**Image Detection – Simpson's characters**

**Assignment 5 - Analytics and Algorithms (Mid Sprint3)**

As part of our core project development scope, the team decided to attempt understanding and developing algorithms for 6 Neural Net Architectures. This would allow us to compare and contrast the accuracy, performance and other metrics associated with these various architectures. These are the 6:

1. Convolutional Neural Network (CNN)
2. Faster R-CNN with different Hyper parameter
3. YOLO (You only look once)
4. Single shot Multi-box detection (SSD).

Before diving into these architectures, there are some basic concepts about Deep Learning that we had to get familiar with.

Activation functions are responsible for transforming the weighted inputs to a neuron and determining if the resulting output is ‘activated’ or not. The simplest type of an activation function is Step function (Perceptron) which can input and output binary values only. A step above is a Sigmoid, which can output any value between 0 and 1. Tanh based activation functions are similar to Sigmoids, but faster and can return values between -1 and 1(scaled sigmoids). Rectified Linear Unit (ReLU) returns 0 if the if the output is negative else returns the output.

Cost functions are used to quantify difference between output received from a neuron versus output expected to be received. Quadratic cost function is similar to the Mean Squared Error metric used in Linear Regression models. The other type of cost function is Cross-Entropy. This log-based function enables faster learning when difference between received and expected values are high.

To learn from a cost function and make changes to rectify the error, a gradient descent approach can be used to reduce or eliminate the magnitude of error. Learning rates that determine the size of step to correct the error, Batch sizes that sample input data to feed one run of a network, second order calculations that can use acceleration or momentum of previous steps to adjust size of next step are all Gradient descent related features that can be used to tune a model.

To reduce overfitting a model, techniques like L1/L2 based normalization, Dropout (where a subset of neurons are dropped), artificially expanding data can be used.

To initialize weights, bias and other outputs, Glorot-normal and Glorot-uniform values can be used.

And various types of layers can be used to build a neural network, like for e.g. Dense layer, where all neurons in one layer are fully connected to neurons in the next layer. Softmax layer that outputs a class probability score based on generated weights, Max Pooling that can be used to reduce the size of an image by grouping pixels, Flatten that can modify a 2D or 3D array to a single dimensional structure and Convolutional layer that can learn shapes irrespective of location.

## Description of the various algorithms:

### CNN

CNN is a multilayer perceptron specially designed for recognizing image contents. This network structure is highly invariant to translation scale slant or covariant deformation. It is the basic algorithm for other algorithms that we needed use in our research and in our project. We used it to do the image classification. CNN though the input image and classifies all dataset into a specifies category the computer treats the input image as an array of pixels. According to the image resolution, we can see h \* w \* d (h = height, w = width, d = size) (Prabhu, 2018).

For the deep learning CNN model, each image will go through a series of convolution layers with kernel, pooling and full connected layer (FC). We can add many convolutional layers and flatten the output and sent to a FC Layer. At last though an activation function to output the class and all classifies images.

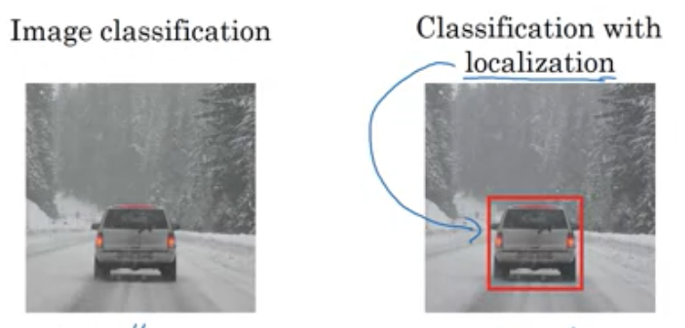
### R-CNN

R-CNN combines region proposals with CNNS, it can use high-capacity convolutional neural networks to bottom up region in order to locate and image segmentation (Girshick, Donahue, Darrell, & Malik, 2014). R-CNN can also be understood as regions with CNN features.

### YOLO: You Only Look Once:

This is an architecture that tries to overcome some of the limitation of segmentation / region based architectures for Image Detection. This algorithm works with Conv Nets that perform Image Classification and Image Localization.

A Conv Net with can perform Image classification which in below case can be trained to return a class “car”. Localization is where in addition to Classification the algorithm returns coordinates associated with where in the image a particular object was detected. This is also referred to as Bounding Box. This Bounding Box consists of 4 sets of values: bx, by, bh and bw. Bx and By are the (x,y) coordinates of the left top corner of box, bh is height and bw is width of object. The output y in case of localization consists of these elements: first element a probability associated with classification, next 4 elements are bounding box related and the rest are binary class variables indicating type of object found.



Localization can also be extended for Landmark detection, where multiple coordinates associated with a part of an object – for e.g. eyes of a person can be returned by the algorithm.

YOLO algorithm works by splitting an image into a grid. Each cell is evaluated for midpoint of an object and if found, that cell is responsible for arriving at bounded box for that object. One Conv layer is used for all cells of the grid – this introduces efficiency, speeds up detection and thereby YOLO can be used for real time video-based detection. In below example, image is of 100x100x3 size, output will be 3x3x8 (3x3 because we chose that as grid size) and 8 to account for probability, 4 bounding box values, and 3 classes (this algo tries to classify 3 types of objects). If detected image spans beyond current cell, height and width of detected object can return values > 1.

Evaluation of object localization is done via IoU – Intersection Over Union calculation which tries to determine how much of the image is present in the calculated box boundaries. Typically IoU values < .5 are discarded. This calculation is also used to aid in Non max suppression where multiple detections of the same object are resolved and best box is retained.

In scenarios where midpoint for 2 or more objects exist in the same cell, anchor boxes can be used. Anchor boxes are predefined shapes used by the algorithm to fit objects in. The ‘y’ output is extended to return all object matches found (Number of Anchor boxes = Number of possible objects that can be detected)

(<https://www.coursera.org/learn/convolutional-neural-networks?specialization=deep-learning>)

SSD

The SSD approach is based on a feed-forward convolutional network that produces

a fixed-size collection of bounding boxes and scores for the presence of object class

instances in those boxes, followed by a non-maximum suppression step to produce the

final detections. The early network layers are based on a standard architecture used for

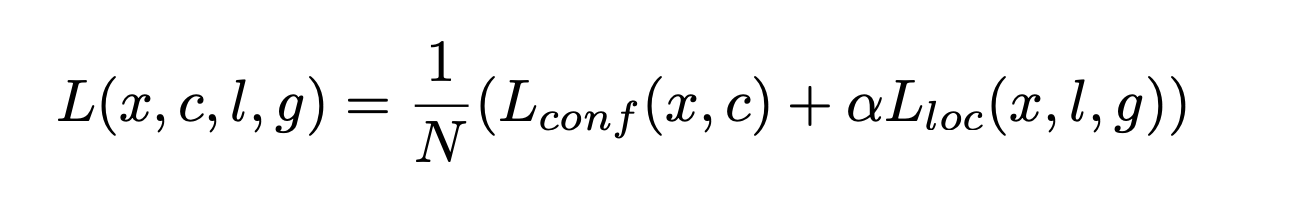
high quality image classification (truncated before any classification layers), which is a base network. We then add auxiliary structure to the network to produce

Detections is based on the following key features:

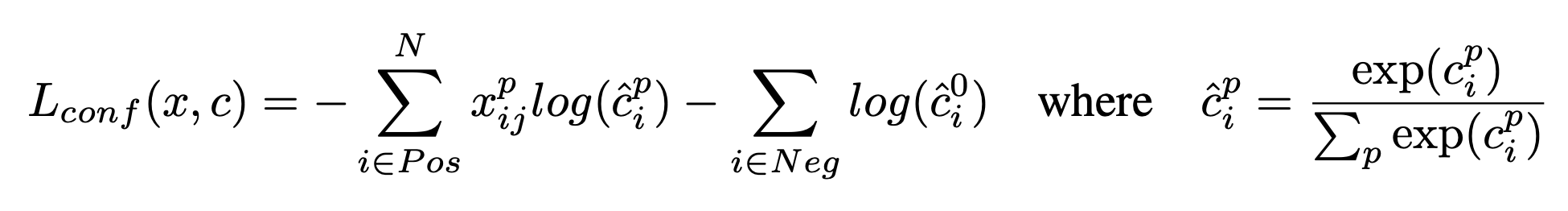
**Multi-scale feature maps for detection** We add convolutional feature layers to the end of the truncated base network. These layers decrease in size progressively and allow predictions of detections at multiple scales. The convolutional model for predicting detections is different for each feature layer (cf Overfeat[4] and YOLO[5] that operate on a single scale feature map).

**Convolutional predictors for detection** Each added feature layer can produce a fixed set of detection predictions using a set of convolutional filters. These are indicated on top of the SSD network architecture.

**Training objective** The SSD training objective is derived from the MultiBox objective but is extended to handle multiple object categories. The overall objective loss function is a weighted sum of the localization loss (loc) and the confidence loss (conf):

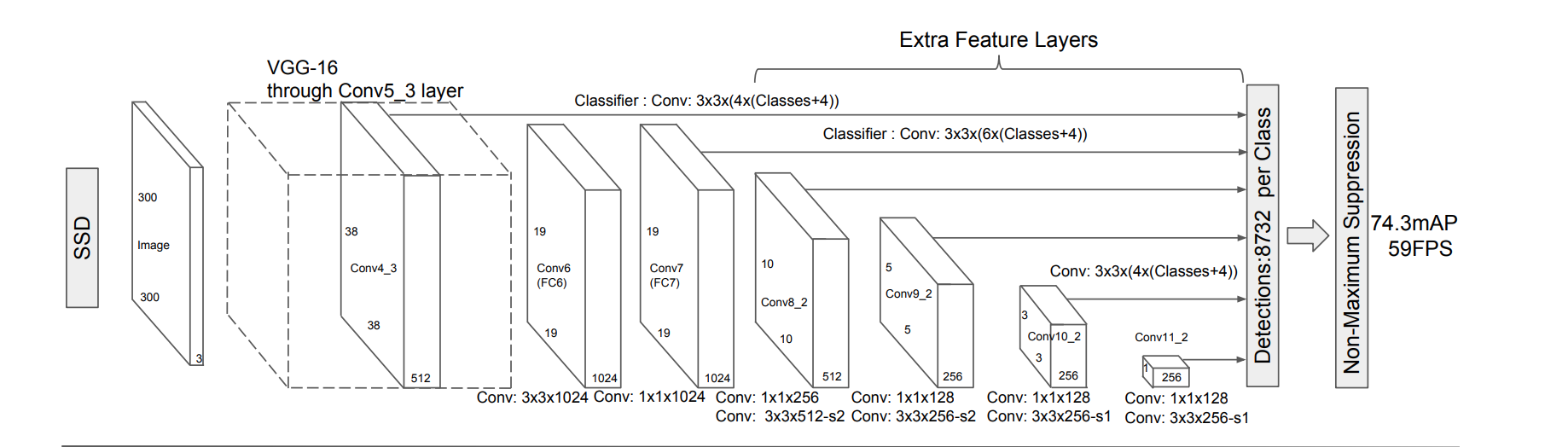


The confidence loss is the softmax loss over multiple classes confidences (c).

****

and the weight term α is set to 1 by cross validation.

**SSD Architecture**



# References

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation.* Berkeley: UC Berkeley.

Prabhu. (2018, 3 4). *A Medium Corporation[US]*. Retrieved from Understanding of Convolutional Neural Network (CNN) — Deep Learning: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

**SSD** <https://arxiv.org/pdf/1512.02325.pdf>