Domain Background

Traditionally images are one of the most important forms of communication. Beginning from the early civilization, humans have used various pictorial representations. It becomes immensely important due to the fact that our human brain is able to recognize familiar images within 100 milliseconds. If we consider other living creatures like fishes, snakes, birds, animals they also can differentiate between different things like food, enemy, terrain etc. We have always been interested in understanding how we see & distinguish images & can a machine understand images.

Task of understanding an image can be divided into a number of components like labeling, segmentation, clustering, box labeling etc. We strive to achieve image segmentation in our project. Our brain is inherently configured to distinguish between background & objects. But for a machine, images are just an array of numbers. Inferring any relevant details from those numbers is a challenge and can lead to interesting results. From medical image analysis to face id detection, image segmentation can be used for critical as well as recreational purposes. A lot of research is actively going on in this area as this lays the foundation of deeper understanding of images like recognizing person, place, etc. in an automated manner. Fully Convolutional Networks for Semantic Segmentation [1] focuses on how to implement image segmentation effectively. My personal motivation for choosing this as project is the self-driving cars. Image segmentation is the first & foremost requirement for making self-driving cars success. I seek to understand how we can achieve state of art performance in image segmentation.

Problem Statement

The aim of the project is to get an understanding of image segmentation using Convolutional Neural Network and create model that can determine semantic boundaries. Image segmentation is task of dividing image into multiple parts. The goal of segmentation is to simplify the image into something that is more meaningful & easier to understand. For example in an image containing humans if we replace all pixels of human with one value & all other pixels with

another value, it becomes immensely easy to find human in that modified image. We are recognizing 20 such classes like bicycle, bird, boat, chair, cow, dog, human, etc. in the project.

Datasets & Inputs

Dataset to be used for the project is VOC Pascal dataset from the *Visual Object Classes Challenge 2012 (VOC2012)* [2]. It consists of ***training image*** as input image and the ***class segmentation image*** as output image***.***

In class segmentation image each pixel indices correspond to classes in alphabetical order (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).

Dataset can be directly downloaded from the PASCAL VOC website .

Solution Statement

To begin with the understanding of the contents of an image, researchers have started from very basic concepts like detection of edges and shapes to labeling pixels of different objects present in an image. While the former tasks have been worked on since early beginning of computer vision, a lot of progress is yet to be made in understanding the image as we human do. One of the important bottlenecks in this direction is when we process image in a sequential manner, we lose its locality information. We here in the project propose to use CNN to effectively capture the spatial knowledge of an image and utilize it to gain insight about the image. We have to categorize pixels of the image into 20 different classes and background. We plan to design a CNN architecture which takes H*W (say 320*160) sized RGB image as input. For the output we as we have 20 classes & 1 class for the background, we are basically doing one-hot encoding with a vector of dimensionality 21. So each pixel in output is 21 sized vector in which only one element is ‘1’ corresponding to the particular class & rest all are zero. Our input is 320*160*3 and our output is

320*160*21. In the last layer we are taking ***softmax*** with 21 filters. Each filter will activate for a particular class pixel in an image. After getting the 21 filter outputs we are taking argmax to find the class which is most likely to represent category of that pixel in the image.

Benchmark Model

The benchmark model used for the project is FCN-8s as proposed in Fully Convolutional Networks for Semantic Segmentation [1] . The benchmark model performs similar task of image segmentation on various datasets like VOC 2011, VOC 2012, NYU-Depth v2, etc. Though we in our project are using solely VOC 2012 for all training & testing purposes. The benchmark model proposes a skip architecture that combines semantic information from a deep, coarse layer with appearance information from shallow, fine layer to produce accurate and detailed segmentations.

Evaluation Metrics

In the project we will use ***categorical accuracy*** and ***mean IU*** as the evaluation metrics. As we have transformed the segmented image output to one-hot encoded array of size 320*160*21, so for each pixel we have 21 element array. Only one element out of the 21 elements will be 1 & rest will be zero. The first step for obtaining both the metrics is converting the 21 element array to a single number representing the class. Argmax is used to find the highest class probability and the corresponding class.

For categorical accuracy we are using pixel-wise accuracy i.e. number of number of pixels correctly predicted out of total number of pixels.

Mean IU is the ratio of true positive and (true positive + false positive + false negative).

Project Design

The architecture of project is motivated by the success of Convolutional Neural Networks like VGGNet, AlexNet that can very well predict the class of the object. But here in the project we strive to have a pixel-wise classification of the image. To begin with the pre-processing of images, we aim to resize both input & segmented images to a fixed size of H*W (say 320*160) to reduce the complexity of the network and preserve the features in an effective way. Our input image now becomes of size 320*160*3. We will then subtract the mean of the dataset from each image (RGB channel wise) to get zero centered data. For segmented output image as we have 20 classes & 1 background we will convert each image into 320*160*21 array. We will then feed this to a Convolutional Neural Network. The architecture of network would be a combination of Conv, MaxPooling , ConvTranspose and Activations layers. Keras inbuilt layers library will be used. We will then analyze the number of filters in each layer depending on the understanding of how much context do we want to retain from the input in each subsequent layers keeping in mind the complexity of the network. Various Normalization layers may be used to get better results. Last layer would be with 21 filters & softmax activation, so that each filter can predict the probability of a particular class for a pixel. As proposed in the paper[1], we will try skip layers depending on the fine tuning we want in our output. We will try to understand the effect on the number of skip layers used & its implications on the overall model predictions.

Referenes :

[1] E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, April 1 2017.
doi: 10.1109/TPAMI.2016.2572683

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn,and A. Zisserman.The PASCAL Visual Object ClassesChallenge 2012 (VOC2012) Results.http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.htm