# Computer Vision Project

## Low Light Imagery

Team - Wanda Vision

TA Mentor – Pulkit Gera

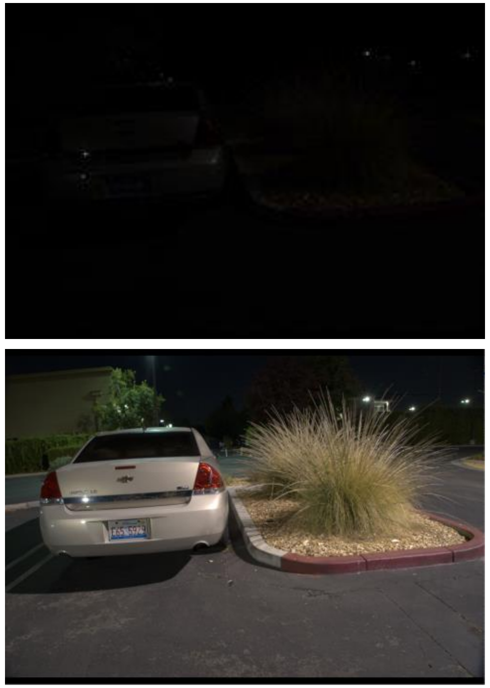

**Team Members:**
1. Dipanwita Guhathakurta - 2018112004
2. Manav Bhatia - 2018102009
3. Sanskar Tibrewal - 2018111034
4. Vedant Mundheda - 2018112006

# Inspiration

# 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.

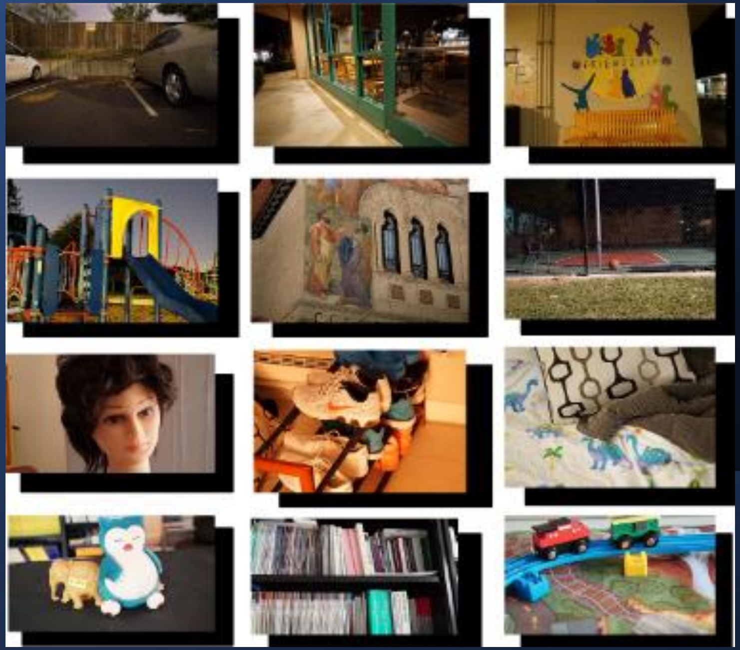
A) Raw sensor data (ISO 8000)
B) Image produced by camera using greater exposure (ISO 409,600)
C) Image produced from the network proposed (example from paper)



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

# Implementation

# DATASET: SID SONY



## See-in-the-Dark dataset (SID)

**Camera:** Sony α7S II

**Dataset Size:** 5094 pairs of short-exposure and long-exposure images

**Image shape:** 4240×2832

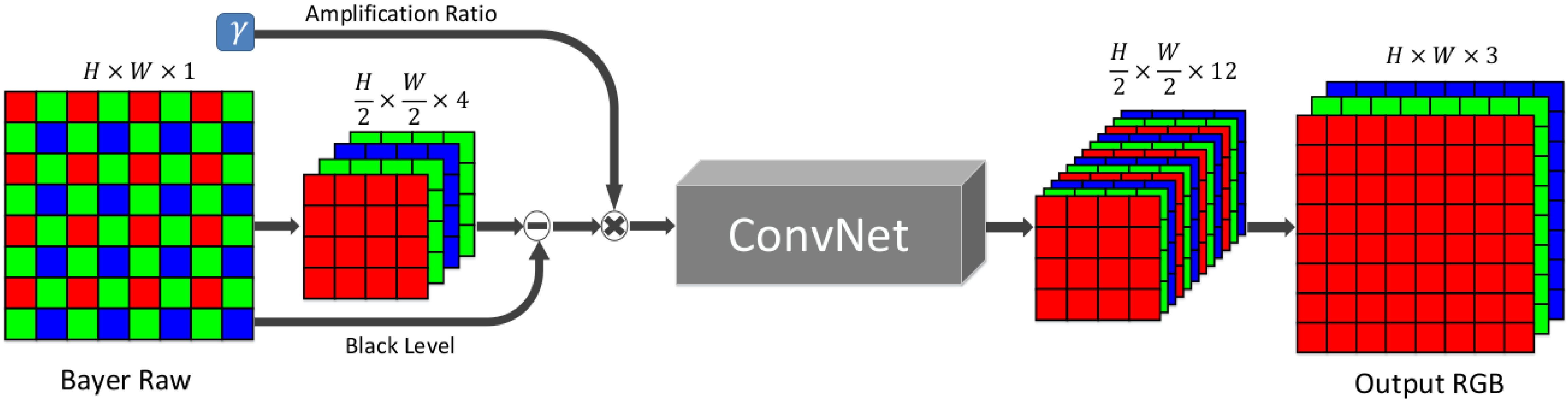
**Exposure for Input Images: 0.03 to 0.1 sec**

**Exposure for Ground Truth Images: 10 to 30 sec**

| Ground Truth (long exposure) | Input Image (short exposure) |
|---|---|
|  |  |

**Dataset link:**
https://drive.google.com/file/d/1G6VruemZtpOyHjOC5N8Ww3ftVXOydSXx/view

# IMAGE PROCESSING PIPELINE



# CHOICE OF ARCHITECTURE & LOSS FUNCTION

## Why FCN?

End-to-end image processing pipeline that runs fast even on large raw images, does not require a fixed size input output like Fully Connected Layers

## Why UNet?

- less memory consumption
- doesn't use fully connected layers to work on small image patches

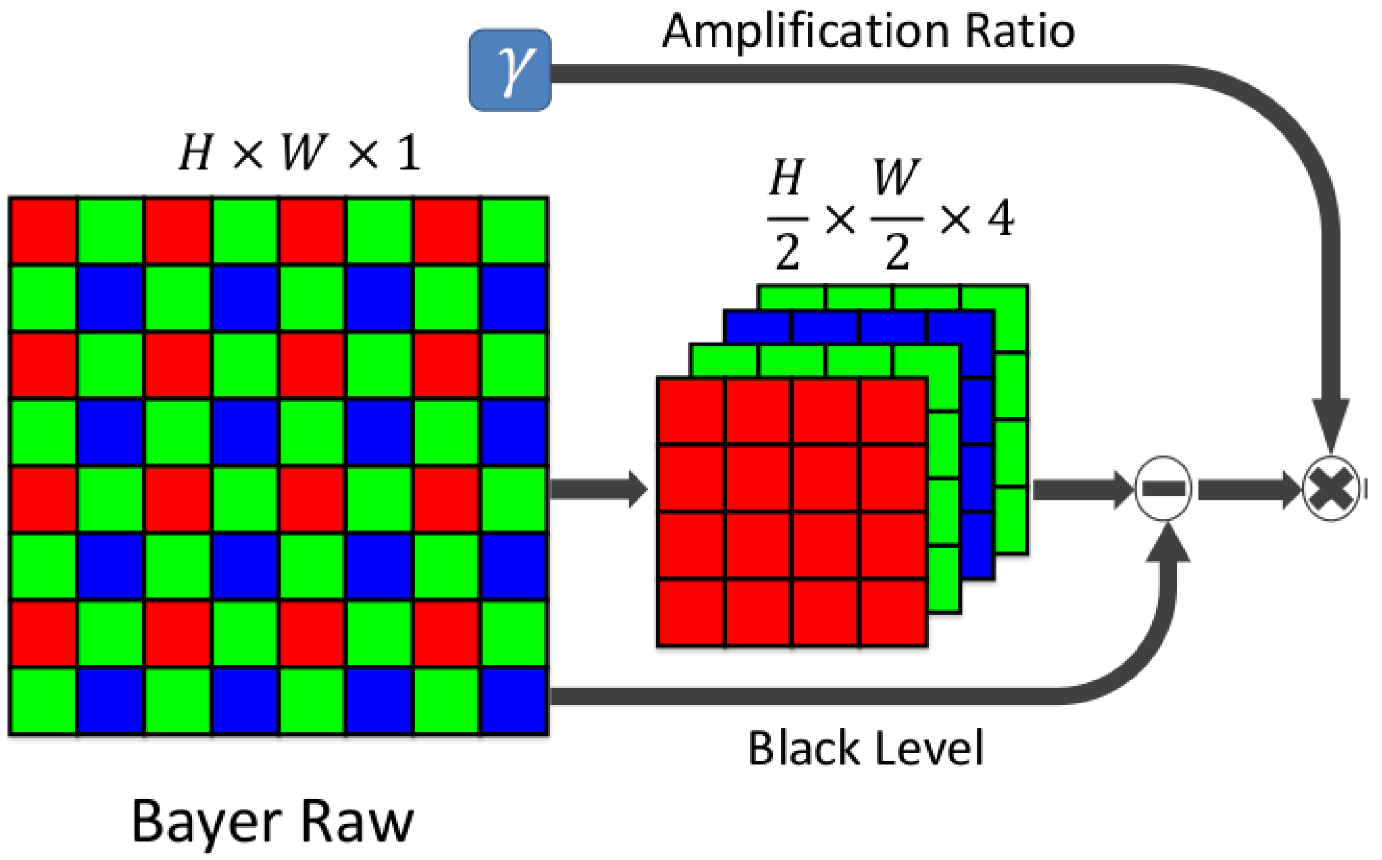
## Why L1-loss?

| | |
|---|---|
| **Our default pipeline** | 28.88/0.787 |
| $L_1 \rightarrow$ SSIM loss | 28.64/0.817 |
| $L_1 \rightarrow L_2$ loss | 28.47/0.784 |

From paper: PSNR/SSIM values on replacing L1 loss

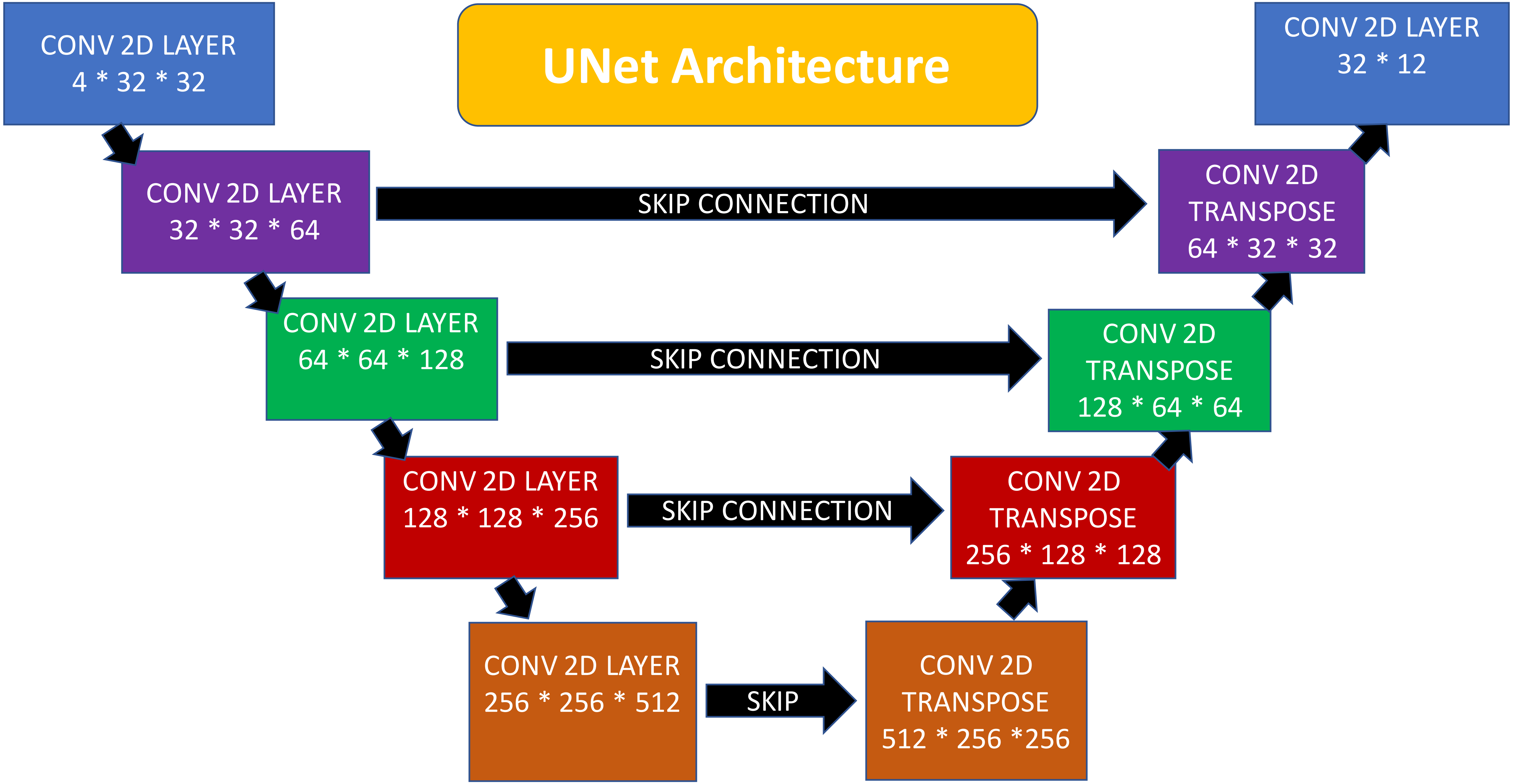
# PRE-PROCESSING

| Libraries Used |
| --- |
| RawPy |
| NumPy |



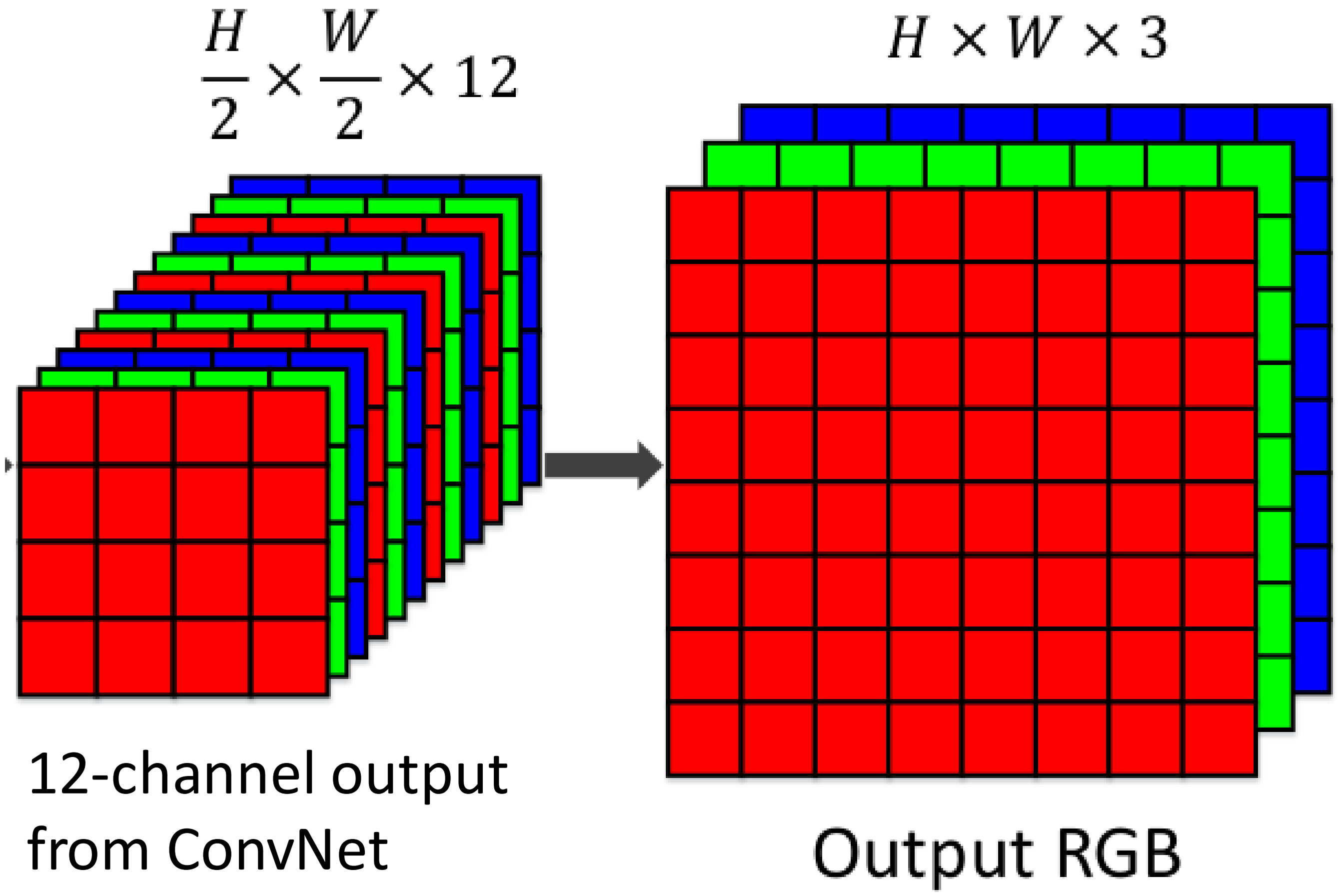

Halve dimensions → Pack Bayer Raw data into 4 channels → Subtract black levels → Amplify by amplification ratio

**Amplification Ratio**
= Exposure difference between input and reference images


UNet Architecture
CONV 2D LAYER
4 * 32 * 32
CONV 2D LAYER
32 * 32 * 64
SKIP CONNECTION
CONV 2D
TRANSPOSE
64 * 32 * 32
CONV 2D LAYER
32 * 12
CONV 2D LAYER
64 * 64 * 128
SKIP CONNECTION
CONV 2D
TRANSPOSE
128 * 64 * 64
CONV 2D LAYER
128 * 128 * 256
SKIP CONNECTION
CONV 2D
TRANSPOSE
256 * 128 * 128
CONV 2D LAYER
256 * 256 * 512
SKIP
CONV 2D
TRANSPOSE
512 * 256 *256

# POST-PROCESSING

| Libraries Used |
| --- |
| RawPy |
| PyTorch |
| NumPy |




Pass (C x r2)*(H/2)*(W/2) output through sub-Pixel layer

Obtain C*((H/2)*r)*((W/2)*r) image

# TRAINING THE NETWORK

**Input:** short exposure images

**Loss Function Used:** L1 loss

**Optimizer:** ADAM

**Batch:** 512*512 window from original image

**Data Augmentation:** Random flipping and rotation

**Learning Rate :** $10^{-4}$ after 2000 epochs, $10^{-5}$

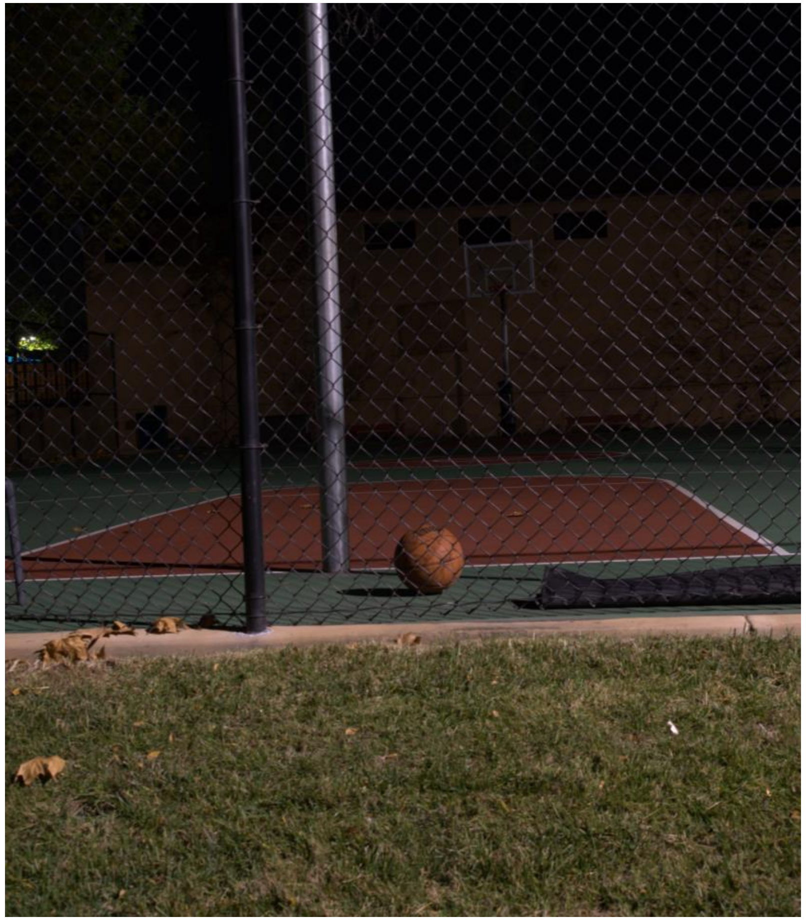
**Iterations:** 4000 epochs

**Ground truth:** long-exposure images

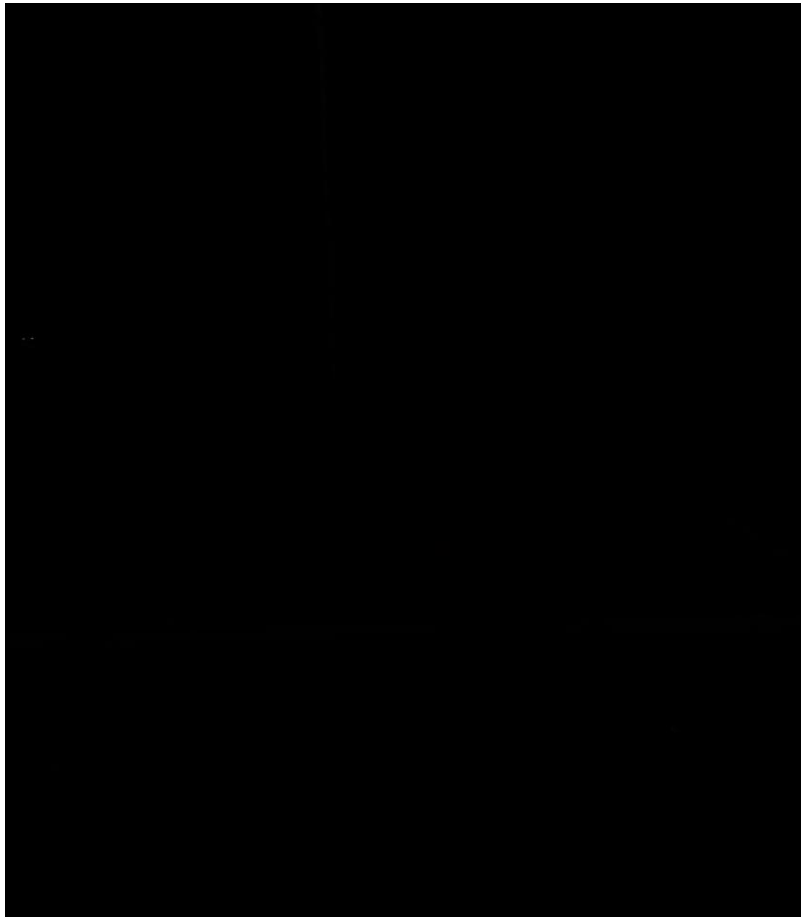
# RESULTS OBTAINED

**Ground Truth** | **Input Image** | **Output Image**

## Test Image 1: Exposure = 100



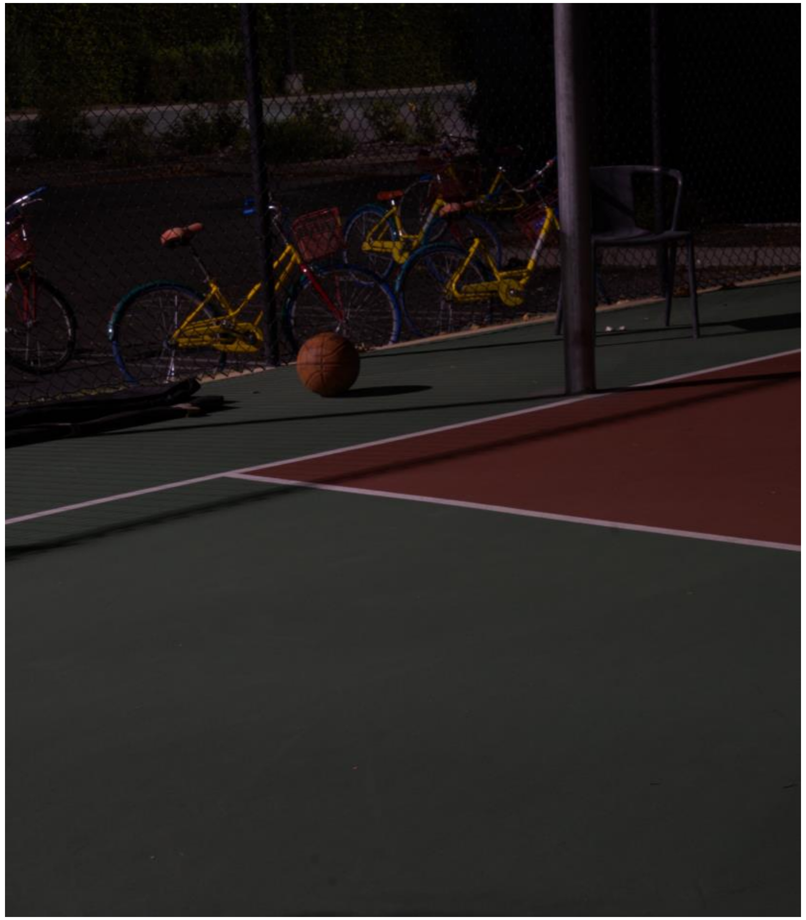
Ground Truth

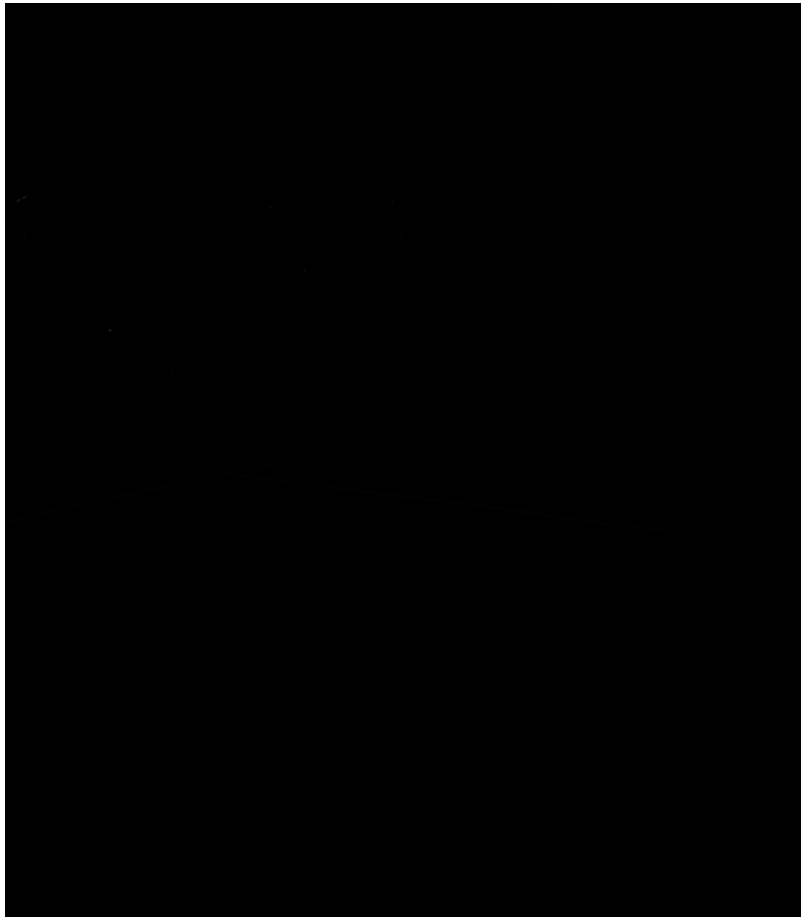


Input Image

Output Image

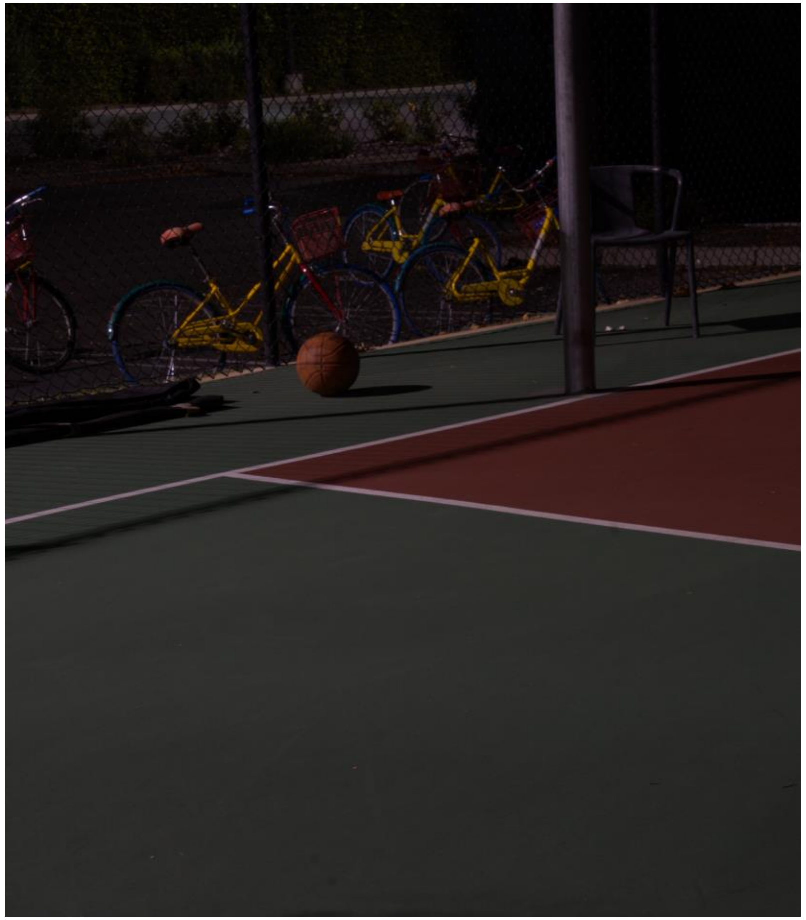
Test Image 1: Exposure = 250



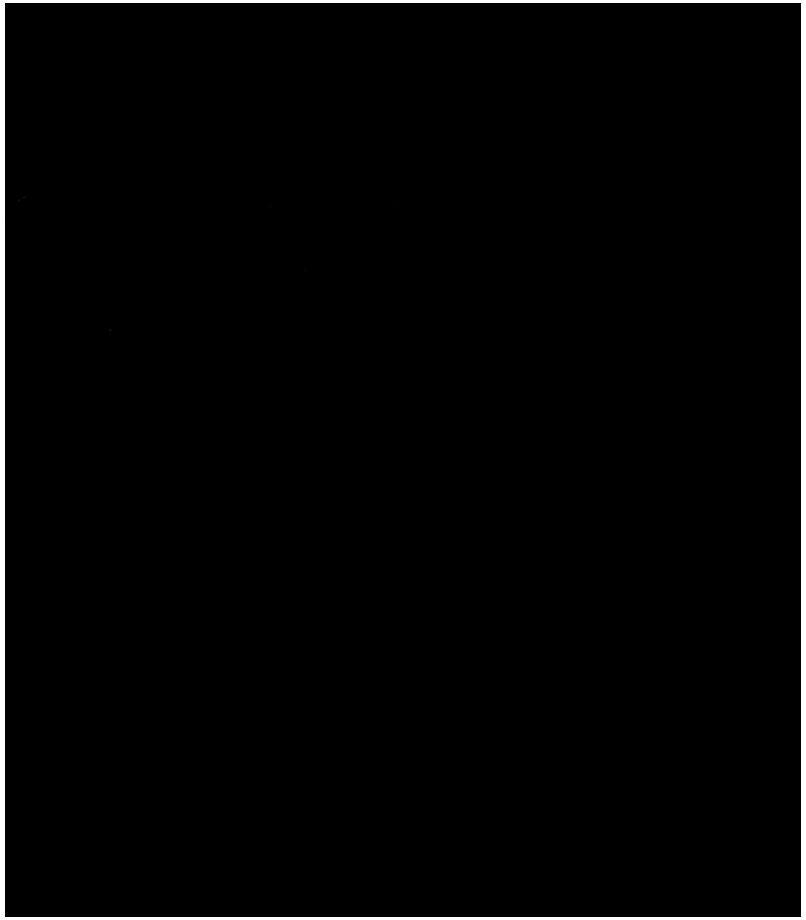


Ground Truth | Input Image | Output Image

**Test Image 2: Exposure = 100**



Ground Truth



Input Image

Output Image

## Test Image 2: Exposure = 250

# THANK YOU