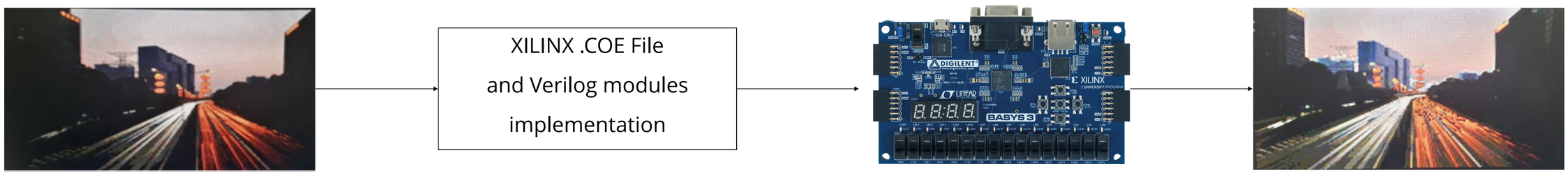
ES203: DIGITAL SYSTEMS

LOW LIGHT IMAGE ENHANCEMENT

This project deals with the implementation of low-light image enhancement on an FPGA and removal of noise (fog, haze, smoke). Low-light image enhancement is one of the major preprocessing tasks in many computer vision applications. Images which are captured in irregular brightness conditions and low-light have less dynamic range and high noise levels. These reduce the performance of algorithms used for computer vision. Hence, we have worked on enhancing the underlying features in an image.

Methodology:

Images cannot be directly loaded on an FPGA. Basys 3 is used as FPGA. Therefore, the input image is converted to Xilinx coefficient (.coe) file. In the coe file, each pixel of the image is stored as a binary value. This file is loaded in FPGA through the use of BRAMs. BRAMs serve as memory component. The synthesized Verilog modules are implemented on FPGA device where all the processing takes place. The final output image is displayed on a monitor through the use of VGA interface.



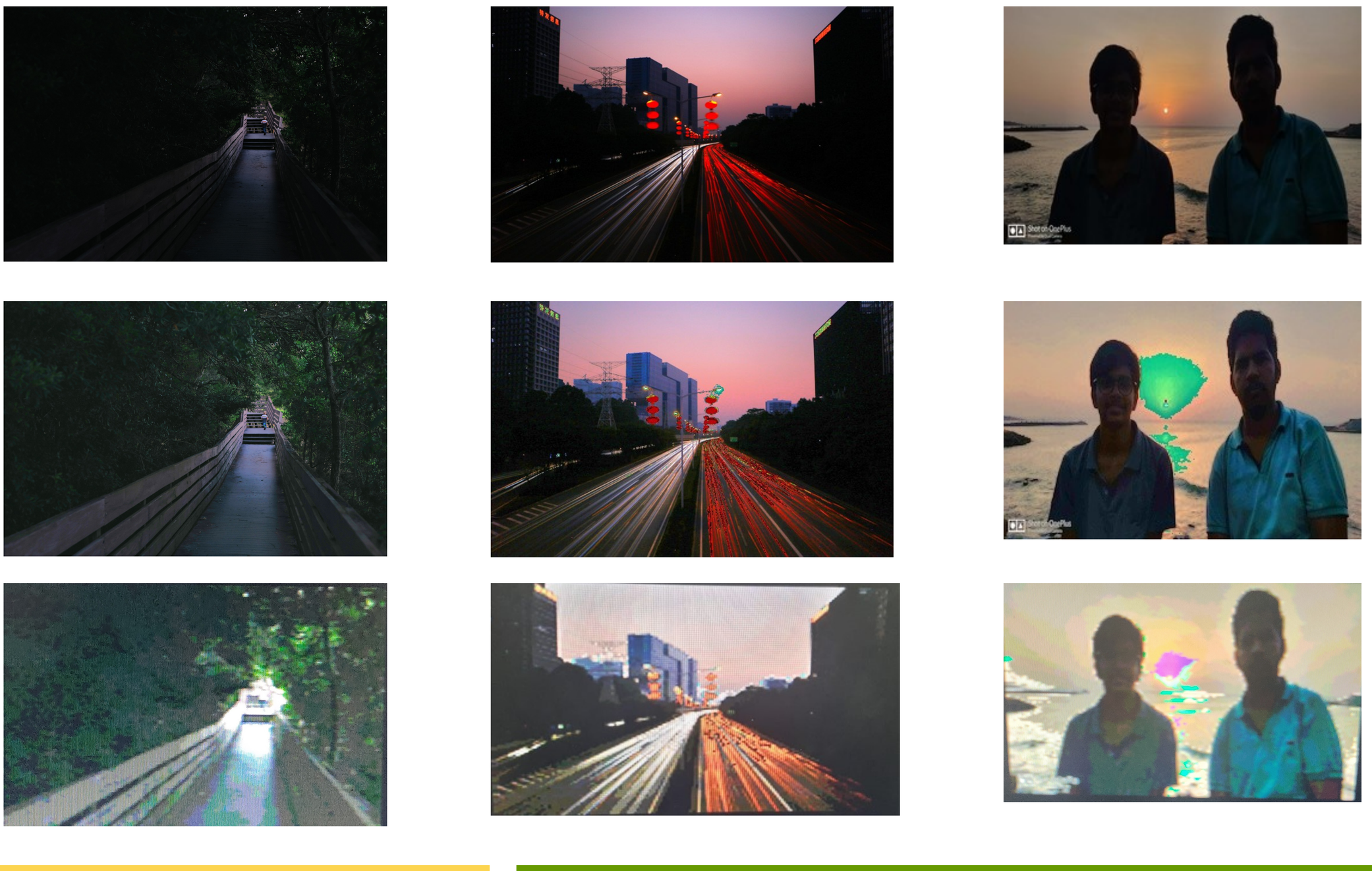
Input Image — FPGA Processing — Output Image

Block diagram of algorithm:

Input image → Inversion → Dehazing using dark channel prior algorithm → Inversion → Output image

Initially, the input image is inverted. After inversion, the pixels in the dark region have low intensities in at least one color channel out of the three color channels. This image is similar to hazy images. The low intensity of the pixels in the dark region is mainly due to scattering or airlight. For improving the dark channel in the inverted image, the algorithm enhances the airlight image. It is done using the dark channel estimation and then refined with a smoothing filter. But, over-enhancement increases noise in the image. To avoid it, a correction is applied before restoration.

Apart from enhancement of low light images, we also implemented median and mean filters to remove noise like salt-pepper noise and Gaussian noise. To remove haze from foggy or hazy images, dehazing algorithm has been implemented.



Synthesis Report

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	500	0	20800	2.40
LUT as Logic	500	0	20800	2.40
LUT as Memory	0	0	9600	0.00
Slice Registers	201	0	41600	0.48
Register as Flip Flop	201	0	41600	0.48
Register as Latch	0	0	41600	0.00
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

Comparison between FPGA (hardware) and software implementations:

FPGA implementation is relatively faster than the software implementation of the same algorithm especially for images of high size. But, various operations like divisions and exponentials becomes complicated to implement on FPGA. In our case, we have simplified those operations such that it's easily implementable on hardware.

References:

1.https://ww2.mathworks.cn/help/visionhdl/examples/low-light-enhancement.html#:~:targetText=This%20example%20shows%20how%20to,capture%2C%20and%20general%20visual%20enhancement.&targetText=This%20algorithm%20improves%20the%20visibility%20of%20the%20underlying%20features%20in%20an%20image.

2.Kang, H. J., Kim, Y. H., & Lee, Y. H. (2015). FPGA implementation for enhancing image using pixel-based median channel prior. International Journal of Multimedia and Ubiquitous Engineering, 10(9), 147-154.

Prudvi 18110057| Kalyan 18110079| Permender Kumar 18110117 | Neeraj 18110138