Dark Frame Generation Using pix2pix for Noise Reduction in Amateur Astrophotography

Arjun Zarzuela Lathi and Elliot Justice

1 Abstract

We investigate the use of image-to-image translation neural nets to generate dark frames, a type of calibration frame used in contemporary amateur astrophotography. Dark frames are used to subtract out fixed pattern noise, dark current noise, and thermal noise present in camera sensors which are amplified in the long exposure time necessary to image faint nebulae and galaxies. Because dark frames must be taken using the same exposure settings as the image target, they are often captured near the end of an observing session to maximize limited observing time, which restricts the dark frames effectiveness for images taken early in the session under warmer temperatures. We aim to test if generated dark frames can act as sufficient substitutes for naturally acquired dark frames when used in an image stacking workflow commonly used in amateur astrophotography.

2 Introduction

Image stacking and dark frame subtraction is the backbone image processing technique for amateur astrophotographers. It involves capturing several types of calibration images and multiple image frames of the target object. Individual images of each type are referred to as "subs", and stacked images of those subs are referred to as a "master". There are several slight variations to the technique that use different types of calibration frames. The most commonly used variation requires the use of bias subs, dark subs, flat subs, and light subs. Light subs are images of the object of interest, such as a galaxy, star cluster, or comet. Bias subs are a measure of the read noise from the camera sensor, and are taken at the fastest exposure settings the camera supports, but with the same sensor gain. Dark subs are used to subtract out fixed pattern noise, dark current noise, and thermal noise. Exposure time, sensor sensitivity settings, and sensor temperature of dark subs should match that of the light subs. Flat subs are used to correct for uneven light distribution and vignetting that originates from imperfections in the telescope system.

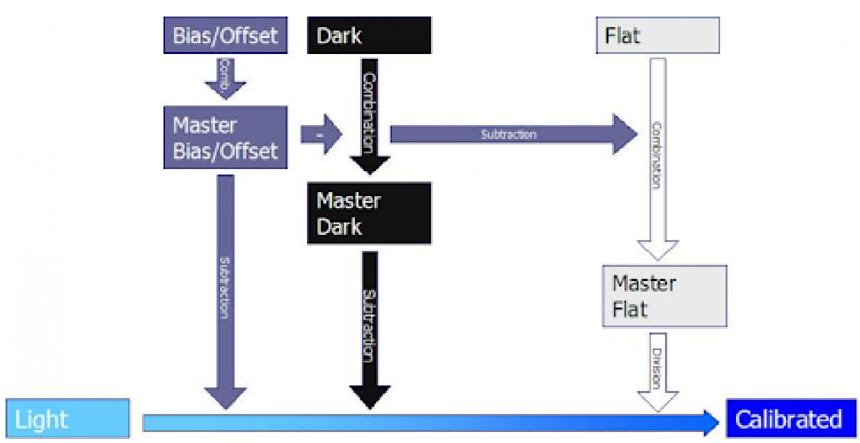

Figure 1: Dark Frame processing pipeline [20]

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Figure 1 shows the data flow for one common dark frame processing pipeline. In it, multiple bias subs are stacked into a master bias which is then subtracted from each dark sub and flat sub. The calibrated dark subs and flat subs are then stacked into a master dark and master flat. Finally the three calibration masters are subtracted from each light sub and the light subs to generate a final image. Of the three calibration frames mentioned, this project focuses on dark frames. Unlike bias and flat subs, dark frames are sensitive several condition, such as the temperature of the camera sensor and exposure time. The are also unique to each individual camera due to manufacturing defects such as hot and cold pixels..

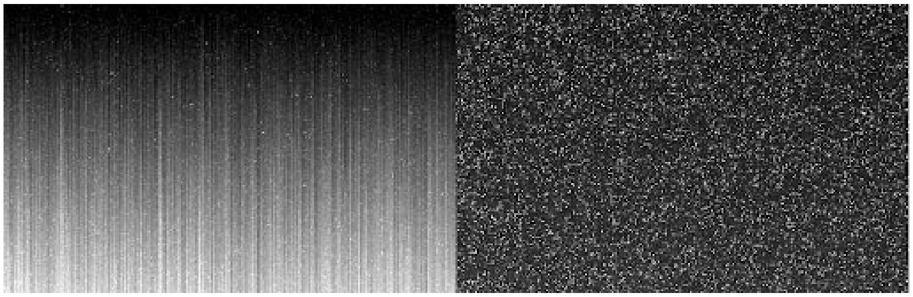

Figure 2: Dark frames at different exposure settings [22]

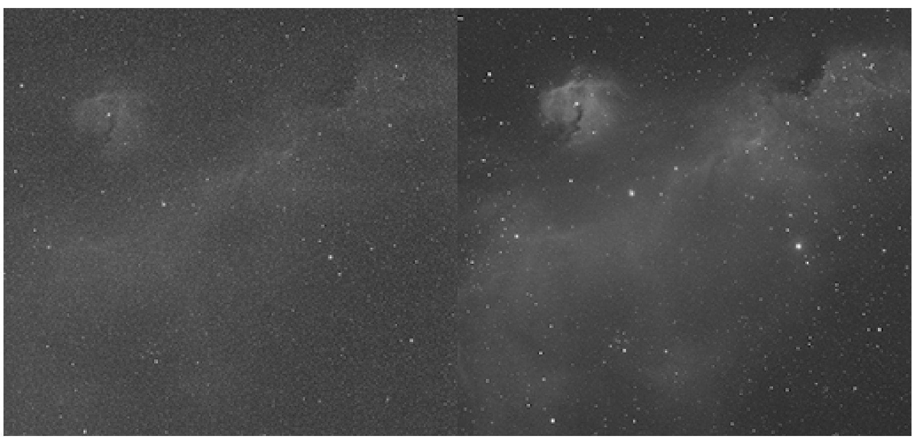

Figure 3: A light frame with unsubtracted and subtracted dark frame noise [21]

Figure 2 provides an example of how noise captured in a dark sub changes with exposure time. The right image was taken using a 1 second exposure and the right image is at 40 seconds of exposure with camera temperature held constant. Figure 3 shows an astronomical image without dark frame subtraction (left) and the same image with dark frame subtraction applied (right). Because dark subs are also sensitive to temperature changes, they are best captured after each light sub to minimize the temperature difference between the frame types. Community wisdom recommends 10-20 dark sub frames should be captured to assemble a single master dark. However, because individual light subs of an astronomical image have exposure time ranging from tens to hundreds of seconds, they are often taken at the end of an observing session to maximize image capture time of the target object which can add significant thermal variance to the dark subs.

This project's goal is to investigate if we can train a neural network to take a single dark sub frame and infer/generate a finalized master dark. This would give significant time savings in the field to amateur astrophotographers, since they would only need to capture a single dark sub instead of the recommended 10-20. For example, at 600s of exposure time per light sub (which is not uncommon), capturing 20 dark subs can take away over 3 hours of an imaging session that could otherwise be used to capture light subs. As mentioned in the project proposal, we chose to focus on dark frame generation instead of the light frames because "of the tendency of the algorithms [...]

to insert artifacts into the images from the training set used to create the network. For many in the community, these artifacts are a non-starter because they can reduce the scientific value of the image [by altering the data of the light sub], as well as destroy fine structure elements in the underlying [light sub] image."

3 Data and Methods

The data for this project comes from the author's own data collection. The full data set consists of 1900 dark subs that are paired against 86 master darks. However data processing issues resulted in us discarding 30 of the master darks and all of their corresponding dark subs, leaving us with 1100 dark subs and 59 master darks to use in training and validating the model. The data was captured using a Canon Rebel t7i that was modified to remove it's built-in infrared filter and were stored in Canon's CR2 image format at 6000x4000 pixels and 16 bit color depth. Because CR2 is a proprietary format that python cannot read, the data was converted to TIFF images using a Photoshop script so it could be processed by the network during training and testing.

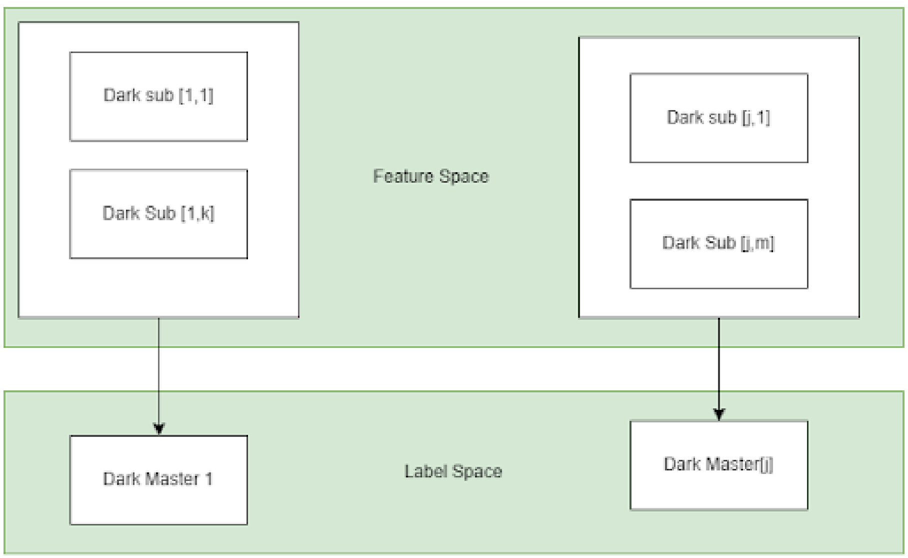

Figure 4: Feature and label space pairings

Figure 4 shows the mapping for the model's feature and label sets. Individual dark subs represent the feature space for the network and the master darks are the label space. This pairing is used because of the stacking process used to create the dark masters for the label set. We also utilize the fact that because the noise in the dark subs is proportional to exposure time and temperature, we can use the dark sub to implicitly represent these properties in the network and avoid having to explicitly specify them as additional feature vector elements.

For this project we chose the pix2pix [3] cGAN neural network to perform the image-to-image translation. The generator in pix2pix follows the U-net encoder-decoder model with skip connections and has convolution-BatchNorm-ReLu hidden layer modules in both the encoder and decoder. The discriminator uses a PatchGAN, which was introduced by [3] specifically for pix2pix. Unlike normal discriminators that classify an entire image as real or fake, the PatchGAN classifies each an N by N patch of the images as real or fake. Together the generator and discriminator are working to optimize the following objective function.

$$G* = arg\ min_G max_D L_{cGAN}(G, D) + \lambda L_{L1}(G)$$

$$L_{cGAN}(G, D) = E_{x,y}[logD(x, y)] + E_{x,z}[log(1 - D(G(x, z))]$$

$$L_{L1}(G) = E_{x,y,z}[\|y - G(x,z)\|_1]$$

Where L_{cGan} is the loss of the Conditional GAN and L_{L1} is the L1 error distance of the generated image from the ground truth image. As described in [3] L1 loss is used instead of L2 because L1 loss tends to produce sharper images.

Once the network has been trained, we then use the generated master darks to assemble a stacked calibrated image of an astronomical object and compare that to the stacked+calibrated image using the original dark master. Due to time constraints, we were only able to prepare a single image for comparison. We use both a visual inspection of the two calibrated images as well as a comparison of their peak signal-to-noise ratio. For the PSNR analysis, we will use a stacked image of the light subs without any subtraction as the baseline by comparing PSNR(Calibrated[pix2pix], noSubtraction) to PSNR(Calibrated[realDarkMaster], noSubtraction).

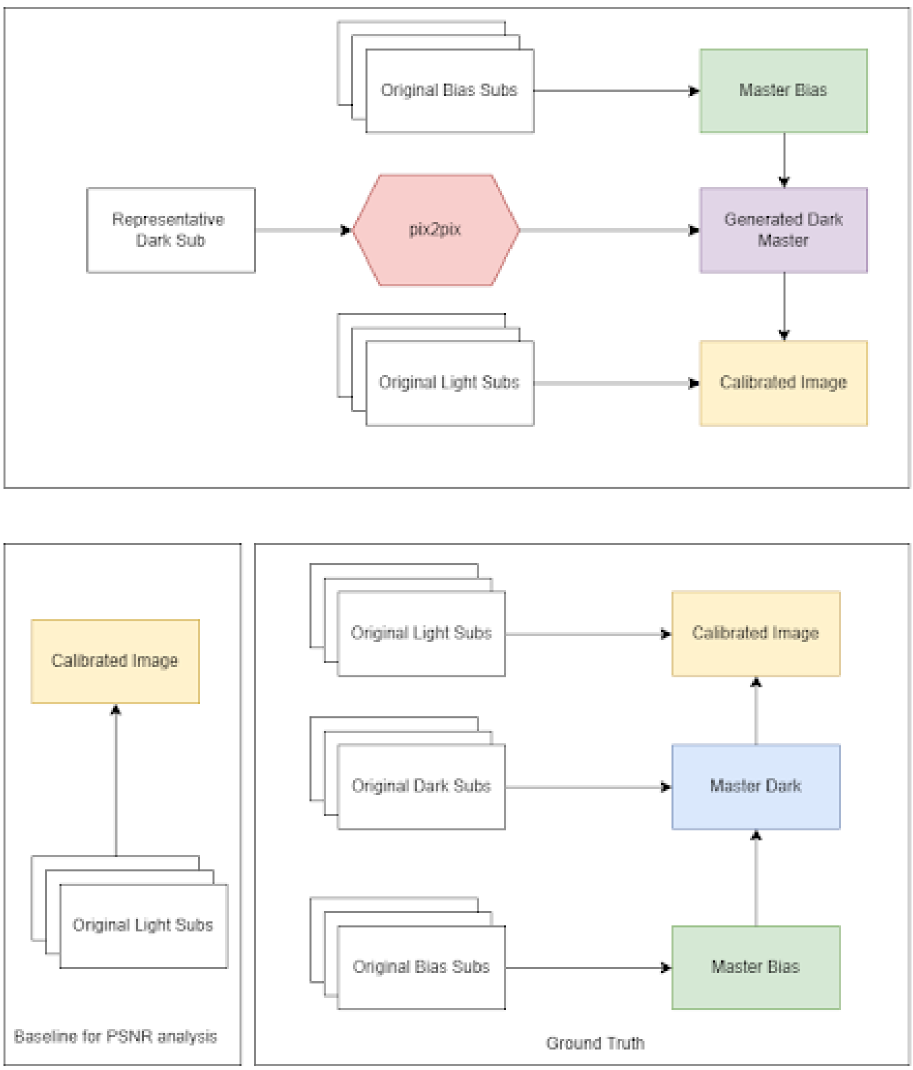

Figure 5: Generation of images for analysis

4 Analysis

Due to the very large dimensionality of the training data, we anticipated that we would have to resize the images in order to train the network on a GPU. As a pre-processing step, we cropped all images down to 1500px by 1000px using a Photoshop script. This first training attempt did not go well, as we found that pix2pix would throw various tensor dimension errors for images that were not in a 1:1 aspect ratio. Reviewing [3] again we determined that this restriction was likely caused by the N by N filter used by the PatchGAN discriminator. In addition, memory limitations on the GTX 3070 forced us to further restrict to images size of at most 1024px by 1024px. Late in the project we realised that we could have instead taken the train+test sets and sliced both the dark subs and dark master into individual 1024x1024 images and then re-assemble a complete master dark from the corresponding slices. Unfortunately we ran out of time to execute this approach.

For the first training pass we ran at 200 epochs with the 1500x1000 images. The first pass ended up generating images that were nothing but flat gray images. We reasoned that because the data was already downsampled, we needed to go back and downsample from the original 6000 by 4000 images so that PyTorch was not trying to downsample an image that was already downsampled to 1500x1000.

The second training pass was more promising although not without its own issues.

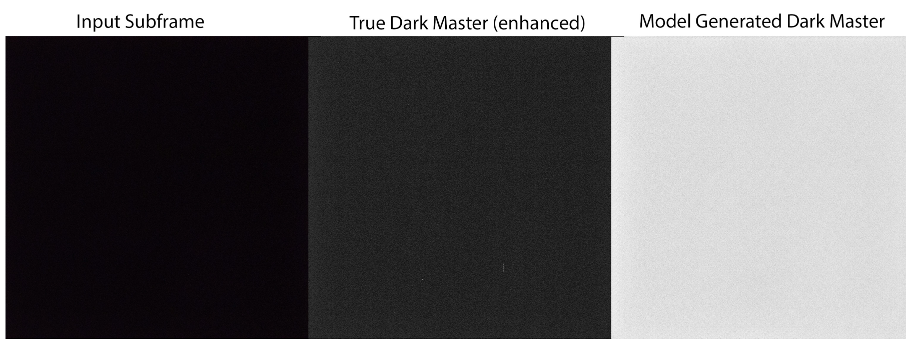

Figure 6: Sample output from second training attempt

Figure 7: Training Graph for Run 2

As seen in figure 6 the generated dark masters from this attempt contained elements of the noise pattern in dark masters, but the output was inverted. Reviewing reported issues in the pix2pix repo, we determined the issue was a result of the training data using 16-bit color channels instead of 8-bit color channels. As a result the larger precision was causing overflows when the model output was written to disk. Still, there were noise patterns present in the testing result which is what we would expect if the model was generating a true dark master.

In our final attempt to get the model outputting meaningful darks master, we converted the TIFF images to JPEG files using a Photoshop script to make sure all the training data would be using 8-bit color channels. We trained this for 200 epochs and again tested the results.

Figure 8: Sample output from final training attempt

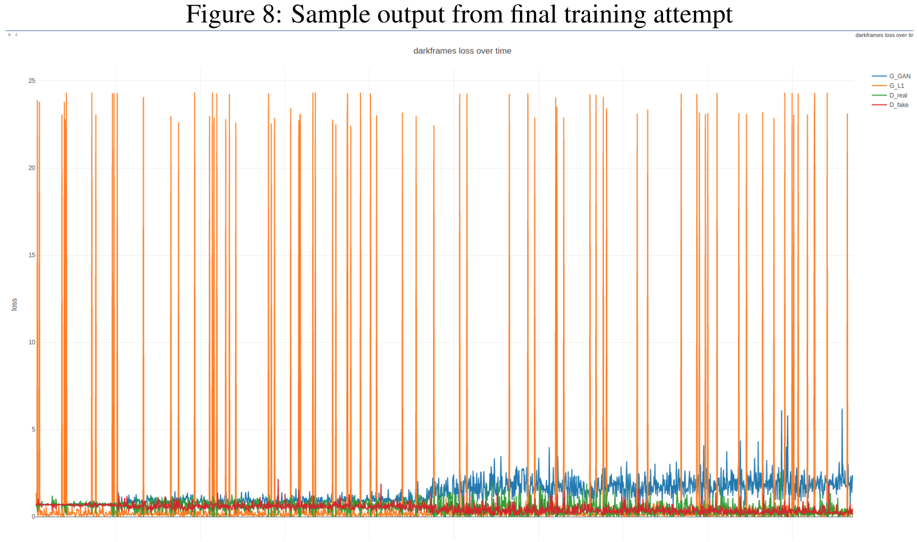

Figure 9: Training Graph for Run 3

Here, the results finally looked more along the lines of what we were expecting, as they appeared to be black/grey with some slight noise patters throughout the image. To better distinguish the differences within Figure 8, both the true dark master and the generated dark master have been amplified using a Photoshop Exposure layer. In Figure 6, only the True Dark master was amplified.

We used the generated dark master to subtract noise out of the original image using the stacking method described earlier. We also used the canonical dark master to subtract noise out to compare the effectiveness of the generated one.

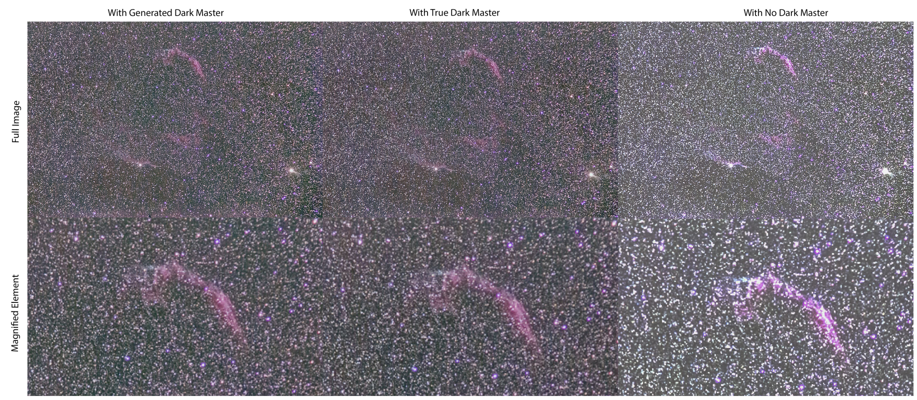

Figure 10: From left to right: With generated dark master, with true dark master, with no dark master

Figure 10 shows an image stack of the Veil Supernova Complex in Cygnus taken composed of thirty four 180 second exposures at 6400iso. The top row shows the full field of view and the bottom row shows a close up of the Eastern Veil nebula locate in the top center of the field of view. The right most image has no dark frame subtraction applied, to serve as a baseline. The middle images has the original dark master and the left most image has the generated dark master subtracted. Although difficult to tell in the embedded images we can see that our generated dark master did a surprisingly good job at subtracting out noise from the original image. to the point where the generated and true dark master are almost indistinguishable. To get a closer look, we can show the following images:

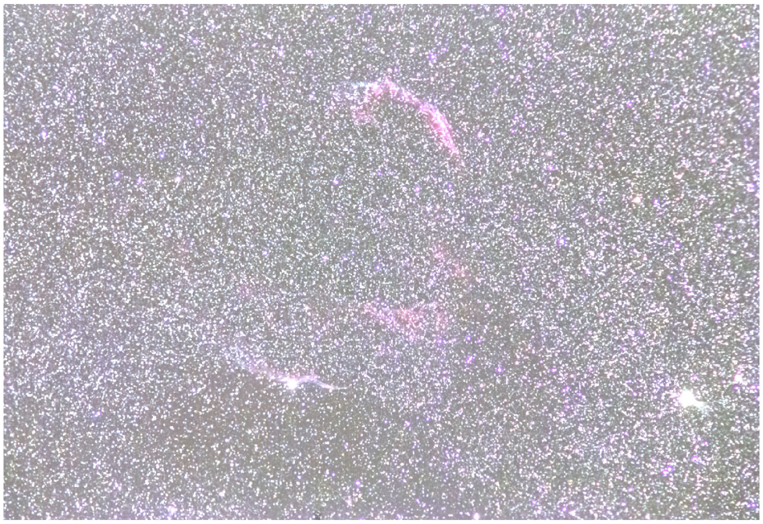

Figure 11: The baseline image

Figure 12: The true dark stacking

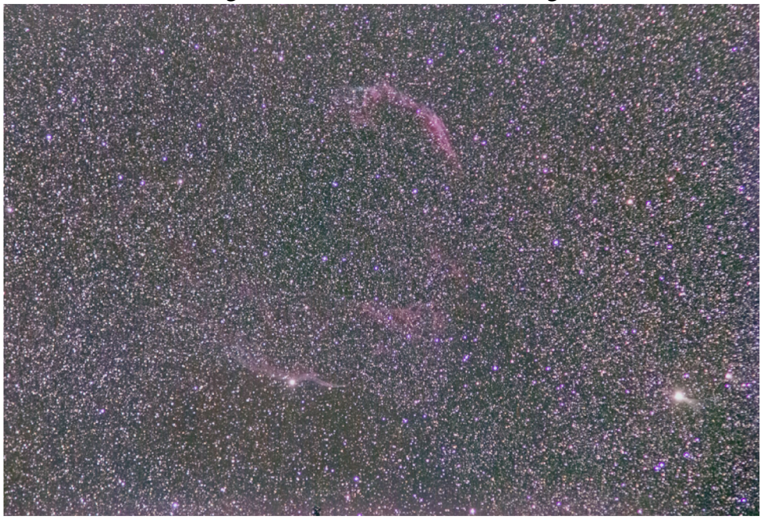

Figure 13: The generated dark stacking

The peak-signal-to-noise values for the generated dark w.r.t to the image with any dark frame subtraction was 27.688395, while the PSNR for the true dark was 27.684649.

5 Conclusions

The results turned out a lot better than expected. When training the images and watching the training graph, the network did not look promising. Also, when generating the dark masters, it was hard to make sense of the results, because they are essentially just blank images of random noise patterns. Through our training process, we also ran into numerous issues. One was the actual time it took to train the network. To run for 200 epochs, our model had to train the network for approximately 16 hours. This led to long times in between being able to view and improve upon results. We had issues when using the .tif files, and then had to convert them to .jpg format for them to work due to the truncating of pixel values. This led to a loss of detail, as 8 bit color channels are not as fine grained as 16 bit color channels values, which could be problematic with how abstract and sensitive the dark frames are, being just images of noise. However, when we finally tested stacking the dark master onto the baseline we were pleasantly surprised with how the results turned out, as they were extremely close to the true dark master.

Given more time, we could likely have worked around thee need for cropping the data down to 1024x1024 images to maximize the value of the original image. However, the main focus of this project was to see if it was even possible to generate dark masters with a neural network, and for that goal, we can say that we were moderately successful. We approached this project uncertain if we would find any success at all, as pix2pix generally is used to generate smaller, less detail-focused image transformations, such as black and white to colored images, or horses to zebras. Our project used pix2pix for a problem that used images that made little sense to the human eye, and to see that a neural network was able to make much more sense of these noise images than a human might was extremely interesting to look into.

6 References

[1] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, Chia-Wen Lin (2020) "Deep Learning on Image Denoising: An overview". arXiv:1912.13171

[2] Ireland, Tom C. (2019) "Improving Photometry and Astrophotography by Eliminating Dark Frames and Flat Fields". Honors Theses and Capstones. 463. `https://scholars.unh.edu/honors/463`

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros (2016) "Image-to-Image Translation with Conditional Adversarial Networks" arXiv:1611.07004v3

[4] Deep Learning for random noise attenuation (2019) `https://www.cloudynights.com/topic/657928-deep-learning-for-random-noise-attenuation`

[5] Machine Learning algorithm that significantly improves quality of low-light images (2018) `https://www.cloudynights.com/topic/619142-machine-learning-algorithm-that-significantly-improves-quality-of-\low-light-images/`

[6] Chen Chen, Qifeng Chen, Jia Xu, Vladlen Koltun (2018) "Learning to See in the Dark." arXiv:1805.01934v1

[7] Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky (2017) "Deep Image Prior." arXiv:1711.10925

[8] Alexander Krull, Tim-Oliver Buchholz, Florian Jug (2018) "Noise2Void - Learning Denoising from Single Noisy Images." arXiv:1811.10980v2

[9] Chaoyue Wang, Chang Xu, Chaohui Wang, Dacheng Tao (2017) "Perceptual Adversarial Networks for Image-to-Image Transformation." arXiv:1706.09138v2

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros (2017) "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." arXiv:1703.10593v7

[11] Shorten, C., Khoshgoftaar, T.M. (2019) “A survey on Image Data Augmentation for Deep Learning.” J Big Data 6, 60. https://doi.org/10.1186/s40537-019-0197-0

[12] Curtis A Corum (1999). “Dark frame subtraction.” US Patent US6101287A. https://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&d=PALL&s1=6101287.PN.

[13] Michael Goesele, Wolfgang Heidrich, Hans-Peter Seidel (2001). “Entropy-Based Dark Frame Subtraction.” https://download.hrz.tu-darmstadt.de/media/FB20/GCC/paper/Goesele-2001-EBD.pdf

[14] Levesque, Martin P. Lelievre, Mario (2007). “Evaluation of the Accuracy of the Dark Frame Subtraction Method in CCD Image Processing”. Defense Research and Development Canada. https://apps.dtic.mil/sti/citations/ADA479357

[15] Newberry, Michael V. (1991). “SIGNAL-TO-NOISE CONSIDERATIONS FOR SKY-SUBTRACTED CCD DATA.” Publications ofthe Astronomical Society of the Pacific https://iopscience.iop.org/article/10.1086/132801/pdf

[16] I. Takayanagi, J. Nakamura, E. R. Fossum, K. Nagashima, T. Kunihoro and H. Yurimoto,(2003) ”Dark current reduction in stacked-type CMOS-APS for charged particle imaging,” in IEEE Transactions on Electron Devices, vol. 50, no. 1, pp. 70-76, doi: 10.1109/TED.2002.806963.

[17] N. V. Loukianova et al., (2003) ”Leakage current modeling of test structures for characterization of dark current in CMOS image sensors,” in IEEE Transactions on Electron Devices, vol. 50, no. 1, pp. 77-83, doi: 10.1109/TED.2002.807249.

[18] William C. Porter, Bradley Kopp, Justin C. Dunlap, Ralf Widenhorn, Erik Bodegom,(2008) ”Dark current measurements in a CMOS imager,” Proc. SPIE 6816, Sensors, Cameras, and Systems for Industrial/Scientific Applications IX, 68160C; https://doi.org/10.1117/12.769079

[19] Nikita Misiura (2018) ”StarNet”. https://github.com/nekitmm/starnet/

[20] Unspecified. “Frame Calibration types” http://deepskystacker.free.fr/english/index.html

[21] Richard S. Wright Jr. (2021) “DARK FRAMES AND BIAS FRAMES DEMYSTIFIED” . Sky and Telescope Magaizne. https://skyandtelescope.org/astronomy-blogs/imaging-foundations-richard-wright/dark-frames-and-bias-frames-demystified/

[22] Unknown. “CCD Bias and Dark Frames.” https://www.carleton.edu/goodsell/observing/knowledge/topics/introtoccd/ccdimages1/

[23] Unknown. “How Pix2Pix works?”. https://developers.arcgis.com/python/guide/how-pix2pix-works/