Image Segmentation Using Semantic Learning

Prudhvi Suggula
Ps5m6 @mail.umkc.edu

Nikitha Kona

***Abstract*—This paper gives a detailed explanation of retraining the last layer of the inception model and incorporating two fully connected layers one for image classification and another layer for bounding boxes**

***Index Terms*— Semantic Learning, Inception Model, Semantic Learning**

I. Introduction

Image Classification and Segmentation has become a part of our everyday life such as face recognition, tracking, object classification. We have seen many applications that classifies different objects in a given Image, but drawing bounding boxes around the objects or semantic segmentation of different objects in a given image yields specific positions of the objects and provide the user with better understanding. In this paper, we discuss how we chosen different parameters in building a network with maximizing the Accuracy. We had analyzed pacel voc 2005 and CIFAR datasets for our problem. We use bottleneck features of inception model [1] and trained a sub-nets to solve both classification and regression .

II. Related WORK

Before the success of Convolution neural network the widely-used object detection methods include HOG, SIFT and Fisher Vector for feature extraction. By the advances in the deep learning CNN had made a remarkable improvement in classification [2] . For a image feature extraction CNN are best suited for this task because the convolution helps to search the similar patterns in the filter bank this made a better representation of image which resulted good accuracy in ILSVRC . The present state of art for localization, object detection is RCNN which divides the detection into two categories namely classification and then using regression to get the coordinates for bounding boxes. There are several recent works such as YOLO and RCNN that are used for object detection and classification . We have chosen their approach and made analysis on two datasets , with different learning rate , activation functions and number of neurons per layer observed the convergence .

III. PROPOSED WORK

There are various ways to choose the model architecture, we chosen to use the weights of the inception model which is presented by google for ILSVR 14. This model have some the significant features such as the inception block and very efficient use of parameters to meet the real time computation on mobile devices. The key reason to use to use the inception model is their architecture and the intuition in paper " the correlation statistics of the last layer and cluster them into groups of units with high correlation "[1] .

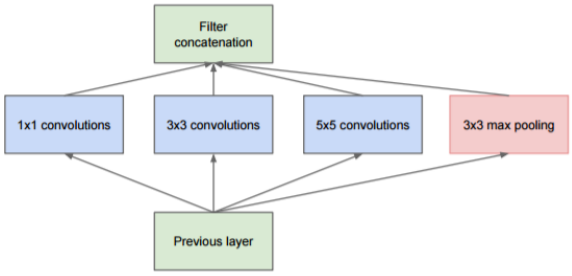


Figure 1, inception block [1]

The inception model solved the problem of dimensionality reduction and gave us the efficient and compact way of representation of a input image . We implemented a classifier upon this to separate the different classes of the both datasets.
We just used the 2 datasets to know the set of parameters work well for both of them. We consider the detection problem as a finding a bounding box covering the subject in the given image. This problem can further be incremented to a full segmentation where all the background needs to be removed rather than the subject. For now we just considered the problem to predict a box around a subject. We are predicting the top left and the bottom right corners of the box to plot on the image . so, this turned out to be a regression problem. We used the fully connected layers of different sizes for both classification and repression problem .

A. Algorithms and Pseudocode

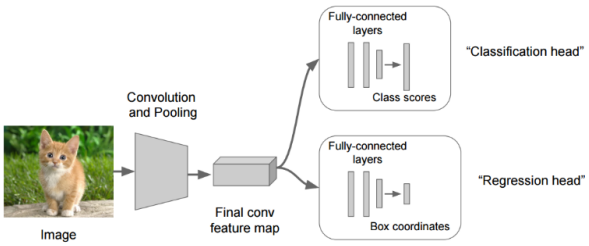
In our paper, we have used inception model and retrained the last layer of inception model to fit to our dataset .this just made the learning process so easier because we have the optimal weights available for the net [4], this helps in described the each the data set in a way that small classifier can further able to decide each class with the good accuracy[3]. We then trained the added layers by freezing the weights of the inception module .this makes the problem converge the problem very fast than training a whole new net. We used both fully connected neural nets for the regression and the classification ends. There are various net implementations available to train the dataset like VGNET and various algorithms like Fast RCNN, YOLO for detection and localization but our main focus is to evaluate the performance changes resulted when tweaking the parameters .

IV. IMPLEMENTATION AND EVALUATION

A. System Design and Implementation

I. SOFTWARE ARCHITECTURE

As explained in the previous sections the below image depicts the architecture of our implementation. Given an image we use convolution and pooling to extract features from different scales and we perform pooling for dimensionality reduction and after achieving the final convolution feature map we have two fully connected layers one that predicts the class score which can be done by using classifiers such as softmax and the next one is regression head that produces coordinated for the bounding boxes.



II. IMPLEMENTATION DETAILS

We have implemented the inception model by downloading the weights and we have removed the fully connected layer and passed our dataset and achieved optimal weights. We have used python and Tensor Flow for implementation purpose. We haven't used any GPU for implementing our model. For the web application we have used HTML,jquery and we have connected tensorflow at the backend that fetches the details based on the user input

III. EVALUATION PLAN

A. Dataset

We have used PASCAL VOC 2012 dataset that is mainly used for object localization and detection. It consists of 20 classes. And we have taken three classes for our project

Each image has annotation which has the below format where we extract the (xmin,ymin) and (xmax,ymax) for drawing the bounding boxes

```
# PASCAL Annotation Version 1.00

Image filename : "TUDarmstadt/PNGImages/motorbike-testset/motorbikes024.png"
Image size (X x Y x C) : 300 x 207 x 3
Database : "The TU Darmstadt Database"
Objects with ground truth : 1 { "PASmotorbikeSide" }

# Note that there might be other objects in the image
# for which ground truth data has not been provided.

# Top left pixel co-ordinates : (1, 1)

# Details for object 1 ("PASmotorbikeSide")
Original label for object 1 "PASmotorbikeSide" : "motorbikeSide"
Bounding box for object 1 "PASmotorbikeSide" (Xmin, Ymin) - (Xmax, Ymax) : (21, 64) - (184, 172)
```

Figure 2 sample image from pascal voc

Each image has a label in the form of a text file which has the above description.

B. System Specifications and Measurements

Imput image size for cifar is 32*32 color image where the inception net accept a data size of 299*299 , we just upscaled the size to get reduced representation for the end layers of the model . We have 10 classes and 5000 samples for each .We used different learning rates and activation functions the accuracy will be provide in the table below. In contrast pascal voc have a slighter large images than inception mode accepts so , we need to down scale the image which is the fed to the model to find the bottle neck features.

I. EVALUATION AND RESULTS

Hidden layers	**Reg**	**Classification Error**
(500,125)	(0.04)	6.9 , 3.6
(1000,400)	(0.15)	7.5 , 4.6
(400,40)	(0.25)	6.2 , 2.1
(125,400)	(0.03)	8.9 , 4.4

Table 1 : classification accuracy for different hidden layers (cifar, PASCAL voc)

Regression

hidden layers	accuracy
400 200	59.73
1000 250	56.52
125 500	68.5
400 25	61.4

Table 2 : regression *accuracy for different hidden layers (PASCAL voc)*

II. Discussion and Limitations

- Cifar vs pascal error (6.2,2.1), data representation is 5.5 smaller than pascel (228 / 32)
- But cifar excels by its large dataset compared to pascel (5000 / 600) images per class .

III. Conclusions

We implemented different optimization algorithms and found that ***Adadelta*** converges faster than an sgd , momentum , adagrad , rms prop . There is no much difference we tried to change activation functions in the added layers.

Acknowledgment

References

[1] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[3] https://www.kaggle.com/c/dogs-vs-cats

[4] https://github.com/BVLC/caffe/wiki/Model-Zoo