

Machine Learning Engineer Nanodegree

Capstone Proposal

Alberto Vigata

Proposal

Domain Background

This proposal deals with the creation of a system to detect traffic signals in an image, and in turn, detect the state of such signals. In order to create this system we'll need the help of tools in the realm of computer vision.

Historically image semantic segmentation and inference has been an active field of study, but it wasn't until the advent of deep neural networks when breakthroughs did happen. In particular, [this paper released in 2014 \(2014, Fully Convolutional Networks \(FCN\) by Long et al.\)](#) spearheaded a new successful approach in the field. Since then, multiple new architectures have improved in the segmentation accuracy, a good review can be seen [here](#)

As a personal note, having been in touch with image segmentation for over a decade, and being familiar with older and less sophisticated techniques it's of great interest to apply state of the art solutions to the problem to see what can be achieved.

Problem Statement

The problem to be solved consists in creating a system that is fed real world images and it's able to identify traffic signals in the images and then in turn, identify the state of such signal as *green*, *yellow* or *red*. Such system could be used in autonomous systems to gain awareness of right of way.

The inputs to the system are single images, and images only. The outputs of the system will be a logical output affirming if a traffic light is present, and if present what's the current state of the traffic light, *green*, *yellow* or *red*.

Because we'll have a ground truth from our testing images we'll be able to easily identify if our system is performing as expected or not.

Datasets and Inputs

I'm planning to use the COCO dataset, publicly available at <http://cocodataset.org>. COCO dataset is a large dataset that has a lot of the features we need, in particular object segmentation. It's also a large dataset of 330K images with over 1.5 million objects represented.

One the categories that the dataset classifies for is "traffic light" which will be of interest in this project. We will use the traffic light images for classification, and in addition we will create labels with the signal state.

Solution Statement

The solution to the problem will be a system that correctly identifies signal and their status as *green*, *yellow* or *red*.

Benchmark Model

As a benchmark model we will use the evaluation metrics defined below trying to achieve the highest values possible. Given the best possible values for precision and recall are 100% (or 1) that will be our goal benchmark. Achieving values close to 100% for both precision and recall will signify our system is working properly and it's identifying the signals correctly.

Evaluation Metrics

Our evaluation metric will be that of recall and precision.

```
tp = true positive
fn = false negative
fp = false positive

we then define
      tp
precision = -----
      tp + fp

      tp
recall =  -----
      tp + fn
```

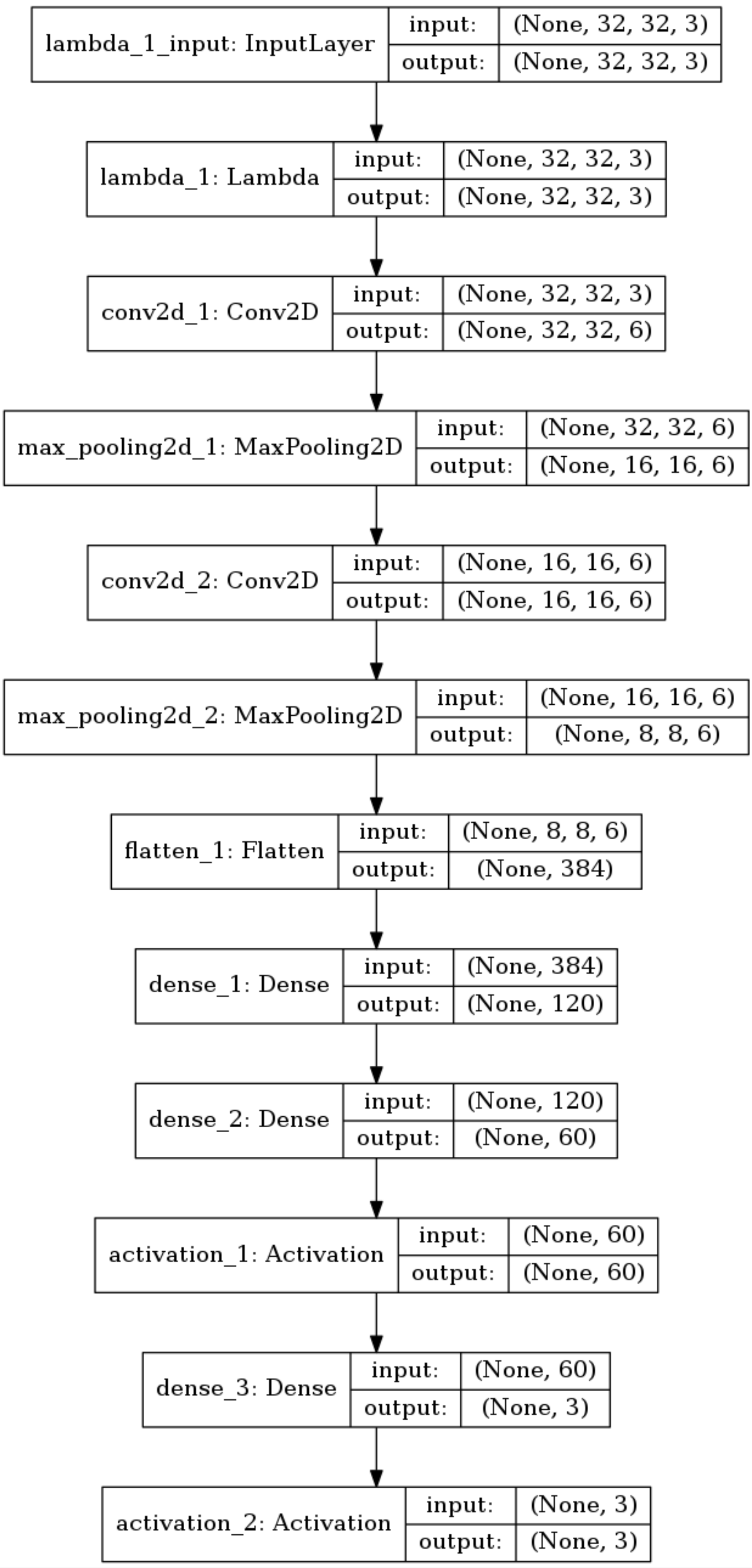
We'll have one metric for precision and recall regarding the presence of a traffic light. Then additional metrics for the state of the light *green*, *yellow* or *red*.

Project Design

Workflow

The current plan of attack for this project is the following:

- A segmentation deep neural network will be used on the COCO dataset to detect the traffic signal and its location. Research will be needed to find the appropriate network for this task, but the FCN metioned [here](#) look promising.
- A second deep neural network, a simpler CNN, will be presented with the cropped versions of the traffic signal images to predict status of signal. Given the simplicity of this task we should use a smaller simpler well known CNN. I propose a simple Lenet style CNN with a model similar to this:



Both nets will be trained separately. The first one against the COCO dataset with the traffic signal as output. The second one with images of the traffic signals scaled and normalized with 3 outputs to determine their color.