Machine Learning Nanodegree Capstone Project

S.Mohana Krishna

Definition

Project Overview

In computer vision, Image segmentation is the process of partitioning into multiple segments (sets of pixels, also called super pixels). The goal is to simplify the representation of an image to make it more meaningful and easier to analyze. More precisely, Image segmentation is the process of assigning a label to every pixel in the image such that pixels with the same label share certain characteristics. If the Image segmentation task involves identifying semantically meaningful regions in an image i.e. belonging to same object class, then it is called semantic segmentation and is an active area of research in computer vision.

Traditional methods for semantic segmentation used hand crafted features like SIFT, HoG etc. with classification algorithms like SVM, Random Decision Forest. Recently deep learning methods have become more become prevalent and most of the state of the art algorithms use these methods. For this project we follow the approach discussed in `*DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*`[1]. The research tries to solve 3 challenges faced in the application of Deep Convolution Neural Networks to semantic image segmentation: (1) reduced feature resolution, (2) existence of objects at multiple scales, and (3) reduced localization accuracy due to DCNN invariance. My personal motivation behind choosing semantic segmentation is the challenging nature of the problem and its wide range of applications in computer vision tasks like self driving cars, understanding aerial imagery, object detection etc.

Dataset used for training and validation of the project is augmented PASCAL VOC dataset, available at
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/semantic_contours/benchmark.tgz

Dataset used for testing the model is the PASCAL VOC dataset, available at
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval_11-May-2012.tar

Problem Statement

The aim of the project is to understand and implement semantic image segmentation using Deep Convolutional Nets and Atrous convolutions. Semantic segmentation is the task of assigning a label(class) to each and every pixel in the image, and dividing the image into semantically meaningful parts.

Deep Convolutional Neural Networks (DCNNs) face with an issue of reduced spatial resolution due to repeated max pooling layers. In this project we try to address this by using atrous convolutions. We finetune a pre-trained VGG16 network, trained for the task of object recognition, with some minor modifications such as reducing the number of filters from 4096 to 1024 in the last layer, omitting last pooling layers. We try to optimise the loss function which is the sum of cross entropy terms at each spatial position in the DCNN's output matrix. If the input is 500*500*3 matrix , at this stage we have an output matrix of 500*500*21 with each value in the third dimension denoting the probability of the pixel belonging to that class. We take an argmax over the output to get a matrix of 500*500 with each position having the label of the class the pixel belongs to, which can later be converted to an RGB image.

Metrics

We use *mean IOU* (Intersection over Union), as an evaluation metric for the task of semantic segmentation. *IOU* can be mathematically defined as follows:

$$IOU = tp/(tp + fp + fn)$$

Where tp – number of true positives
fp – number of false positives
fn – number of false negatives
This calculation is done for each semantic class in the output, and the mean is taken over all the classes to get the desired *mean IOU*.

Analysis

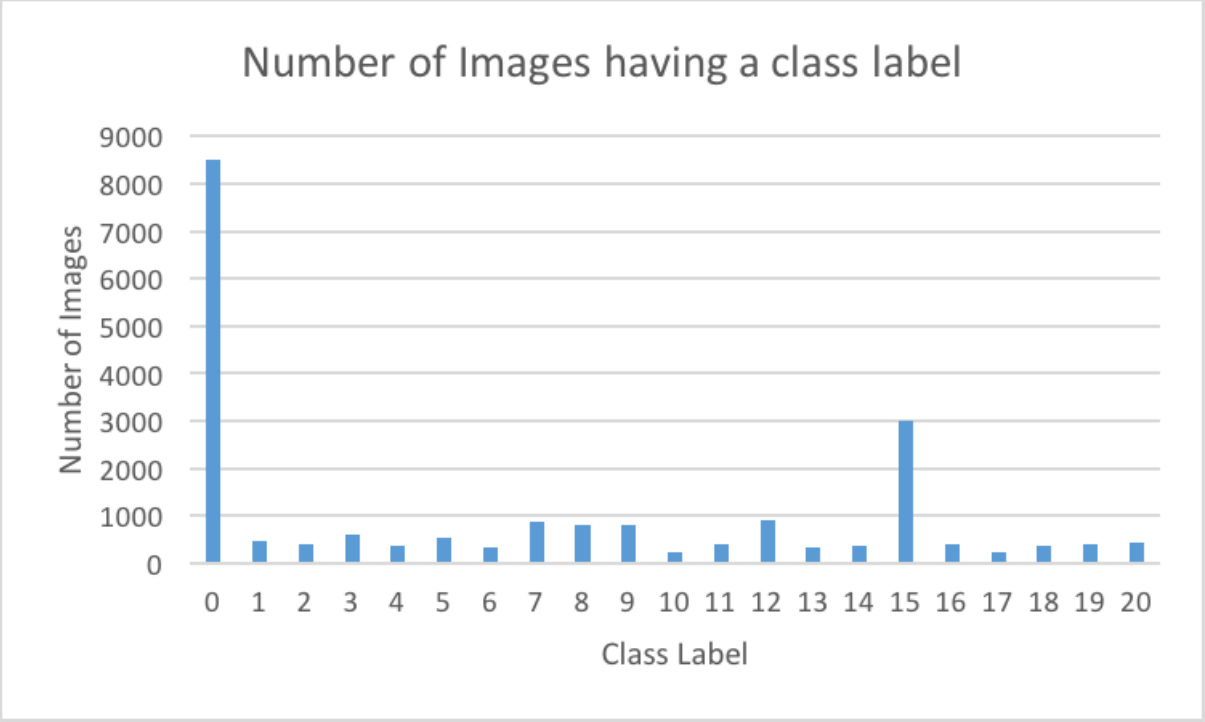
Dataset

For this project we use publicly available augmented PASCAL VOC dataset. There are a total of 11355 images in the dataset of which 8498 images are categorised as training set and

2857 images as validation set. Images are all of different shapes, ground truths are provided in .mat files, which are converted to png files. Each pixel in the ground truth images is labelled among the following 21 classes (including background) (0=background, 1=aeroplane, 2=bicycle, 3=bird, 4=boat, 5=bottle, 6=bus, 7=car, 8=cat, 9=chair, 10=cow, 11=dining table, 12=dog, 13=horse, 14=motorbike, 15=person, 16=potted plant, 17=sheep, 18=sofa, 19=train, 20=tv/monitor).

Exploratory visualisation

The occurrences of classes in the training data is as follows:



It can be seen that

- Background pixels are present in every image, hence the frequency of background is highest,
- ‘person’ class occurs second highest number of times, others occur in less than 1000 images.

Algorithms and Techniques

With the increase in the available data and high processing power DCNN’s have become very popular and are shown to perform exceedingly well in image classification and recognition tasks. In our project we use VGG16, trained in the task of image classification is re-purposed

to the task of semantic segmentation by (1) transforming all the fully connected layers to convolutional layers and (2) increasing feature resolution through atrous convolutional layers.

DCNN Architecture

Input

Augmented PASCAL VOC dataset, has RGB images of different sizes, we preprocess the images and reshape them to a standard size of 500x500x3.

Convolutional and Atrous Convolutional Layers

A convolution layer is indicated by the fig 1 and is defined by the following parameters

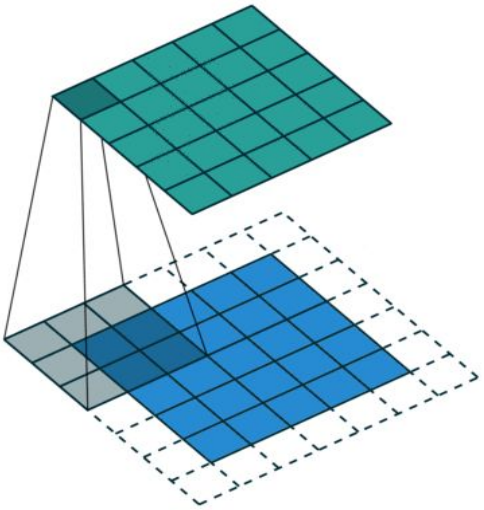


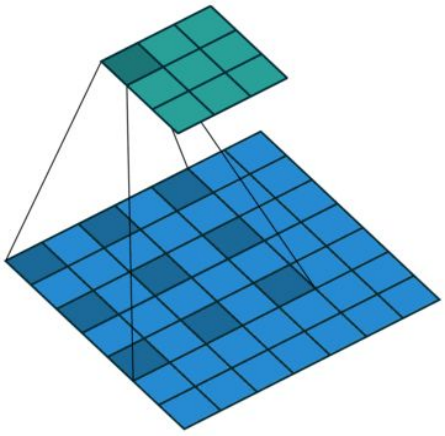
Fig 1- 2D convolution using a kernel size of 3, stride of 1 and padding

Kernel Size : It defines the field of view of the convolution, a normal choice as shown in Fig 1 is 3.

Stride : The stride defines the step size of the kernel when traversing the image. While its default is usually 1, we can use a stride of 2 for downsampling an image similar to MaxPooling.

Padding : The padding defines how the border of a sample is handled. A (half) padded convolution will keep the spatial output dimensions equal to the input, whereas unpadded convolutions will crop away some of the borders if the kernel is larger than 1.

Atrous (Dilated) convolution introduce another parameter called **Dilation rate.**



Dilation rate : It defines the spacing between the values in a kernel, a 3x3 kernel with dilation rate 2 would have a field of view as a 5x5 kernel, as shown in the fig 2.

Fig 2 - 2D convolution using a kernel size of 3, dilation rate of 2 and no padding

Relu Layer

ReLU is the abbreviation of Rectified Linear Units. This layer applies the non-saturating activation function **f(x)=max(0,x)**. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

Other functions are also used to increase nonlinearity, for example the saturating hyperbolic tangent **f(x)=tanh(x)**, **f(x)=|tanh(x)|**, and the sigmoid function **f(x)=(1+e^{-x})^{-1}**. ReLU is often preferred to other functions, because it trains the neural network several times faster without a significant penalty to generalisation accuracy.

Pooling Layer

This layer contains both the activation operation that is applied to each elements after convolution and subsampling of the data after activation. Activation operation is typically a nonlinear operation that is applied to each elements of convolution output such as max(0, *x*)(which is also called Rectifier Linear Unit) or etc where x is the input data. Activation operation does not change the sizeof the input. Subsampling is applied after activation that reduced the size of the input to typically 1⁄2 at each dimension. Window size that is used during subsampling is also 2D such as 2x2 or 3x3. If the input data size is (W,H) then the size after pooling will be (W/2,H/2) if 1⁄2 subsampling is used.

Output

For an Input of size 500x500x3,output of the network is a matrix of dimensions 64x64x21, where 21 is the number of classes possible in the segmented image.

Benchmark Model

The benchmark model for our project is the DeepLabCRF-LargeFOV model as proposed in the `*DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*`. The benchmark model does a semantic segmentation task using Deep Convolutional Neural Networks with a combination of Atrous convolutions and finally couple the output with Fully Connected CRFs to increase the localisation accuracy of the output.

Methodology

Data Preprocessing

Augmented PASCAL VOC dataset has a total of 11355 images, the following steps are taken in the preprocessing stage

- mean of red, blue and green of the input images in the whole dataset is taken and resulting mean is subtracted from all the images.
- Input images are resized to a standard size of 500x500x3.
- Ground truth images in the dataset are also reshaped to size 500x500 with value at each position denoting the class of the pixel.
- The modified input images are saved along with the corresponding ground truth images in batches of 16.

Implementation

Network Architecture

After preprocessing, the network is fed with input images of dimension 500x500x3. The network of 16 layers is divided into 8 blocks, with optional pooling and dropout after each block.

- First block consists of 2 convolution layers with 3x3x64 filters and relu activation layer, followed by a max pool layer with stride 2.
- Second block consists of 2 convolution layers with 3x3x128 filters and relu activation layer, followed by a max pool layer with stride 2.

- Third block consists of 3 convolution layers with 3x3x256 filters and relu activation layer, followed by a max pool layer with stride 2.
- Fourth block consists of 3 convolution layers with 3x3x512 filters and relu activation layer, followed by a max pool layer with stride 1.
- Fifth block consists of 3 atrous convolution layers with 3x3x512 filters, with a dilation rate of 2 and relu activation layer, followed by max pool layer with stride 1 and an average pool layer with stride 1.
- Sixth block consists of an atrous convolution layer with 3x3x1024 filters, with dilation rate of 12 and relu activation layer and a dropout layer.
- Seventh block consists of a convolution layer with 3x3x1024 filters , a relu activation layer and a dropout layer.
- Eight block has the final convolution layer with 3x3x21 filters

In our implementation we use a 25% dropout rate.

Environment

To set up the environment for training, an instance of a virtual machine on google cloud, with a 56 GB RAM, 150 GB hard disk, 1 GPU and ubuntu OS is configured. All the required libraries are installed, along with tensorflow, scipy, numpy and other dependencies. Augmented PASCAL VOC dataset is downloaded.

Training the network

We use one hot encoding on our ground truth images, so the ground truth images of 500x500 are converted to 500x500x21 with 1 in the channel representing the class of the pixel and 0 otherwise. This is resized to the same size as that of the output obtained from the last stage of the network. Our loss function is the sum of cross entropy at each spatial position, and we optimise our loss function using stochastic gradient descent algorithm.

Stochastic gradient descent algorithm is a variant of gradient descent where we use a small random sample of data at every iteration in order to find optimum weights of the CNN. One of the important parameters of gradient descent is the learning rate which is typically represented by α. Basically, large α helps gradient descent to converge faster but with decent quality of weights, while small α has lower converge speed but achieves better weight.In our implementation we use a low learning rate of 1e-4, with which we get descent results.

To avoid overfitting, we use dropout layers in our network. Dropout is a regularisation technique where at each stage individual nodes are dropped out with a probability ***1-p*** and kept with a probability ***p***.

Initially I faced a lot of issues with the training process as the program used to crash or google used to shutdown the VM for maintenance. To overcome that problem I used tensorflow's **Supervisor**, which is basically a training helper that creates checkpoints and summaries. In case our program crashes, Supervisor initialises the program from the most

recent checkpoint. We train the model on 8498 images for 50 epochs, which takes around 2 days to complete on the configured VM.

Refinement

We use meanIOU as the metric to measure the performance of the model. In my initial training phase I used PASCAL VOC dataset, which contains 1464 training images. After training when the model was run on the validation dataset of 1449 images, the meanIOU was 0.12 which indicated that the model had overfit. The network is huge with millions of parameters and 1469 images of training data was very less for the model, so I decided to increase the training data and so used augmented PASCAL VOC dataset. After increasing the data the model did not overfit and the meanIOU score on the validation set of 2857 images, it was 0.52, and on a test set of 1449 images, it was 0.573.

Results

Model Evaluation and Validation

In this project we use **mean IOU** as the evaluation metric. Mathematically IOU is defined as follows

$$IOU = tp/(tp + fp + fn)$$

Where tp – number of true positives
fp – number of false positives
fn – number of false negatives

In the implementation
true positives for a class can be calculated as the number of pixels which belong to the particular class in the ground truth image and are classified correctly in the predicted image.

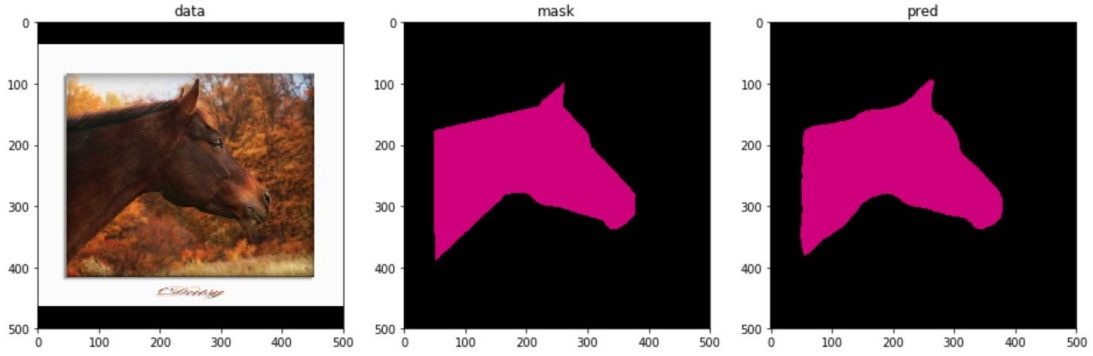
false positives for a class can be calculated as the number of pixels which are classified to belong to the class in the predicted image, but do not actually belong to the class as per ground truth.

false negative for a class can be calculated as the number of pixels which belong to the particular class in the ground truth image and are classified incorrectly in the predicted image

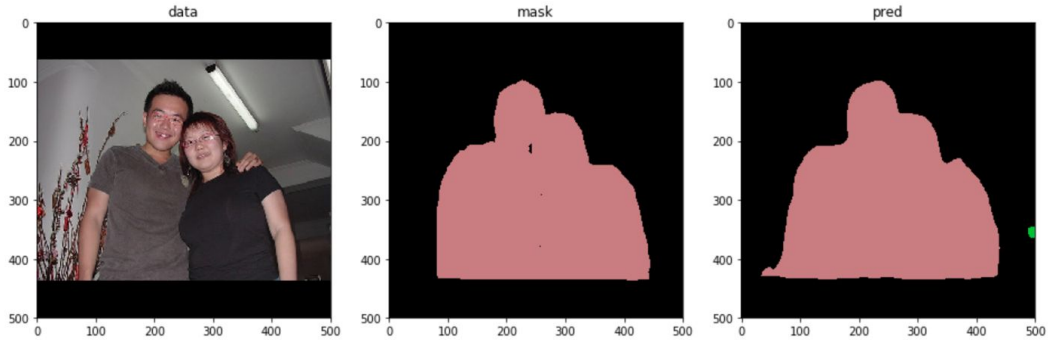
With mean IOU as the evaluation metric, we can safely say that the model has generalised well, because on the validation set of 2857 images the model performed well with a mean IOU of 0.52. For testing, I downloaded the standard PASCAL VOC dataset and used the validation set as the testing set and model gave a mean IOU of 0.573 on the test data. A better score on the test data than validation data shows that the model has generalised pretty well.

Justification

The results obtained are less than the benchmark model. The reason for that is that in the benchmark model, Conditional Random Fields are used, which as mentioned in the paper improve the localisation accuracy of the model. Also due to the restricted resources available I could only run the model for 50 epochs, where as the benchmark model was trained for 21k epochs. The results of the trained model on the training dataset and validation dataset are mentioned below:



Output on the input from training dataset



Output on an input from validation dataset

Conclusion

Reflection

The aim of the project was to implement and achieve semantic segmentation, which is the task of labelling each pixel in the image such that pixels with the same label belong to same class/Object. We started with the idea that Image classification and segmentation are related tasks and so knowledge of Image classification can be used for the task of segmentation. So we started with VGG16 network, pre trained for the task of Image classification. We modified the network, removed the fully connected layers with convolutional layers, to

handle the issue of reduced spatial resolution in the output due to repeated max pooling, we used atrous convolution. The final output of the network indicated the probability of a pixel belonging to each of the 21 classes. We used *argmax* to get the class with highest probability, and used bilinear interpolation to resize the output to the original input size, which can now be decoded based on the label of the pixel and converted to RGB image.

One interesting thing to learn was the idea of transfer learning, where we could take models trained on similar tasks and finetune the model to use the knowledge for our tasks. This is a very useful technique as it saves a lot of computational time and resources spent on the task.

Another interesting thing was the use of atrous convolutions. Reduced spatial resolution of the output of a CNN was a major issue, but atrous convolution helps solve the problem without any increase in computational complexity.

Improvements

Scene understanding has been an important task in computer vision, because of its wide use case like that of self driving cars, and semantic segmentation is an important part of scene understanding.
The model proposed in the project could be improved, by allocating proper resources and allowing the model to train for a large amount of time. Resources were a limitation for this project.The model can also be improved by the use of conditional random fields as implemented in the benchmark model, to improve the localisation accuracy of the model.
A recent advancement in this area is the use of adversarial networks, which was shown to produce state of the art results, in *Semantic Segmentation using Adversarial Networks* [2].

References

[1] DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs
[2] *Semantic Segmentation using Adversarial Networks*
[3] https://en.wikipedia.org/wiki/Convolutional_neural_network
[4] https://www.tensorflow.org/