Project Title: Neural Network Based Correction of Fixed Pattern Noise in Image Sensors

Category: Computer Vision

Team Members: Paul Lim (paulglim@stanford.edu), Steve Mims (mimsy@stanford.edu)

I. Introduction and Statement of the Problem

We work in a company (Fairchild Imaging) that designs and fabricates image sensors. To increase readout speeds, it is often necessary to read more than one row of pixels at a time. So, in an image, pixels adjacent both the vertical and horizontal direction use different amplifiers, which have different offsets, gain, and linearity. Raw images have unwanted fixed-pattern-noise (FPN) as a result of these differences. Customers require that this FPN be as low as 1/20th of the random temporal noise of the pixels.

One of our sensor products produces frames at 240fps by reading out 8 rows at a time (4 at a 30x gain, 4 at a 1x gain). In addition to differing gains/offsets & linearity for each row-type (these differences are unique for each column horizontally), there is a small (fixed over time) gain change in the rows that appears in the vertically direction. All these relationships change with temperature. A manifestation of the effects of FPN is shown in Figures 1a, 1b, 1c, and compared against the corrected images.

We'd like to look at minimizing (correcting) this FPN over a small subset of the pixels in a frame. A perfect correction in a flat field would be a frame where all pixels have the same value. The biggest challenge is to find an FPN correction algorithm with the requirement that it is calibrated for each sensor within a temperature range acceptable to the customer. Figures 2a and 2b show the additional non-uniformity and nonlinearity issues that make a simple correction algorithm not possible. This project aims to find a universal "algorithm" utilizing a trained neural network for this product that works reasonably well for each and every sensor.

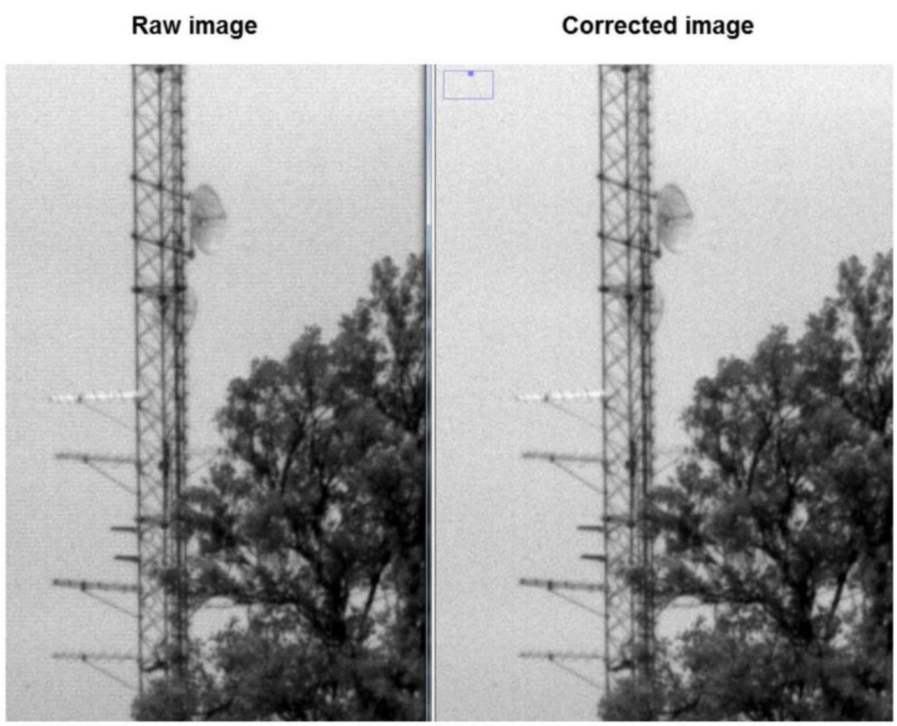

Figure 1a: Real life comparisons; zoom 5x to inspect.

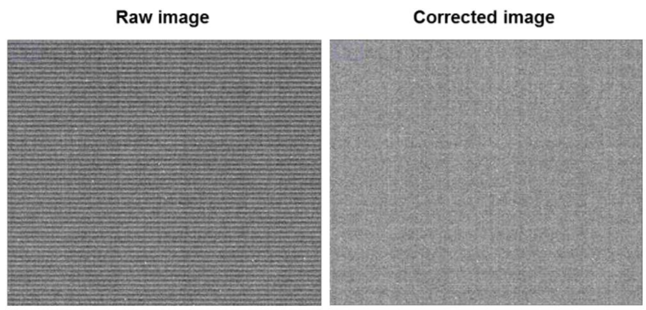

Figure 1b: Flat field illumination comparisons.

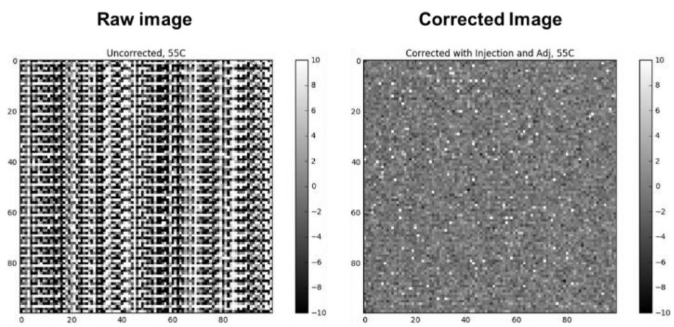

Figure 1c: Dark condition comparisons.

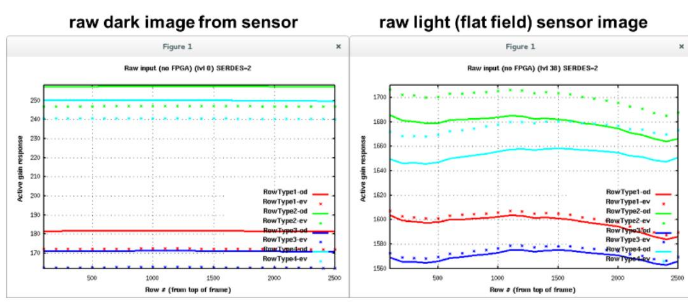

Figure 2a: Nonuniform response behavior, as shown by plots of 25-row averages by row-type and column-type (even vs odd), sweeping down vertically from top of frame.

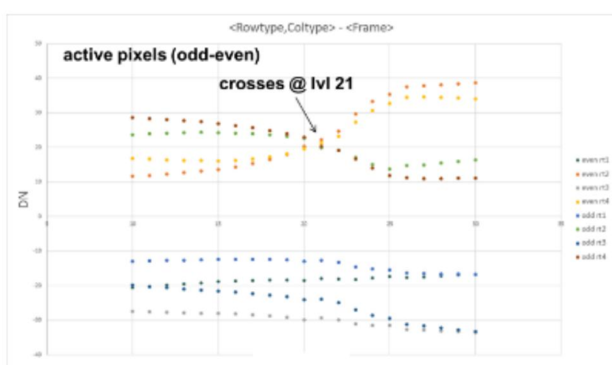

Figure 2b: Difficult section of light response, where odd and even column difference switches polarity.

II. Related Work

We used Reference [1]: He et al, “Single-image-based nonuniformity correction of uncooled long-wave infrared detectors: a deep learning approach” as a starting point for our ConvNet architecture for correction.

III. Data Generation and Augmentation

As it is not practical to capture hundreds of thousands of images on our own, we have decided to collect a few hundred images from the internet and convert them to grayscale. To augment the number of images, we rotated these images at various angles and also flipped them to generate more images. 64x64 slices of the images were then extracted from these images, which serve as the “clean” images of the dataset (see Figure 3). The “noisy” images of the dataset were generated from each of the “clean” images by adding the FPN to them (see Figure 4) based on the reverse of the model that we used to clean the FPN-contaminated images captured by our own sensors. The Matlab/Octave code used to perform this can be seen in the code repository: noise_addFPN.m

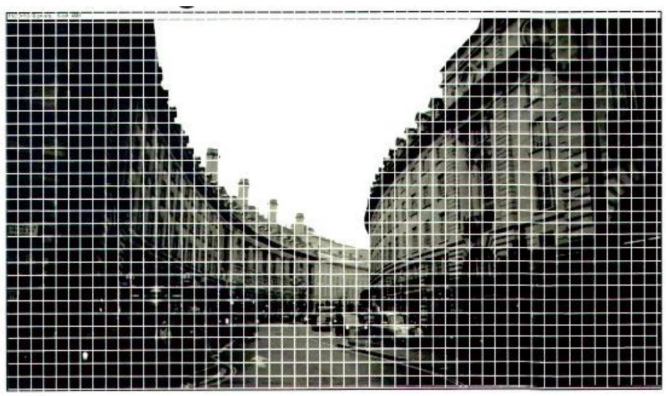

Figure 3: 100 8-bit HD grayscale images were downloaded, sliced up into 64x64 sub-images, then augmented by rotation and flipping, to create 100,000 “clean” sub-images. Then, Matlab code was run to add random fixed-pattern-noise (horizontal and vertical) to these clean images to create 100,000 noisy images. The noise was based on approximately 40,000 parameters (e.g. gain/offset), collected on a Fairchild Imaging 4K image sensor.

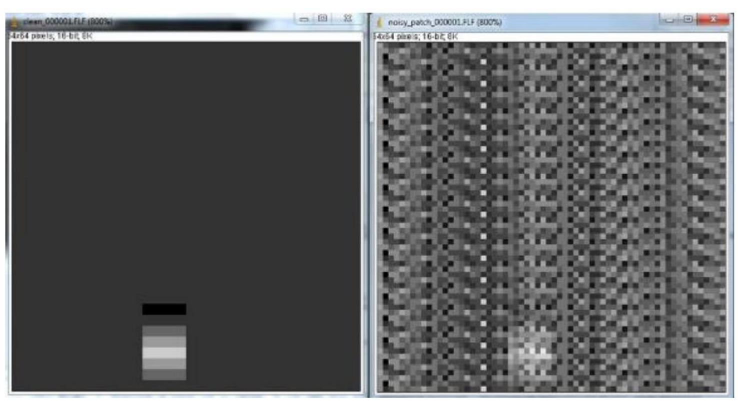

Figure 4: Original clean 64x64 image (left) and image with FPN added (right).

We used a total of 100,000 images for the image correction part of the project: 90,000 for the training set and 10,000 for test set.

IV. Methods

We decided to split the work and try two different approaches, both implemented using the Keras platform.

A. Image Classification on the Presence of FPN:

This part is to see if we could train a ConvNet to train on the dataset and do classification on a test set as way to detect the presence of the pattern of FPN of interest in the images. We used an architecture with two CONV → RELU → MAXPOOL stacks followed by a final sigmoid layer (see Figure 5). After training with 360 64x64 images, the accuracy on a test set of 40 images with equal split of clean and FPN-infused images is 95%. Even though this work appears promising, we decided to focus our efforts on the second approach.

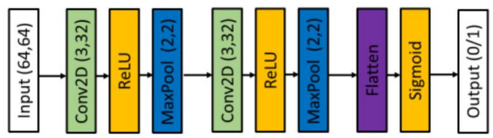

Figure 5: ConvNet architecture used for FPN detection in the images.

B. Image FPN Correction

The second part is to replicate the approach used in aforementioned Reference [1]. Here the researchers utilized a convolutional neural network to reduce FPN while preserving details, the performance of which was deemed to be better than conventional image processing methods (See Figure 6). Both the clean and noisy images are fed into the network, where the noisy image is used to extract the FPN which is then subtracted from it to produce the sanitized output. The sanitized output is then compared to the original clean image as the target.

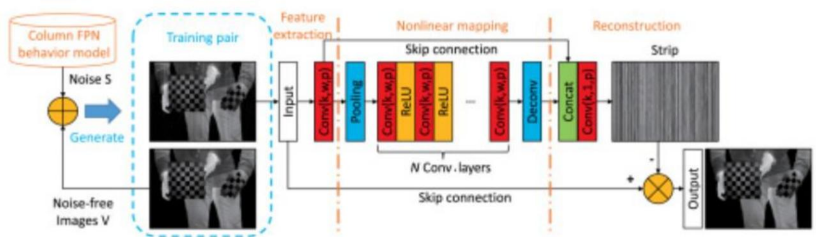

Figure 6: ConvNet used in Reference [1]: He et al, “Single-image-based nonuniformity correction of uncooled long-wave infrared detectors: a deep learning approach”.

In order to span for the four row readout native to our sensor, we decided to use convolution layers with 5x5 kernels instead of the 3x3 ones used in the paper. For the training we used the L1 loss function as recommended by Reference [1], with the given reason being that this produced more human-pleasing images than those produced utilizing the L2 norm:

$$J = \frac{1}{MN} \sum_{i \in M, j \in N} \left| P_{i,j} - T_{i,j} \right|$$

where M,N are the number of rows and columns of the input images respectively, and P is the predicted image output and T is the original clean image..

V. Experimental Results and Discussion

We used an Adam optimizer with mini-batch size of 64 sub-images. A total of 100,000 image pairs were used, in a randomly selected order, with 90,000 pairs used in the training set, and 10,000 pairs used in the test set. We set the learning rate at 1e-4 for the first 100 epochs, then 1e-6 for the next 2500 epochs, then 5e-7 for the final 1000 epochs. The final training/validation loss was 8.07e-4, equivalent to 0.21DN, which is way below human detection capability.

The flow of our training is described in Figure 7, which includes the histograms of both the original clean image and the final sanitized image, demonstrating the efficacy of the FPN removal process. The histograms before and after the FPN removal show that the model is able to correct for very low-level noise, which is necessary for correcting low-light imaging. The histograms basically demonstrate that the light condition is such that only 2% of the sensor dynamic range is being utilized, in the low range where shot noise will not hide the FPN problem.

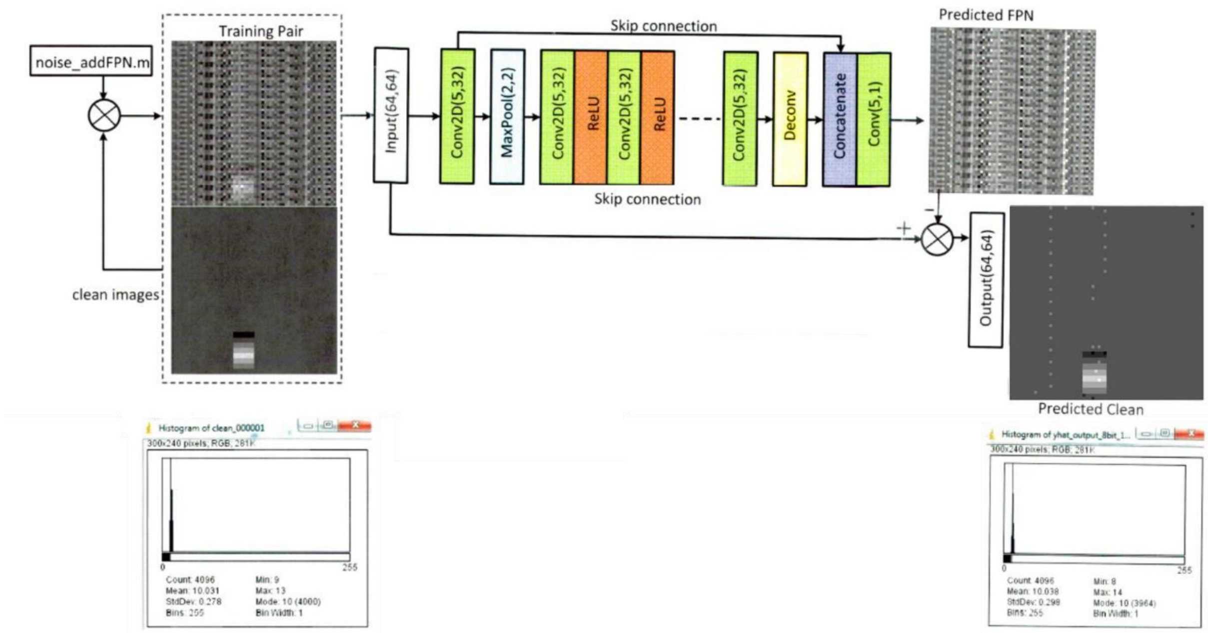

Figure 7: FPN removal training flow. Note that the histograms of both the original clean image and the final sanitized image are very similar, demonstrating the efficacy of the FPN removal process.

Table 1 below shows the summary of the training results. We also tried using the mean_abs_percent_error metric but we decided that the results from this were not meaningful, possibly due to too many zero-value pixels leading to their over-emphasis, so we ended up discarding this metric.

Loss function	final learning rate	epochs	final training loss	final validation loss
mean_abs_error mean(abs(yhat-y))	5.00E-07	3660	**8.06E-04**	**8.06E-04**
mean_abs_percent_error mean(abs(y-yhat) /max(y,epsilon))*100	2.00E-06	130	5.36E+03	5.51E+03

Table 1: Summary of results

VI. Conclusion and Future Work

From the results, we determined that using ConvNets to remove FPN in image sensors is a viable method as opposed to using complex mathematical functions that need to be tailored for each sensor. The advantage of ConvNets needing only to be trained once and gain universal FPN removal ability (for a particular sensor product) is undeniable.

More work is required to expand the effort from 64x64 images to the full 4K HD images generated from our sensor, and further on to the debayering process for colored versions of the sensors. We also plan to explore other architectures which may be more tailored to the nuances of our sensor.

VII. Member Contributions

The contributions by the team members are as follows:

Steve Mims: image collection, image manipulations, addition of FPN to images, ConvNet for FPN removal, report writing

Paul Lim : image manipulations, ConvNet for FPN Classification, setup of AWS for training, report writing

VIII. Code Repository

https://github.com/mimsborne/cs230_project

(This is a private repository with collaborator invites to cs230-stanford and CS230GIT)

IX. References

[1] He et al, "Single-image-based nonuniformity correction of uncooled long-wave infrared detectors: a deep-learning approach", Applied Optics, Vol 57, No. 18, June 2018.

[2] Keras github repository (https://github.com/keras-team/keras)