Internship Report

Submitted to:

CTO Laboratory
Wipro Digital
Electronics City, Bengaluru

Submitted by:

Raghav Sharma
B. Tech. IIIrd year, Electrical Engineering Department
Indian Institute of Technology, Ropar

Supervisor:

Mr. Yadhunandan US

Start Date for Internship: 21st May, 2018
End Date for Internship: 27th July, 2018

Preface

This report documents the work done during the summer internship at Chief Technology Officer (CTO) laboratory, Wipro Digital, Electronics City, Bengaluru under the supervision of Mr. Yadhunandan US. The report shall give an overview of the tasks completed during the period of internship with technical details. It shall also elaborate on the future work which can be persuaded as an advancement of the current work.

I have tried my best to keep this report simple yet technically correct. I hope I succeed in my attempt.

Raghav Sharma

Acknowledgment

Simply put, I could not have completed this internship without the guidance and support I received from the people working in CTO laboratory. The work culture is really motivating. Everybody is such a friendly and cheerful companion here that stress never comes in way.

I would specially like to thank Mr. Yadhunandan US, Mr. Siva Rao, and Mr. Rahul Siripurapu for their unconditional support. Not only did they helped me in learning new concepts, but listening to their discussions have evoked a good interest in the field of Machine Learning.

Raghav Sharma

Abstract

Automated Machine Learning (AutoML) is the process of automating the end-to-end process of applying machine learning to real-world problems. In a typical Machine Learning application, practitioners must apply the appropriate data pre-processing, feature selection and hyperparameter optimization to maximize the predictive performance of their final machine learning model.

Many of these steps are often beyond the abilities of non-experts, AutoML proposes an Artificial Intelligence based solution to the ever-growing challenge of applying Machine Learning. Moreover, these models often outperform models that were designed by hand.

The main aim of this internship was to automate the training of pva-faster-RCNN architecture (https://github.com/sanghoon/pva-faster-rcnn), based on custom dataset and tweaking the layers of the architecture based on user input.

Wipro Digital

(https://wiprodigital.com/)

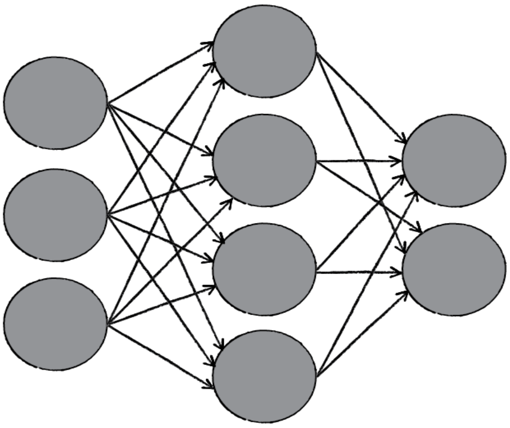
Wipro Digital puts the customer at the center of a multidisciplinary, agile approach: alignment of strategy, design, and technology around the customer journey, enabling innovation of the product service experience through iterative, incremental activations and deliver continual enterprise transformation at speed and at scale.

Being an innovation-led, enterprise transformation partner built for today’s digital challenges, Wipro Digital focusses on the things that matter – insights, interactions, and innovations that make extraordinary things happen for brands, businesses and their customers.

It’s parent company, Wipro is the first software technology and services company in India to be certified for ISO 14001 certification. Its high ethical standards are evident by the fact that in March 2017, Wipro was recognized as one of the world’s most ethical companies by US-based Ethisphere institute for sixth consecutive year.

Machine Learning Basics

(1) Neural Network:



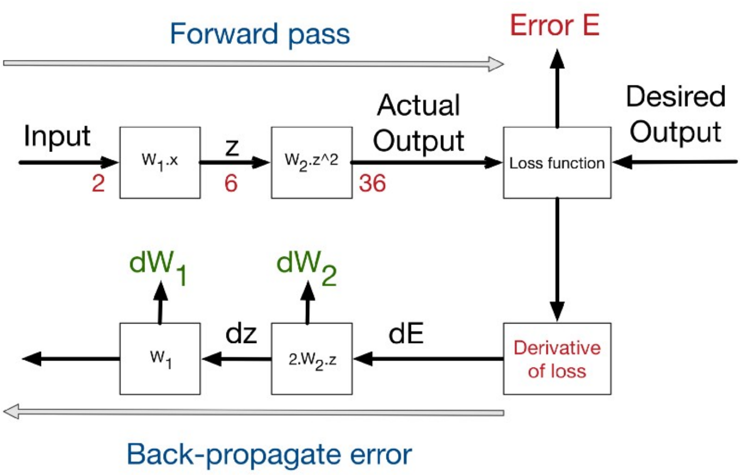
Artificial Neural Networks are computing systems inspired by biological neural networks that constitute animal brains. Such systems “learn” to perform tasks by considering examples, generally without being programmed with any task-specific rules.

The connections between artificial neurons are called ‘edges’. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. Artificial Neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold.

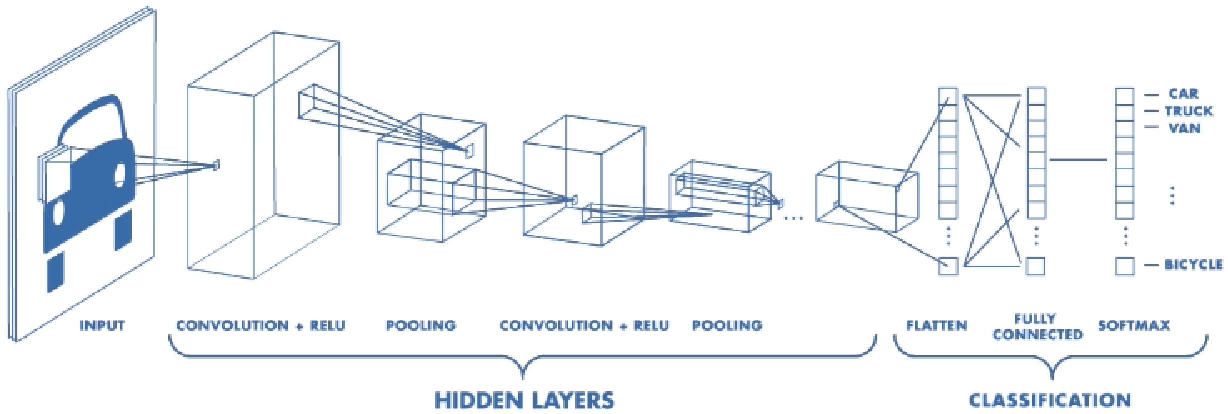
Modern neural networks generally have multiple layers between their input and output, called ‘hidden’ layers. With each layer, the network transforms the data, creating a new representation. When we get to the final representation, the network will just draw a line through the data (which is called a hyper-plane in higher dimensions).

Neural Network primarily operates on the two algorithms:
(a) Forward Propagation: The natural step to do after initialising the model at random, is to check its performance. If we start from the input we have, and pass it through the network layer and calculate the actual output of the model straightforwardly, the process is known as <u>*Forward Propagation.*</u>

(b) Backward Propagation: It is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. This is sometimes called <u>*Backward Propagation of Errors*</u>, because the error is calculated at the output and is distributed back through the network layers.



(2) Convolutional Neural Network:



Convolutional Neural Networks have a different architecture than regular Neural Networks, and this architecture is especially useful for image classification and recognition.

Regular Neural Networks transform an input by putting it through a series of hidden layers. Where every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. The output layer, which is the last fully connected layer, represents the predictions.

In the case of Convolutional Neural Networks, the layers are organised in 3-dimensions: width, height and depth. The neurons in one layer are not connected to all the neurons in the next layer, but only to a small region of it. The final output is reduced to a single vector of probability scores.

In Convolutional Neural Network, the convolution is performed on the input data with the use of a filter/kernel to produce a feature map. Various convolutions are performed on the image, where each operation uses a different filter. In the end, all these feature maps are taken and put together as final output of the convolution layer. The output of the convolution is passed through an activation function (ReLU), so as to make the output non-linear.

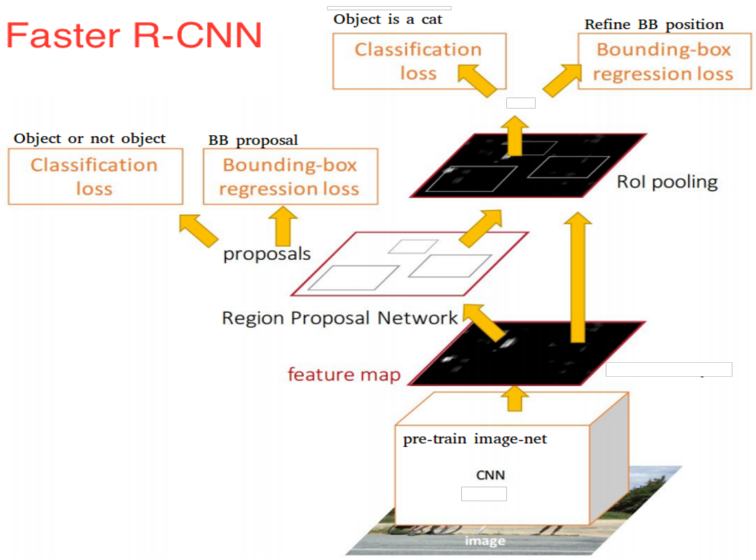
A *Stride* is the size of the step the convolution filter moves each time. If the stride size is increased, the filter will slide over the input with a larger interval, and hence has less overlap between the cell.

A *Pooling Layer* is commonly added after the convolution layer so as to continuously reduce the dimensionality to reduce the parameters and computation in the network. This shortens the training time and controls overfitting.

Hence, while using a Convolutional Neural Network, the four important hyperparameters that are needed to be decided are:
(a) Kernel Size
(b) Number of Filters
(c) Stride
(d) Padding

After the hidden layers, the *classification part* consists of few fully connected layers which can accept only 1-dimensional data. This part in principle is the same as regular Neural-Network.

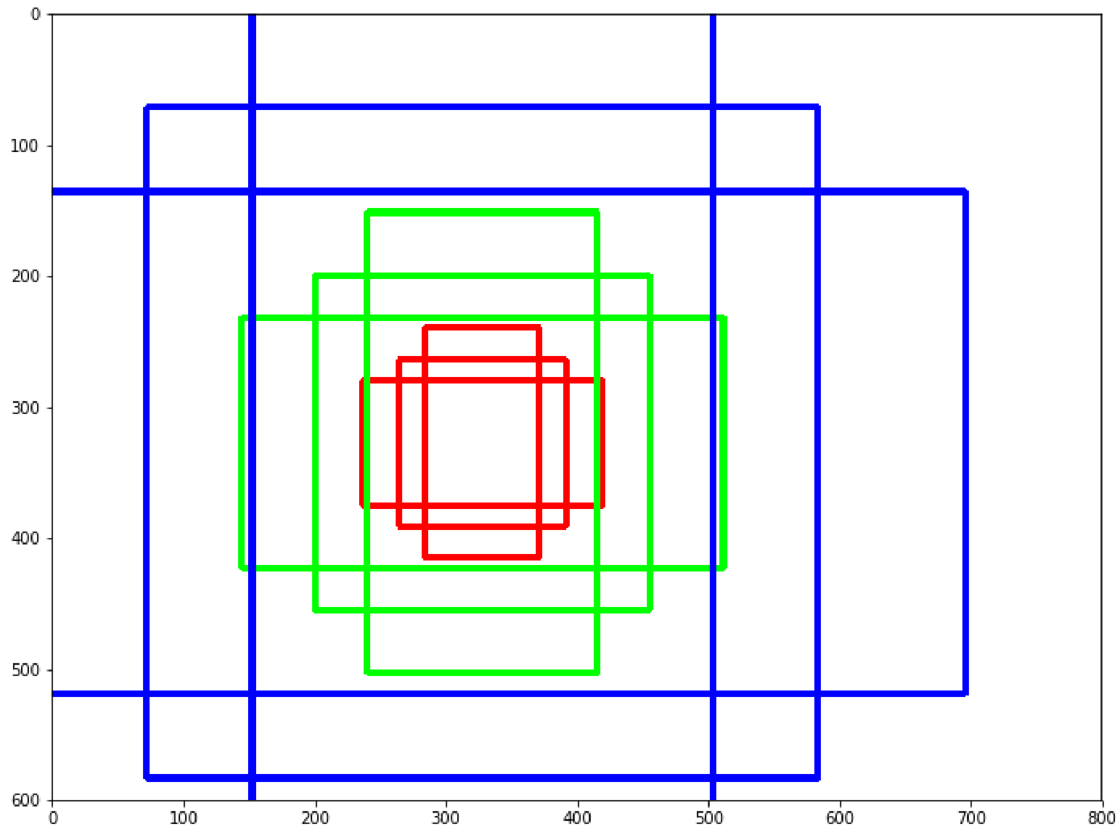
(3) PVANet Architecture:



Based on research paper: https://arxiv.org/pdf/1608.08021.pdf
GitHub Repository: https://github.com/sanghoon/pva-faster-rcnn

Faster R-CNN has two networks working simultaneously: Region Proposal Network (RPN), and a network using these proposals to detect objects. RPN ranks region boxes (called anchors) and proposes the ones which are most likely to contain objects.

Anchors: An Anchor is basically a box, which plays an important role in Faster R-CNN. In the default configuration, there are 9 anchors at a position of an image. The following image shows 9 anchors at the postion (320, 320) of an image with size (600, 800).



Observations:
(a) Three colors represent three scales/sizes: 128x128, 256x256 and 512x512.
(b) The three boxes of a certain color have height:width ratios 1:1, 1:2 and 2:1 respectively.

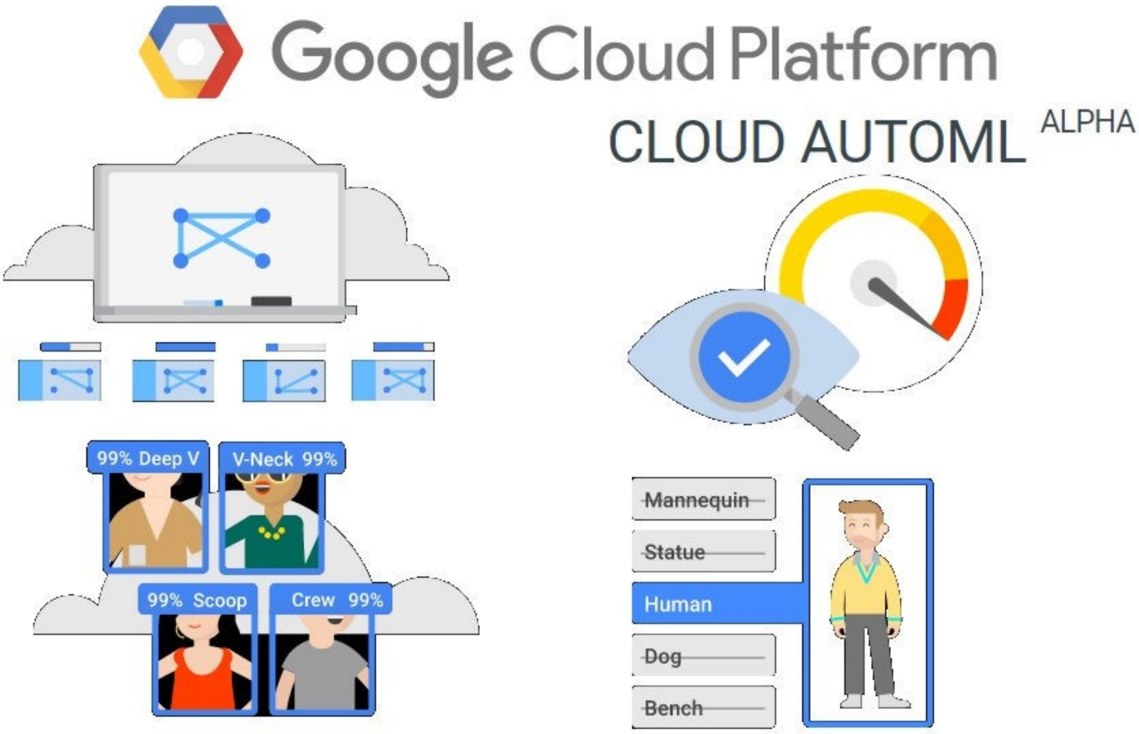
These anchors work well for Pascal VOC, as well as COCO dataset. A neat seat of anchors may increase the speed as well as accuracy.

The output of a region proposal network (RPN) is a bunch of boxes/proposals that will be examined by a classifier and regressor to eventually check the occurrence of objects. RPN predicts the possibility of an anchor being background or foreground, and refine the anchor.

Region of Interest Pooling (ROI Pooling):
After RPN, proposed regions with different sizes are obtained. Different sized regions means different sized CNN feature maps. ROI Pooling reduces the feature maps into the same size, so as to generate an efficient structure to work on features.

Google AutoML

Commercial products like Google AutoML, Data Robot, Auto Sklearn, Auto WEKA, and TPOT shows that training high-quality custom machine learning models with minimum effort and machine learning expertise can be the next big step in the Machine Learning and Artificial Intelligence research.



Cloud AutoML provides its user a simple graphical user interface (GUI) to train, evaluate, improve, and deploy models based on user's data. AutoML makes it possible for developers with limited machine learning expertise to train high-quality custom models. After uploading and labelling images, AutoML trains a model that scales as needed to adapt to demands. It offers higher model accuracy and faster time to create a production ready model.

Google Cloud Vision products are:
(a) <u>AutoML Vision:</u> It enables the user to train his own, custom machine learning model to classify his images based on defined labels.
(b) <u>Vision API</u>: It allows the user to easily integrate vision detection features in an application, like image labelling, optical character recognition, etc.
(c) <u>Vision Product Search</u>: It allows retailers to create a set of products and reference images that visually describe the product from a set of viewpoints.

Objective

Automating the training of PVA-Faster-RCNN on a custom dataset.
Neural Network Architecture link: https://github.com/sanghoon/pva-faster-rcnn

Develope a single program that performs the following functions:
(1) Collecting the datset in the specific format, ie one-one and onto relationship between ".jpg" and ".xml" formats Ex: im1.jpg – im1.xml

(2) Creating a new folder named <project_name>devkit<year>. Creating another folder inside this devkit folder named <project_name><year>. Keeping year as 2017 default, creating folders named 'JPEGImages' and copying all '.jpg' dataset images to this folder. Creating another folder named 'Annotations' and copying all the '.xml' files to this.

(3) Creating a folder named 'ImageSets' and creating 3 subfolders named 'Layout', 'Main' and 'Segmentation'. Creating text files corresponding to training, validation and test data splits of the data. Ex: 'train.txt' – which consists of all names of training set, 'val.txt' – which consists of all image names of validation set etc. It should be made sure that only names are present inside these '.txt' files without full path or extension.

The folder structure, after completing the data setup, should be as follows:

```
|---  pva-faster-rcnn
|    |--   data
|    |    |---  projectName_<devkit>_projectYear
|    |    |    |---  projectName_projectYear
|    |    |    |    |---  JPEGImages
|    |    |    |    |---  Annotations
|    |    |    |    |---  ImageSets
|    |    |    |    |    |---  Layout
|    |    |    |    |    |---  Segmentation
|    |    |    |    |    |---  Main
|    |    |    |    |    |    |--- train.txt
|    |    |    |    |    |    |---  val.txt
|---  Master Project Folder
|    |---  output.txt
|    |---  pva_faster_RCNN.py
|    |---  sample
```

Note:
Master Project Folder has been created manually by the user. It contains the following files:
(a) **output.txt**: It contains the desired labels of images, which are selected by the user.
(b) **pva_faster_RCNN.py:** Source code in python
(c) **sample:** This folder contains the supervised-learning dataset, provided by the user. It contains a mix of '.jpg' and '.xml' files in one-one and onto relationship. ',jpg' files contain the image, wherease '.xml' files contains the information about boxes enclosing desired objects.

(4) In ‘pva-faster-rcnn/lib/datasets/’ folder, copying ‘pascal_voc.py’ as a different copy and renaming it as ‘project_name.py’. Opening this ‘project_name.py’ and modifying the ‘pascal_voc’ class according to the following guidelines:

Modifications:
(a) Changing class name, imdb name, datapath, classes, assertion messages inside __init__() function of class.
(b) Changing voc path to project devkit path - get_default_path() function.
(c) Changing the print statement in write_voc_results_file() function.
(d) Changing ‘annopath’ and ‘imagesetfile’ in do_python_eval function.
(e) Changing import filenames as the current file name.

(5) Modifying ‘factory.py’ in ‘pva-faster-rcnn/lib/datasets/’ folder. Importing the recently modified ‘project_name.py’ , and also adding a ‘for’ statement extra similar to the existing one for voc dataset.

(6) Modifying ‘test.py’ in ‘pva-faster-rcnn/lib/fast_rcnn’ by replacing ‘bbox_pred’ by ‘bbox_pred_<project_name>’

(7) In ‘pva-faster-rcnn/models/pvanet/example_train’ folder, copying the existing ‘train.prototxt’, ‘test.prototxt’ and ‘solver.prototxt’ as ‘train_<project_name>.prototxt’, ‘test_<project_name>.prototxt’ and ‘solver_<project_name>.prototxt’ and modifying these files as indicated below:

(a) **train_<project_name>.prototxt:**
- (i) Renaming ‘cls_score’ layer to ‘cls_score_<project_name>’, ‘bbox_pred’ to ‘bbox_pred_<project_name>’ in the file.
- (ii) Modifying the ‘num_classes’ to the number of classes in the dataset in ‘input-data’ layer.
- (iii) Modifying the ‘num_classes’ to the number of classes in the dataset in ‘roi-data’ layer.
- (iv) Modifying the ‘num_output’ to number of classes in ‘cls_score_<project_name>’ layer.
- (v) Modifying the ‘num_output’ to 4*<num of classes> in ‘bbox_pred_<project_name>’ .

(b) **test_<project_name>.prototxt:**
- (i) Renaming ‘cls_score’ layer to ‘cls_score_<project_name>’, ‘bbox_pred’ to ‘bbox_pred_<project_name>’ in the file.
- (ii) Modifying the ‘num_classes’ to the number of classes in the dataset in ‘proposal’ layer.
- (iii) Modifying the ‘num_output’ to number of classes in ‘cls_score_<project_name>’ layer.
- (iv) Modifying the ‘num_output’ to 4*<num of classes> in ‘bbox_pred_<project_name>’ .

(c) **solver_<project_name>.prototxt:**
- (i) Changing the path of train_<project_name>.prototxt in the first line.
- (ii) Renaming ‘bbox_pred’ to ‘bbox_pred_<project_name>’
- (iii) Renaming the snap_prefix from ‘pvanet_frcnn_v4’ to ‘pvanet_frcnn_<project_name>’.

(8) Starting the training using the following commands:

```
python tools/train_net.py --gpu 0 --solver
models/pvanet/example_train/solver_<project_name>.prototxt --weights
models/pvanet/pretrained/pva9.1_pretrained_no_fc6.caffemodel --iters 350000 --cfg
models/pvanet/cfgs/train.yml --imdb
```

Note:

(a) Steps number 4, 5, and 6 are used to modify the dataset from voc, coco to the custom dataset provided by the user so as to facilitate the automation of the training process.

(b) Step 7 is used to tweak the Neural Network architecture, which has been developed using Caffe and stored in ‘.prototxt’ files, based on user’s input.

(c) Step 8 is a standard command for starting the training of this model.

Scope of Improvement

(1) This project utilises just a single architecture and automates its training. In future, this project can contain multiple architectures, compare its performance and then finalise an optimal solution for that problem.
(2) A user friendly interface should be developed to make this project commercially viable in future.
(3) Machine Learning problems are not limited to only object detection, it can belong to language processing, medical analysis, industrial automation, etc. This project could be improved to perform on these problems as well.

Conclusion

The whole experience of working at Wipro Digital was great. This company has good work culture and very high quality of work ethics. I learnt a lot about the latest technology used in the dynamic world of Machine Learning and Artificial Intelligence. The work I could complete here was very satisfactory. I have tried to develop as many add-ons in my skill set as possible. I hope my work at this company could help it meet its goals.

References:

(1) https://github.com/sanghoon/pva-faster-rcnn
(2) https://www.coursera.org/learn/machine-learning
(3) http://colah.github.io/
(4) https://machinelearningmastery.com/
(5) http://neuralnetworksanddeeplearning.com/chap2.html
(6) https://github.com/mnielsen/neural-networks-and-deep-learning
(7) https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-object-detection/
(8) http://www.riptutorial.com/caffe/example/31618/layer-template