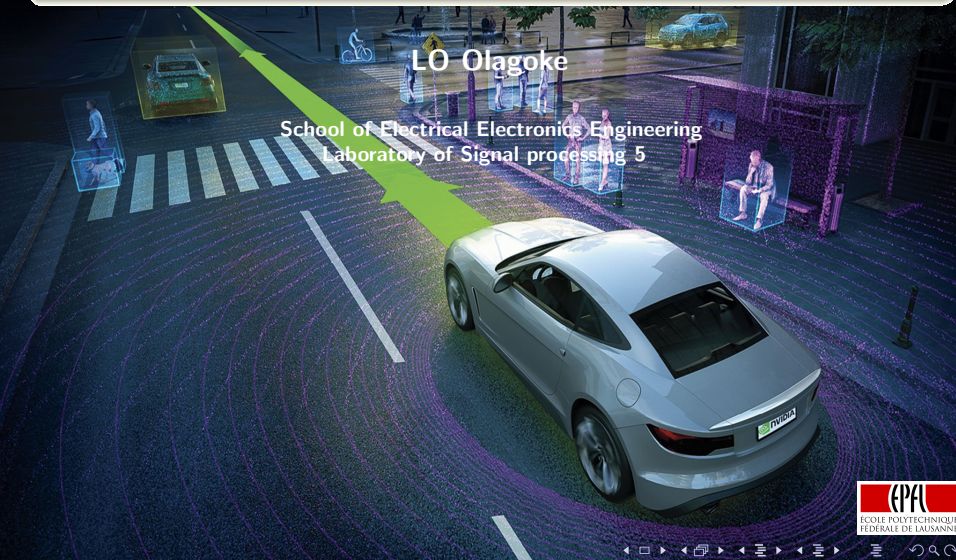


# Object Detection at Multi-Scale Using CNN



# Table of Contents

- 1 Introduction
- 2 Conv Net Architecture
- 3 Fine Tuning Network: The Multi-Scale Net Architecture
- 4 Training Results
- 5 Deductions From The Results
- 6 Conclusion

# Motivation: Autonomous Vehicle Framework

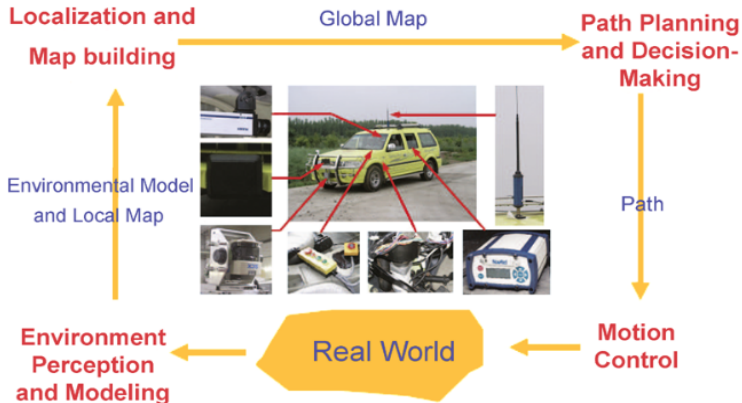


Figure: Autonomous Vehicle Framework<sup>1</sup>

<sup>1</sup> Image Source: Hong Cheng: *Autonomous Intelligent Vehicles - Theory, Algorithms, and Implementation*

# The Goal

- Transfer specific capabilities of Human perception to Autonomous vehicle. **How?**
- (How) To do this we need to gather information about the real world using different sensor types (Examples: LiDAR, Camera, Radar)
- These sensors provide a mapping from a scene to **data**
- The goal is to extract **information** and useful **semantics** of multi-scale object from data.

# Environmental perception and modelling revisited

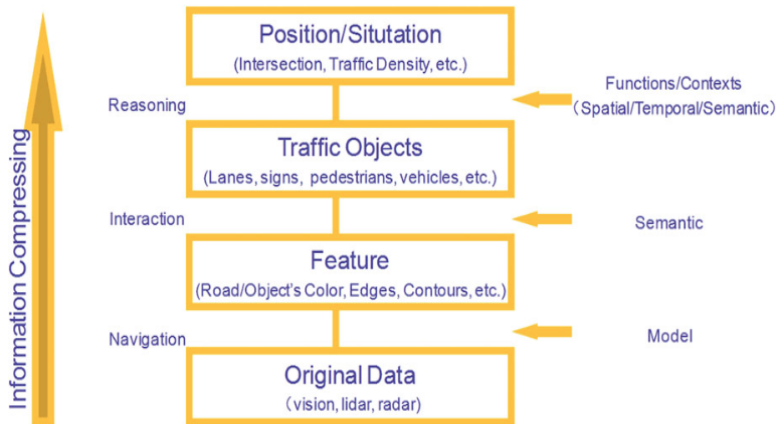


Figure: The Vision model<sup>2</sup>

<sup>1</sup> Image Source: Hong Cheng: *Autonomous Intelligent Vehicles - Theory, Algorithms, and Implementation*

# Sensor Data: The Kitti Data Set

- This project uses the Kitti Data set.  
<http://www.cvlibs.net/datasets/kitti/>
- The Kitti dataset was captured from a VW station wagon moving around Karlsruhe (Germany).
- It consist of 6 hours of diverse real driving scenerio recorded at 10-100Hz using a variety of sensor. The sensor setup is as below:
  - 2 × PointGray Flea2 grayscale cameras (FL2-14S3M-C)
  - 2 × PointGray Flea2 color cameras (FL2-14S3C-C), 1.4 Megapixels.
  - 4 × Edmund Optics lenses.
  - 1 × Velodyne HDL-64E rotating 3D laser scanner, 10 Hz, 64 beams, 0.09 degree angular resolution, 2cm distance accuracy, collecting 1.3 million points/second.

# From Data to Information Extraction (Feature Map)

- This can be achieved through Convolutional Neural Net (CNN or Conv Net for short)
- Feature Maps of object classes at multi scale are extracted from the data using Conv Net
- The model is trained to associate with each feature map the correct object class or label. (semantics).
- Four objects classes/labels were trained : Small pedestrians, Big Pedestrians, Bicycles and cars.

# The Conv Net Components

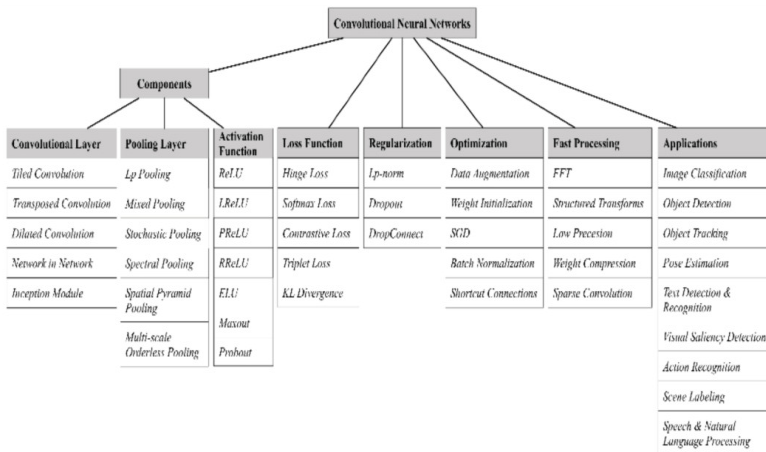


Figure: The Components of Conv Net<sup>3</sup>

<sup>1</sup> Image Source: Jiuxiang et al: *Recent advances in Convolutional Network*



# Base Model : VGG-16<sup>4</sup>

## Salient Features of VGG-16:

- Small size Kernels throughout
- Uses three non-linear rectification layer instead of one to make the decision function more discriminative
- Incorporation of 1 X 1 conv layer to increase the non-linearity of the decision function.
- Large parameter size but takes lesser time to converge (because of pre-initialisation of certain layers and regularisation imposed by greater depth and small filter size).

---

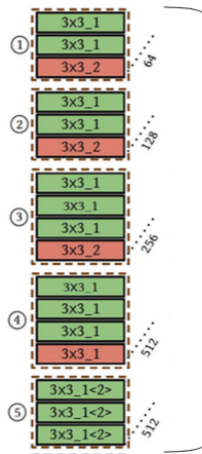
<sup>1</sup> Simonyan, Karen, and Andrew Zisserman et al: *Very deep Convolutional network for large scale image recognition*, arXiv preprint: arXiv: 1409.1556(2014)

# The Multi-Scale Conv Net Architecture

- Inspired from Oxford VGG net (16 layers)
- Small size Kernels
- Weights initialized from VGG-16 net for stable and effective training
- Fine tuning the network to adapt for object recognition at various scales.


Motivation for finetuning: study change in model accuracy for multi scale object as size of network increases. What conv size layer performs best at different object scales?

# The Multi-Scale Net Set-up



Weights initialized from  
VGG-16 net for stable and  
effective learning

 : Convolutional Layer  
followed by ReLU

 : Max pool Layer

# Multi-Scale Net : Fine Tuning Layer

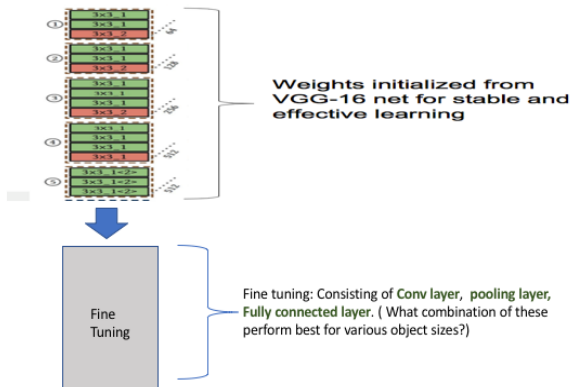


Figure: Finetuning network for Multiscale object detection

# Deep Look at Finetuning Layer

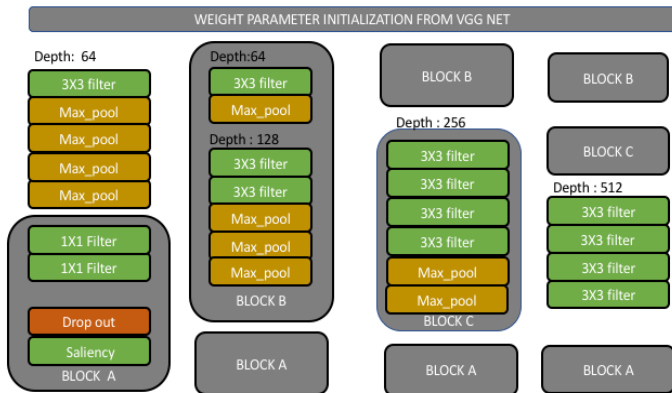


Figure: Finetuning network for Multiscale object detection

# Remarks About Architecture

- Saliency defines another convolution layer but returns returns 5 Feature Maps used together with the input labels for computing the loss . (Small pedestrian, big Pedestrian, Bicycles , cars and no object)
- Each model is run and we try to figure which model performs well for each object class.
- A full network consisting of the Depth:512 block above replicated twice was also tested.
- A plot of error was made for each model. The plot was smoothened using savitzky golay filter using a window size of 51 and polynomial of degree 3. <sup>5</sup>

---

<sup>5</sup><https://en.wikipedia.org/w/index.php?title=Savitzky>

# Results Obtained For Small Pedestrians

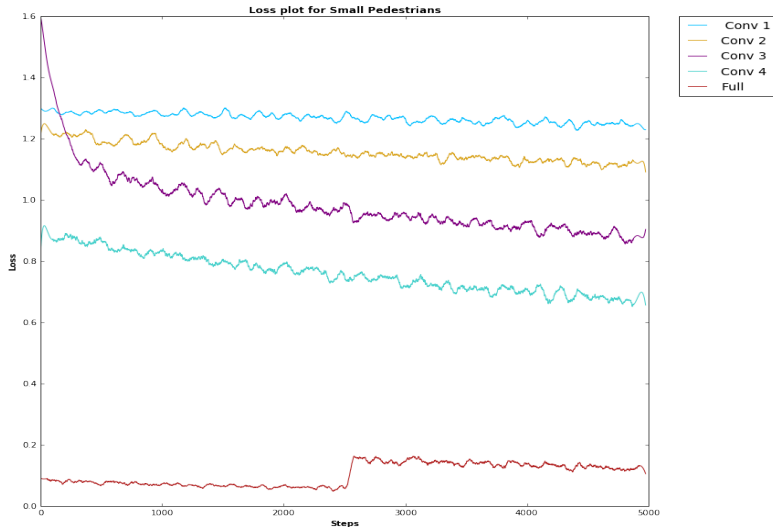


Figure: Loss plot Small pedestrian

# Results Obtained Big Pedestrians

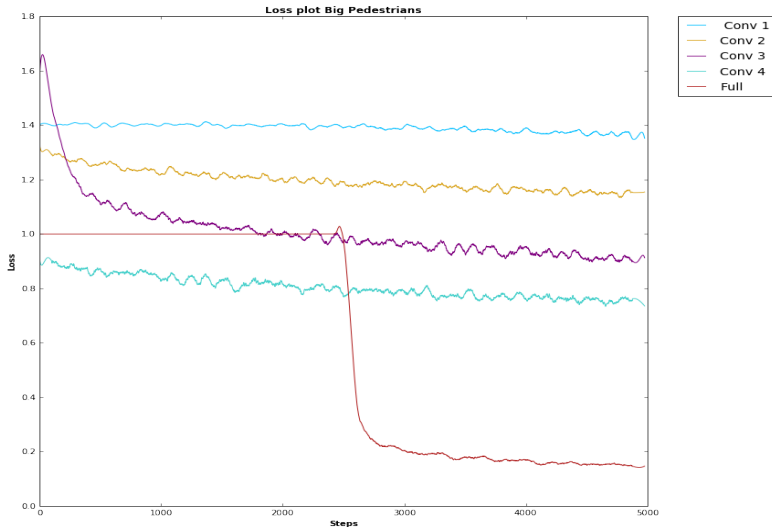


Figure: Loss plot for Big pedestrian



# Results Obtained Cars

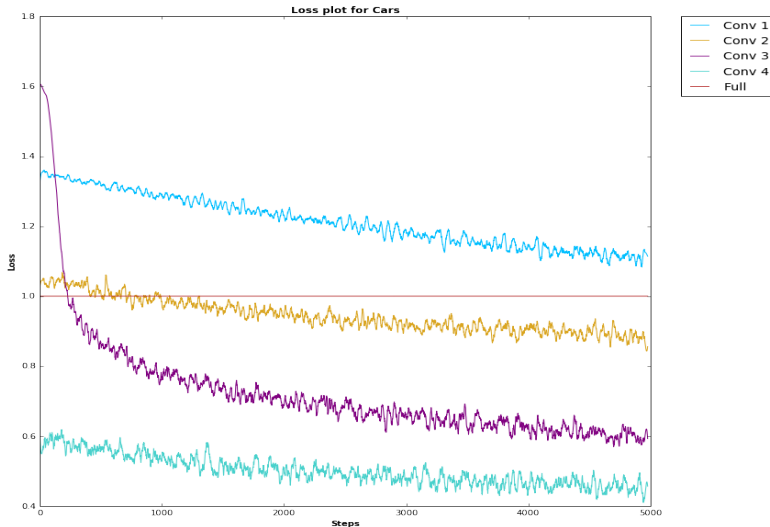


Figure: Loss plot for Cars

# Results Obtained Cyclist

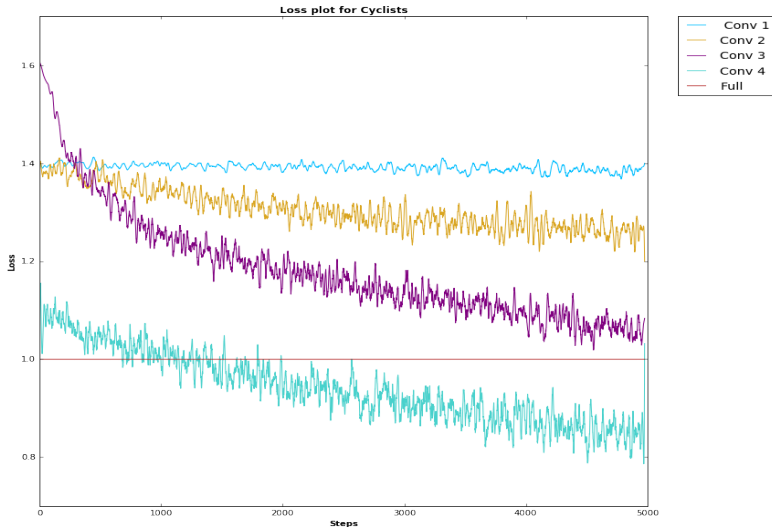


Figure: Loss plot Cyclist

# Deductions From The Results

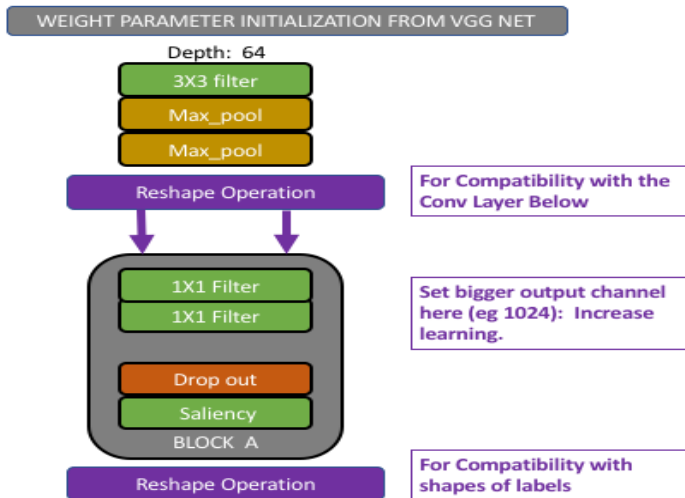
- The Full network is not always the best for all class labels
- The full network performs well on at least small and big pedestrians
- For Cars Conv 4 performs absolutely better
- The models generally don't perform well in identifying cyclist

The need for ensemble model: Since each model performs different on each object class at various scales, an ensemble model that uses the best model for each class or size category category might be more robust

# Can we do better?

Motivation: In search of a simple 1 conv model, that performs well. This would be easy to train than the full model

# Can we do better?



# Results from new Net Model

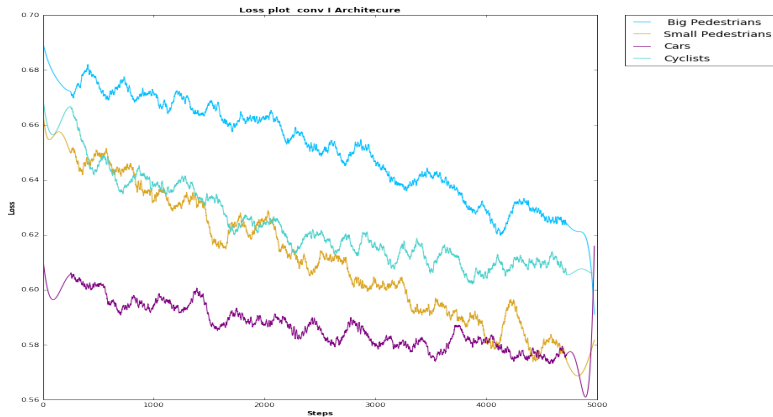


Figure: Simple One Conv Layer results: Loss plot

# Results from new Net Model

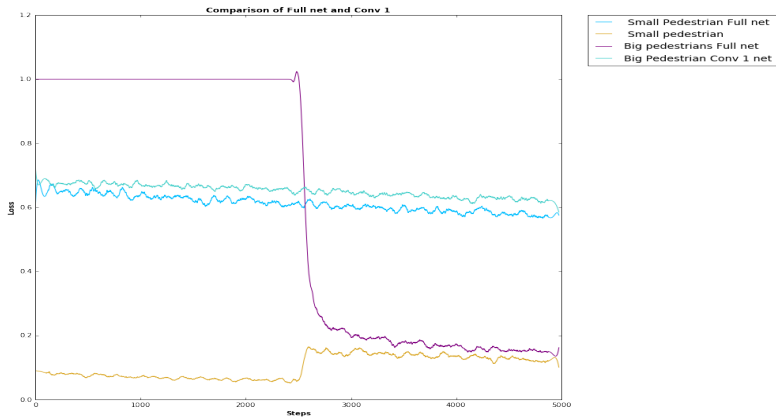


Figure: Comparison with Full net: On class labels small and big pedestrians

# Results from new Net Model

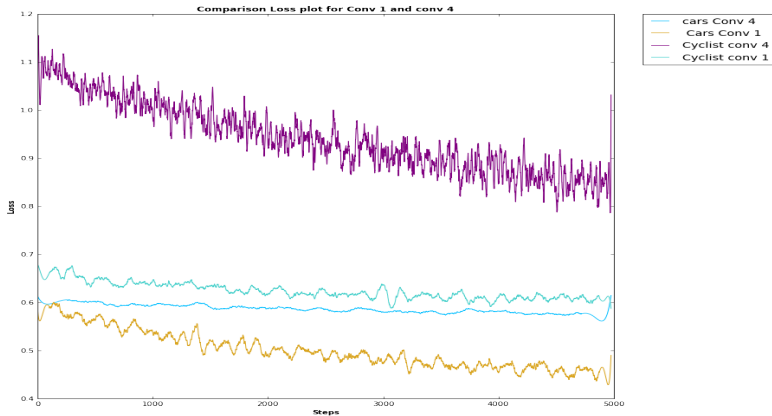


Figure: Comparison with Conv 4 net: On class labels Cars and Cyclists



# Conclusion

- It is a work in progress!
- No discussion of bounding box for the image labels yet. The aim is to build a good architecture and use that as a base model for subsequent improvements.
- Research still on on how to better compute accuracy for each class labels
- Thank You for Your attention.
- Questions ?
- Suggestions ?