing Dataset

In this project we used PASCAL-VOC dataset[4]. We considered 9 classes out of the number of classes given in the dataset. The classes that are considered are aeroplane, bicycle, bird, boat, car, person, horse, dog, cat. This dataset has information of all the objects that are present and the region in which objects are present in each image. First we extracted and saved all the regions from all the images which have the classes mentioned above.

2.2 Feature Extraction

After all the necessary regions have been extracted, We use VGG16 model with ImageNet weights to get feature vector of shape (4096,1) from an image. VGG16 is a pretrained model in Keras library. We used 5 CNN layers followed by one flatten and two fully connected layers of the VGG16 model. We took the output of the layer 'fc2' of this model which is of required shape as we want. For the training purposes, We stored the VGG16 model's output for all the regions.

2.3 Image Classification

For this part we experimented using different algorithms which are Support Vector Machines(SVM), Artificial Neural Networks(ANN), KMeans. The input to all these algorithms is a vector of shape (4096,1). The first two techniques are supervised learning where as KMeans is unsupervised

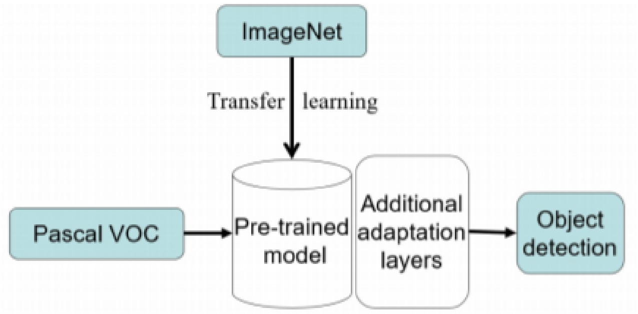

Figure 1: Overall work flow in this project.

2.3.1 Support Vector Machine. In this method we built one versus one SVM classifiers. So here there are 45 classifiers. For every classifiers, We assume one of the class to be positive and other to be negative. For each classifier the main idea is to find that boundary which has maximum margin. To train this we used Unconstrained loss function that combination of margin and Hinge loss function. Loss function is given by,

$$L = \|(W)\|^2 + \left(\sum_{n=1}^{N} max(0, 1 - f(x_n) \cdot y_n) \right) \tag{1}$$

where N is number of training instances, $f(x_n), y_n$ are value predicted and actual values of nth training instance respectively which are either 1 or -1. We used gradient descent to find W. This idea is taken from [2]In the default setup we used 200 iterations and learning rate = 0.02. By this method the accuracy of the test dataset is around 80-82% and train dataset is around 92-95%. We also observed the change in accuracy with respect learning rate the graph is shown below.

2.3.2 KMeans Clustering . In this method, we are using clustering to classify the instances in one of the classes. First we create clusters from the training data and create the mapping based on maximum instances of the class present in the cluster. Then this clustering and mapping is used to find the labels of any future instance. (Used [5] to implement KMeans).

2.3.3 Artificial Neural Networks. In this method, we used ANN layers to classify images into various classes. The layer architecture used is (4096,1024,256,64,K), where K is number of classes. Dropout rate used is 0.2 to compensate overfitting. Categorical cross entropy loss function is used for this method to do back propagation. Elu activation is used in the hidden layers, and Softmax activation used in the output layer. (used [1] to implement ANN)

, ,
2019.

3 EXPERIMENTAL SETTINGS

In all the methods we split given dataset into two parts, test dataset and train dataset. We split the data into with test split ratio of 0.2. This splitting process is random while training this model. So on and average, Accuracy on test dataset in calculated which will be performance metric for the model.

First the input image is passed through CNN network as stated before and then feature vector output is fed into different classifiers.

In case of Support Vector Machine, Hyper parameter that has to be tuned in this case is the learning rate. For this a plot of Accuracy and learning rate is performed. Here We assumed that learning rate can be around 0.01 to 0.5. Since the number of iterations to train this model is fixed .i.e 200 (because it has to learn 45 classifiers and the effect of iteration on accuracy of the model is not significant after 150 iterations)

For the case of kmeans, number of clusters were varied to get the training and testing accuracy on the split dataset.First, sklearns' kmeans method is used to create clusters and then after finding the mapping between clusters and labels from confusion matrix accuracy is tested by seeing the number of correctly classified examples.

ANN model is created by adding layers from the keras library and architecture is fixed but can be varied quite easily. Then the updated model based on training on the training dataset is used to classify the test input.

4 RESULTS AND DISCUSSION

Results and Discussion corresponding to the methods are below

4.1 Support Vector Machine(SVM)

The table for accuracy vs learning rates is shown in Table 1. Observing this table we can see that person and cat classes have been identified with highest accuracy as the number of training instances of person and cat are more. We also tried different learning rates to train the model using same training dataset. The plot is shown in Figure 2. Observing this figure the test accuracy is increasing initially as the learning rate is increasing initially highest accuracy is found around 0.2 to 0.4. This is because the number of iterations that are performed are very low for learning rates less than 0.1 after that there is some kind of deviation in because of higher learning rates.

4.2 KMeans

We have tested accuracy on different number of clusters(see figure-3) and for each class when number of clusters are equal to number of classes chosen(see table-2). We see that accuracy drastically increases when the number of clusters are more or equal than the actual number of classes.This is because certain classes get merged as we can see from table-2 leading to drastic decrease in accuracy. But on increasing the number of clusters these merged clusters get split and similar looking classes get pushed apart so that they can be classified properly. We see that different very looking objects do not get merged but somewhat similar looking objects get merged in clusters.

Table 1: t_p of corresponding labels in case of SVM

Label	t_p rate
aeroplane	89.39
bicycle	92.25
bird	88.60
boat	88.46
car	86.44
person	94.98
horse	91.66
dog	91.33
cat	94.44

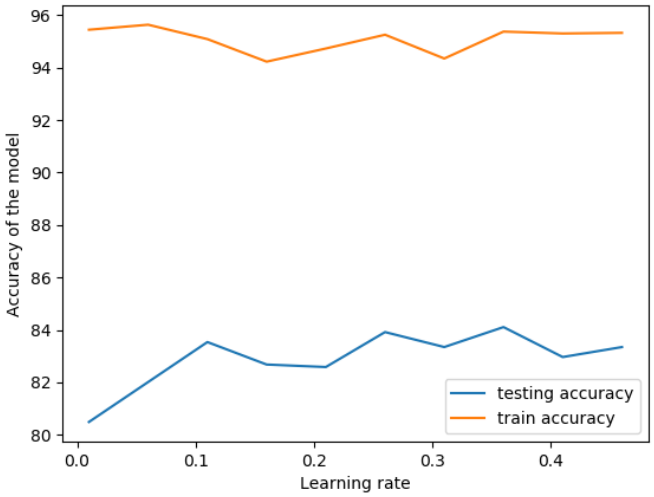

Figure 2: Learning rate Vs Accuracy of SVM.

4.3 Artificial Neural Networks(ANN)

For ANN the training accuracy reached upto 100% in even 30 epochs without dropout which suggested overfitting which reduced to around 95 after apllying dropout. Also the test accuracy reached 90% after dropout which is much better than the other 2 approaches. Also the tp rate for person is much higher as we can see. This is because person images are much more leading to more training instances for a better model. The architecture used also plays a major role in determining the accuracy of the model. In this case, Table for t_p rate for all the classes considered is shown in Table 3

Table 2: Kmeans accuracy for each class

Label	tp rate
aeroplane	93.2
bicycle	80.62
bird	96.4
boat	not classified
car	90.7
person	60.2(avg)
horse	not classified
dog	57.68
cat	93.2

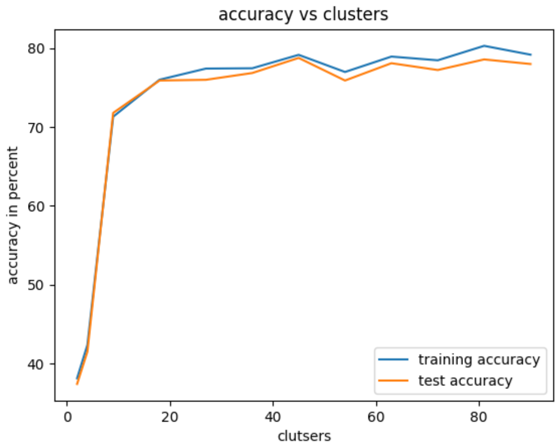

Figure 3: Clusters Vs Accuracy of KMeans.

5 SUMMARY

Thus we have achieved the task of object classification with quite reasonable accuracy. KMeans being unsupervised learning lead to somewhat lower accuracy when compared to other 2 methods of supervised learning. ANN gave the highest accuracy followed by SVM.

6 FUTURE WORK

The task of object recognition is not only classifying the given region but also identifying that region. We tried to perform this task by the method proposed in the [3]. We selected 2000 regions in an image using quality selective search and ran the classifier on all

Table 3: t_p of corresponding labels in case of ANN

Label	t_p rate
aeroplane	83.33
bicycle	79.62
bird	89.87
boat	75
car	84.74
person	91.22
horse	77.08
dog	84.25
cat	87.96

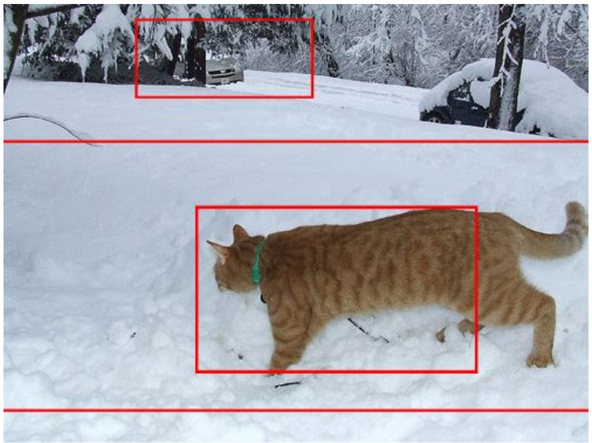

Figure 4: Object recognition.

the regions and returned the regions which were not overlapping and had maximum posterior probability as compared to overlapping regions. Opposite to our expectations, this method did not perform that well as expected. Some of the background regions were also classified as one of the classes with very high probabilities leading to faulty results as shown in figure-4. We see that 2 regions are being detected for cat. The larger region was giving higher probability than the smaller one and therefore the exact object is not recognised. This was the case in almost all the images. Either the background points labelled were too much or correct object was not detected. Thus this approach was not optimal. Anyways that part in also in the code if you want to test. Another approach may be finding those regions with the network as well as suggested by some advanced papers. Also all the classes could have been considered but that was not taken due to lack of resources.

REFERENCES

[1] Keras Model Documentation. [n. d.]. Keras Model documentation. https://keras.io/models/model/

[2] Rohith Gandhi. [n. d.]. SVM model from scratch. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[3] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* abs/1311.2524 (2013). arXiv:1311.2524 http://arxiv.org/abs/1311.2524

[4] PASCAL-VOC-DATASET. [n. d.]. PASCAL-VOC DATASET. https://pjreddie.com/projects/pascal-voc-dataset-mirror/

[5] SKlearn. [n. d.]. Kmeans documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html