Computer Vision Project

Low Light Imagery

Team - Wanda Vision

TA Mentor – Pulkit Gera

Team Members:
1. Dipanwita Guhathakurta - 2018112004
2. Manav Bhatia - 2018102009
3. Sanskar Tibrewal - 2018111034
4. Vedant Mundheda - 2018112006

Inspiration

Motivation

- As the world is progressing and we have seen companies talking about the self-driving cars. One of the important thing that every self-driving car should do is to recognize traffic signs to drive safely. During daytime models can detect traffic signs, but it gets a lot trickier when its low light out there. Hence our model will help to detect traffic signals in low light.
- Our model will convert low-light images into brighter images and recognize the traffic signs from them.
- Physical changes such as extending exposure time and using flash which can introduce blur and requires expensive hardware. This may be dangerous for a self-driving car.

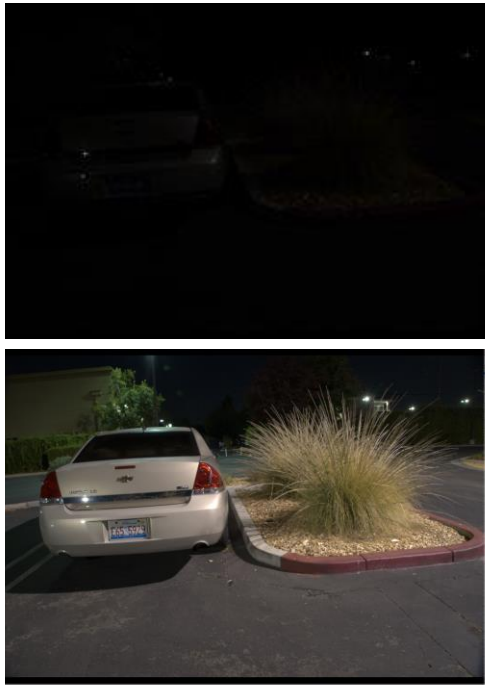
Objective:

From a short exposure, low-light sensor image, get a bright image using the Seeing in the Dark network and recognize traffic signs using custom Convolutional Neural Network.

Low Light Imaging

We derive inspiration from the CVPR 2018 paper "**Learning to See in the Dark**" which proposes a pipeline for processing low-light images, based on end-to-end training of a fully-convolutional network. The pipeline proposed can process raw sensor data as opposed to physical changes such as extending exposure time and using flash which can introduce blur and requires expensive hardware.

A) Raw sensor data (ISO 8000)
B) Image produced by camera using greater exposure (ISO 409,600)
C) Image produced from the network proposed



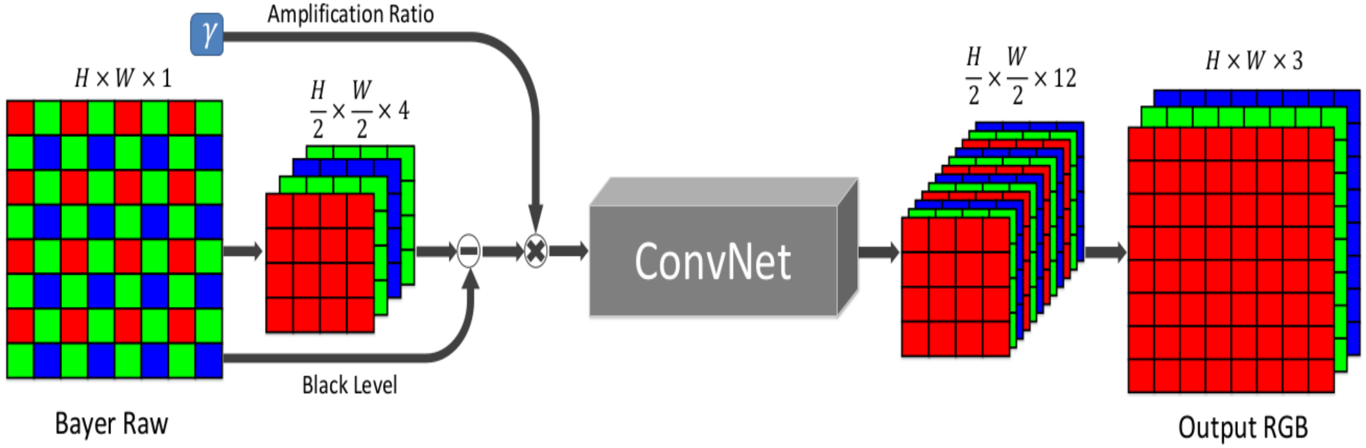
Top: Image from the Sony α7S II sensor with short exposure
Bottom: Output from the model as proposed in the paper

Proposed pipeline for Low Light Imaging

PIPELINE PROPOSED TO BE USED

- We propose to use end-to-end deep learning pipeline for direct single-image processing of low-light images, as mentioned in the Research paper.
- We intend to use a Fully-Convolutional Network(FCN) for the end-to-end image processing pipeline as the paper theorized that pure FCN's can effectively represent many Image Processing Algorithms.
- Rather than operating on fully sRGB images, this algorithm intends to work on raw sensor data.
- Also, we intend to use a parameter known as amplification ratio which will determine the amount of brightness that will be present in our output.
- The Convolutional Architecture that is intended to be used is U-Net as it has less memory consumption and doesn't use fully connected layers to work on small image patches.

THE PROPOSED IMAGE PROCESSING PIPELINE TO BE USED



PRE-PROCESSING AND POST-PROCESSING

- **Pre-Processing** –
- For Pre-Processing, we intend to use the Bayer Raw data and pack it into 4 channels and reduce the spatial Resolution along both dimensions by a factor of 2.
- If we get X-trans arrays – we arrange the data into 6 x 6 blocks and pack it into 9 channels by exchanging adjacent elements.
- Then we subtract the black levels from the image and amplify(multiply by the amplification ratio) the input image.
- Now, in the main phase of the operations, we feed the packed and amplified data into a fully-convolutional Neural Network(FCN).
- **Post-Processing**-
- The output is designed such that we get a 12-channel output image with half the spatial resolution post the ConvNet stage.
- The output is then passed through a sub-Pixel layer to resize and recover the original image.

TRAINING THE NETWORK

- The above-proposed networks will be trained from scratch using L1 loss and ADAM optimizer.
- During Training, the input is the raw data of the low-light short-exposed image and the ground truth is the corresponding long-exposure sRGB image(processed by a raw-image processing library).
- In each iteration, we crop a random 512 x 512 patch and apply random flipping and rotation for data augmentation.
- Also, we initially set the learning rate to 10^{-4} , and after 2000 epochs, reduce it to 10^{-5}.
- The training then proceeds for 4000 epochs.

DATASET GENERATION

- The dataset we are going to use for this is a set of 20,000 **images from the Swedish traffic signs dataset of which 20% contain traffic signs.** (sequences from over 350 km of Swedish highways and city roads). The dataset available is already annotated with labels of the type of traffic sign and the location of the traffic sign in the image.
- We will adjust the brightness and exposure of the images to synthetically generate low-brightness low-exposure images.
- Finally, when we feed dim image into our network, we will get a bright image.

Dataset link:
http://www.cvl.isy.liu.se/research/datasets/traffic-signs-dataset/

Fredrik Larsson and Michael Felsberg ***, Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition ,*** In Proceedings of the 17th Scandinavian Conference on Image Analysis, SCIA 2011, LNCS 6688, pp. 238-249. bib , doi:10.1007/978-3-642-21227-7_23

Traffic sign recognition

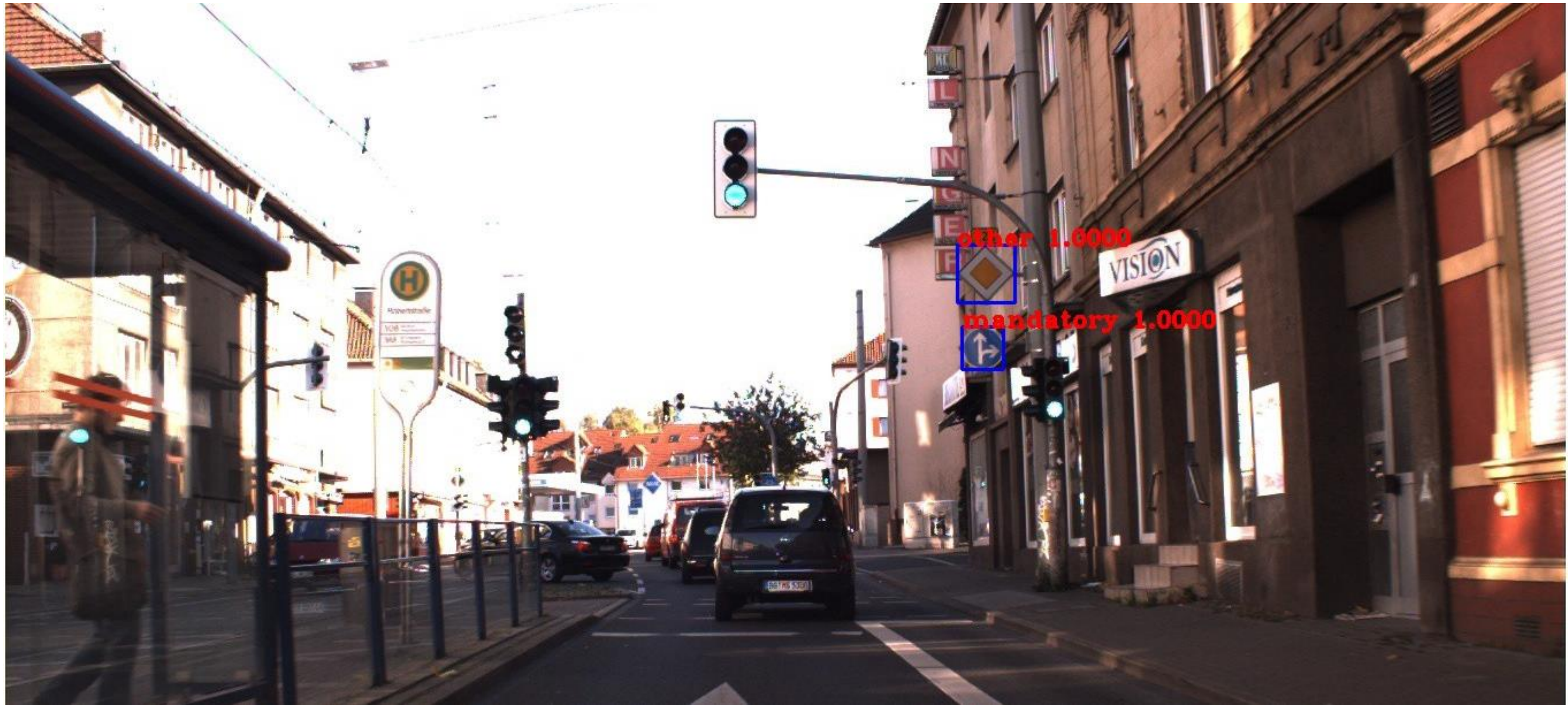
YOLOv3 to detect and recognize traffic signs

- Input: 20000 images of 3488 classes of traffic signs
- Output: Classify each image into one of the 3488 classes.

Model architecture and params:

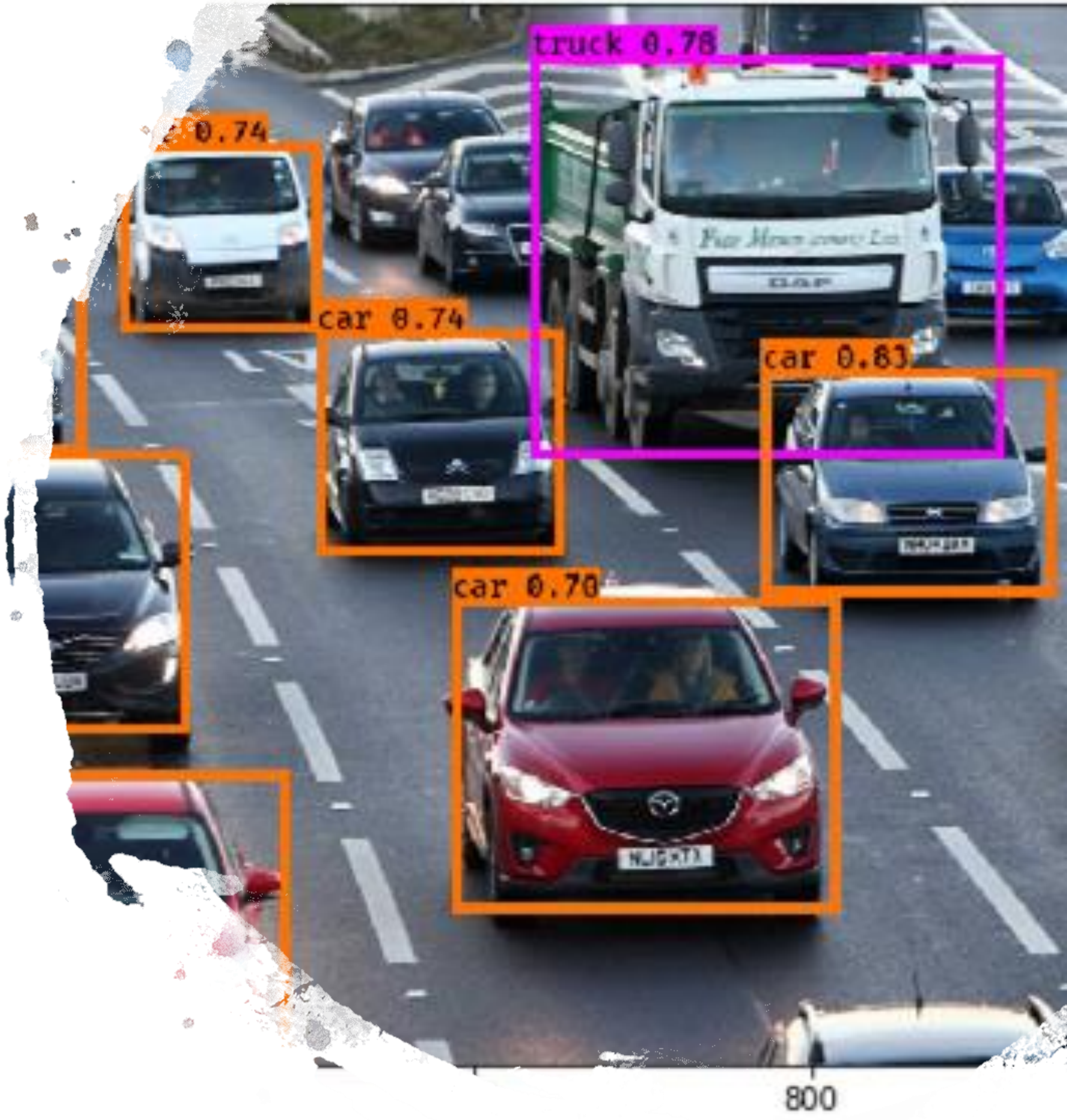
- Feature-extractor: `Darknet53`
- Optimizer: `Adam`
- Loss:

`YOLO loss =` **Localization loss + Confidence loss+ Classification loss**



YOLO FRAMEWORK TO DETERMINE DIFFERENT TRAFFIC SIGNS

- In the final step, we intend to use an Object detection (YOLO) framework and train it such that we can determine whether there is a traffic sign or not in that Image.
- If the object is present, we intend to train it such that we can determine the difference between 3488 classes of traffic signs, as mentioned in the dataset.
- Also, we intend to create Bounding boxes around the signs in the images to actually recognize the signs and the class they belong to.



MID-EVALUATION DELIVERABLE

The mid-evaluation Deliverable that we intend to present is -

- We will be able to prepare our dataset and have it ready with both the short-exposure, low-light images and the long-exposure, sRGB Images.
- We will implement the Image processing pipeline of the "**See in the Dark**" Paper, along with a pre-trained model and doing the training part of the network on our custom dataset.

THANK YOU