1.1.1: RAW image conversion

```
C:\Windows>dcraw -4 -d -v -w -T C:\Users\shrey\OneDrive\Desktop\590_Assignment3\data\baby.nef
Loading Nikon D3 image from C:\Users\shrey\OneDrive\Desktop\590_Assignment3\data\baby.nef ...
Scaling with darkness 0, saturation 16383, and
multipliers 1.628906 1.000000 1.386719 1.000000
Building histograms...
Writing data to C:\Users\shrey\OneDrive\Desktop\590_Assignment3\data\baby.tiff ...
```

1.1.2: Python initials

If the image has 3 channels (RGB):
Bits per pixel: **16 bits**
Width: The image width is **4284 pixels**.
Height: The image height is **2844 pixels**.

1.1.3: Linearization

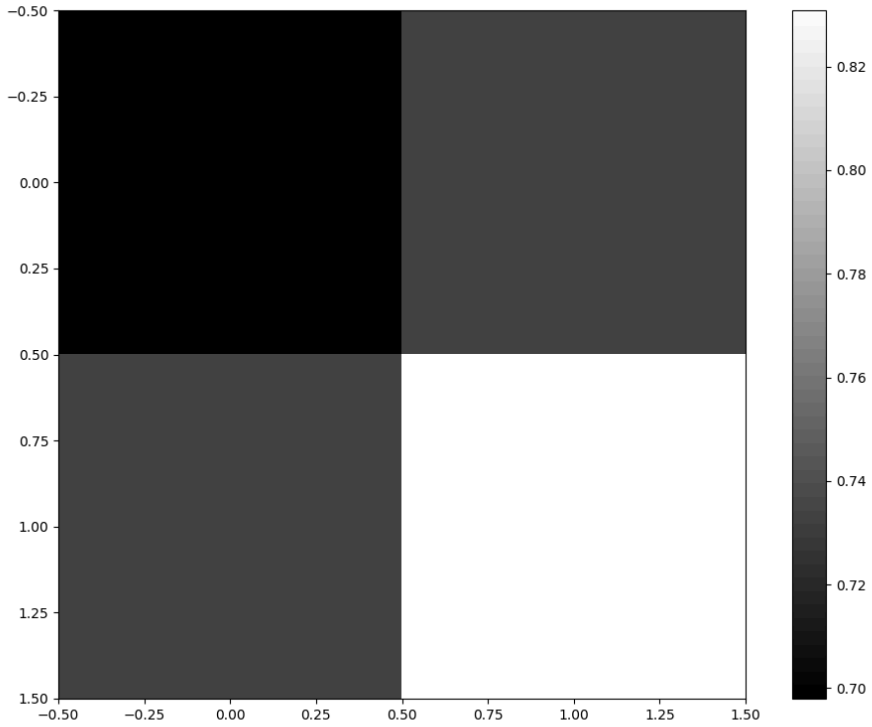
Done in code
Double Image:

Linearized Image:

1.1.4: Identifying the correct Bayer pattern

The Bayer pattern is likely: **rggb**

I determined this through outputting a gray scale 2x2 pattern for the top left pixels in the image. I narrowed it down to either rggb or bggr based on the pattern image. Based on the values of the pixel values. Additionally, I had tried a crop algorithm built into a python library that tested 100x300 sections of the image and it also confirmed the rggb color pattern.



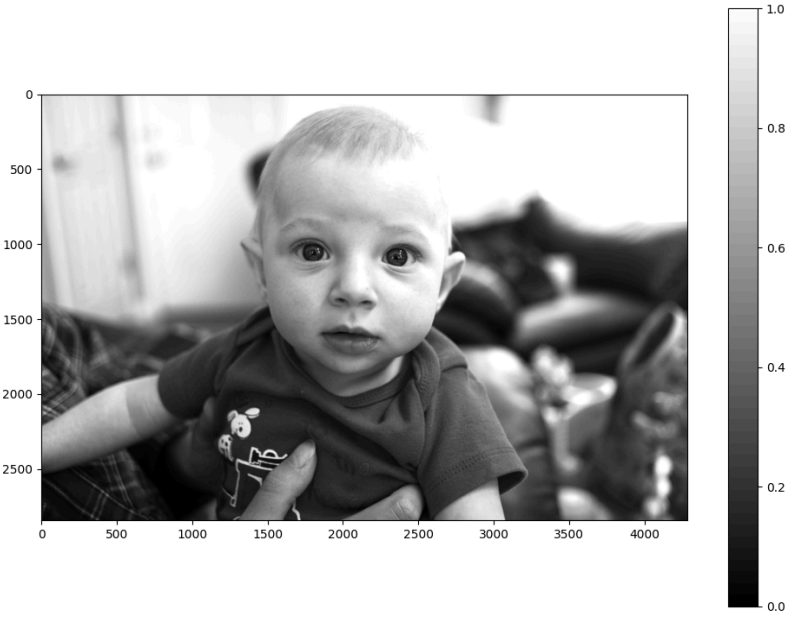
1.1.5: White balancing

To me, they all looked very similar, but personally, I think the camera **preset** one looks the best. The white world one looks good too, but I feel it looks a little extra bright.

White World:


0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
1.0
0.8
0.6
0.4
0.2
0.0

Gray World:

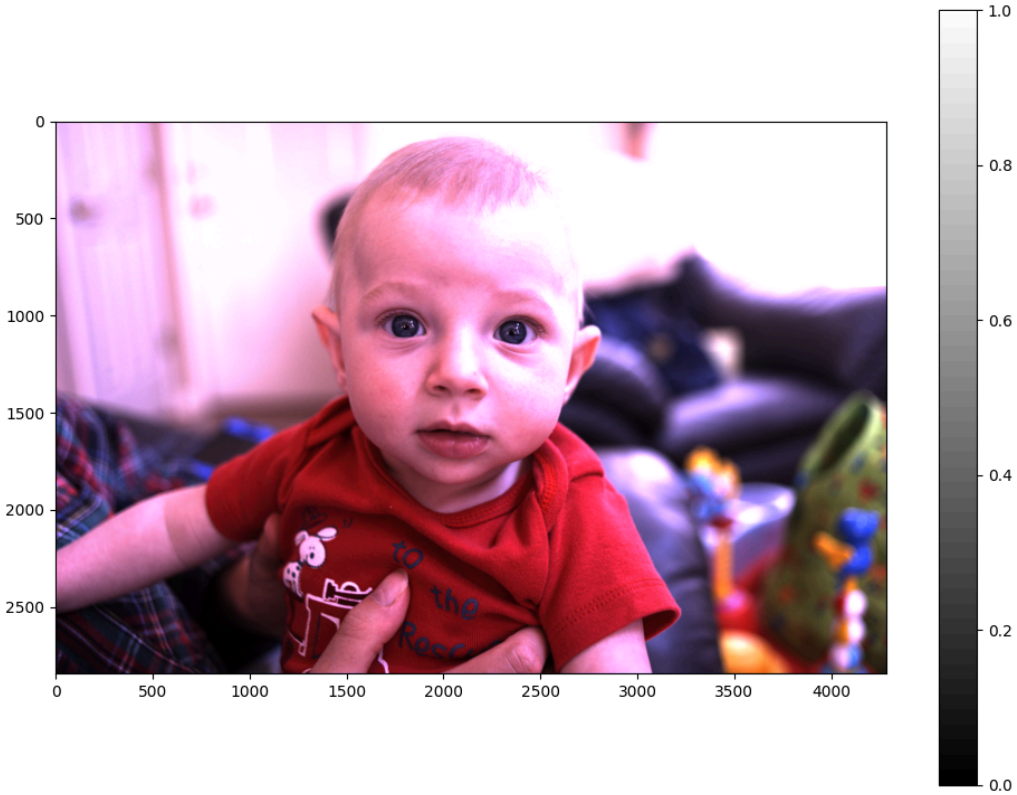

0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
1.0
0.8
0.6
0.4
0.2
0.0

Preset:

1.1.6: Demosaicing

Done in code

Demosaiced from Linearized



Demosaiced from Linearized Bayer Pattern

Demosaiced from White World

Demosaiced from Gray World

Demosaiced from Preset

1.1.7: Color space correction

Done in Code. Even after (I believe) correctly implementing the color correction, there is still a strong pinkish/redish/purplish hue to the image. After testing several different solutions, this problem persisted, leading me to believe it is an issue with the environment or dcraw.

Color Correction Linearized:

Color Correction White World:

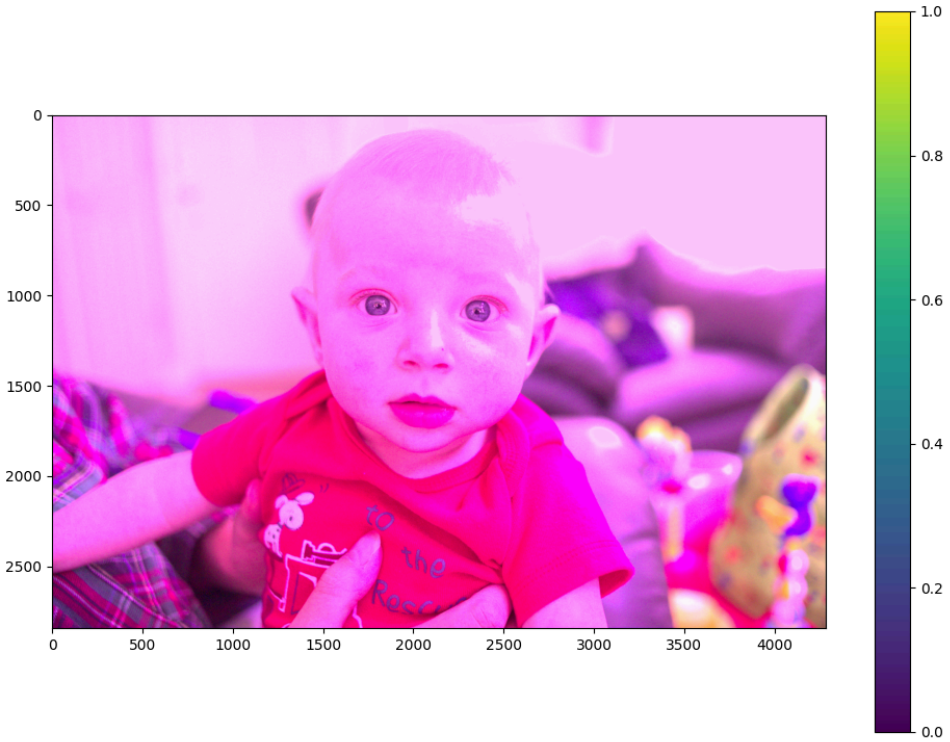
Color Correction Gray World:

Color Correction Preset:

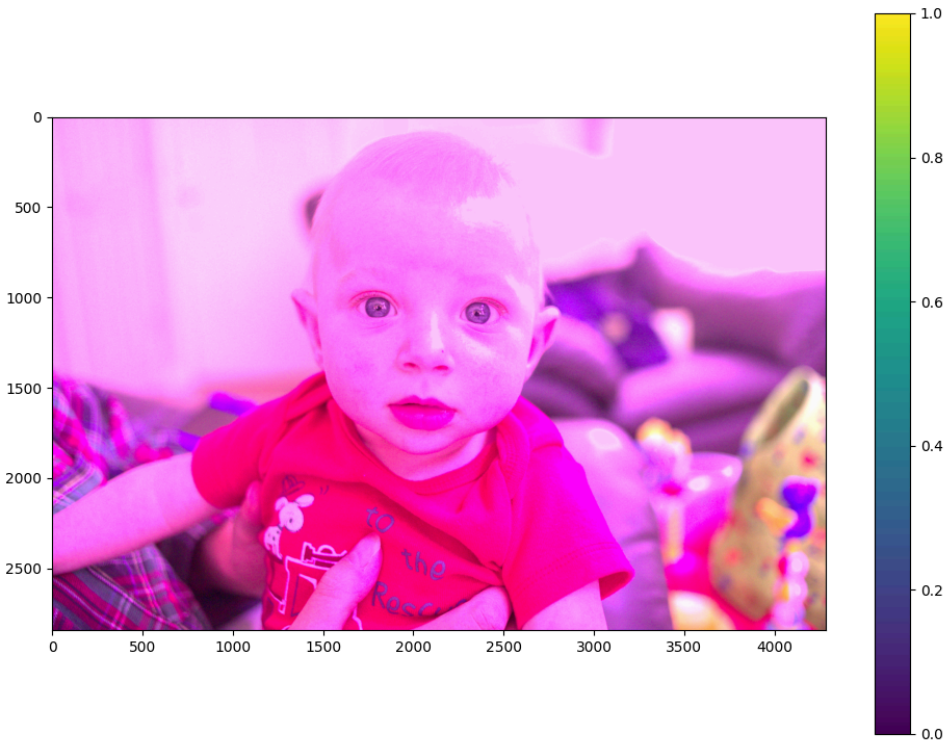
1.1.8: Brightness adjustment and gamma encoding

After experimenting and trying different percentages, **mean = 0.25** looked the best in the preset image to me. The other higher values seem to overexpose the image too much. For some reason, something in my code, either for white balancing or gamma adjusting, or dcraw, made the picture more reddish/pink. I attempted several different fixes and spent dozens of hours trying to fix this, but I was unable to alter the color tones to be correct (attempted fixes messed it up in other ways, sometimes shifting green, blue, etc.).

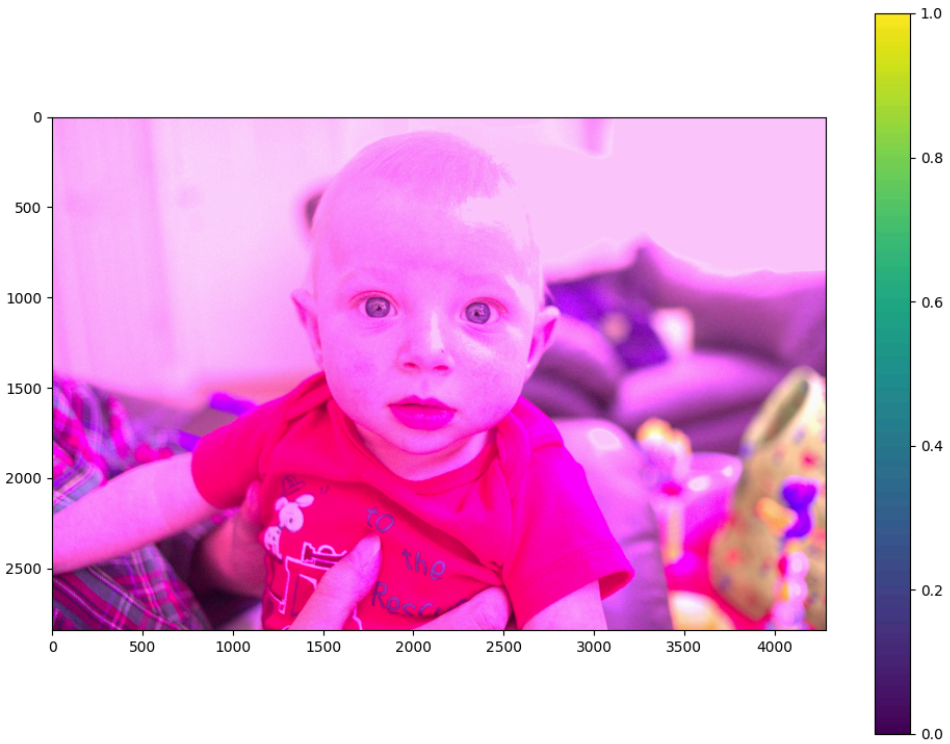
Bright Linearized:



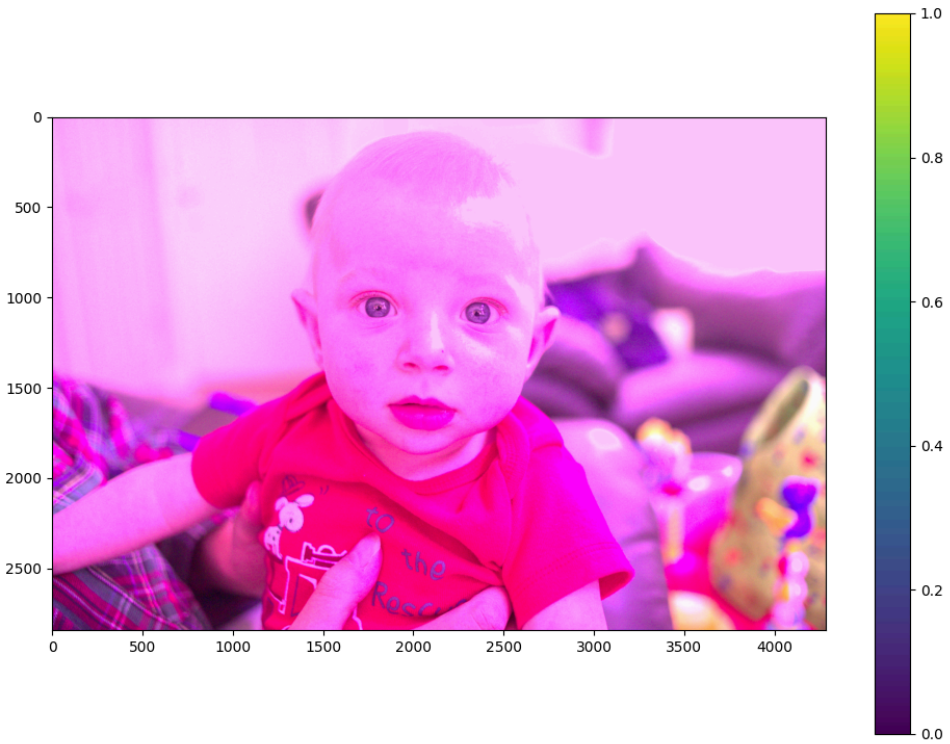
Bright and Gamma Adjusted White World:



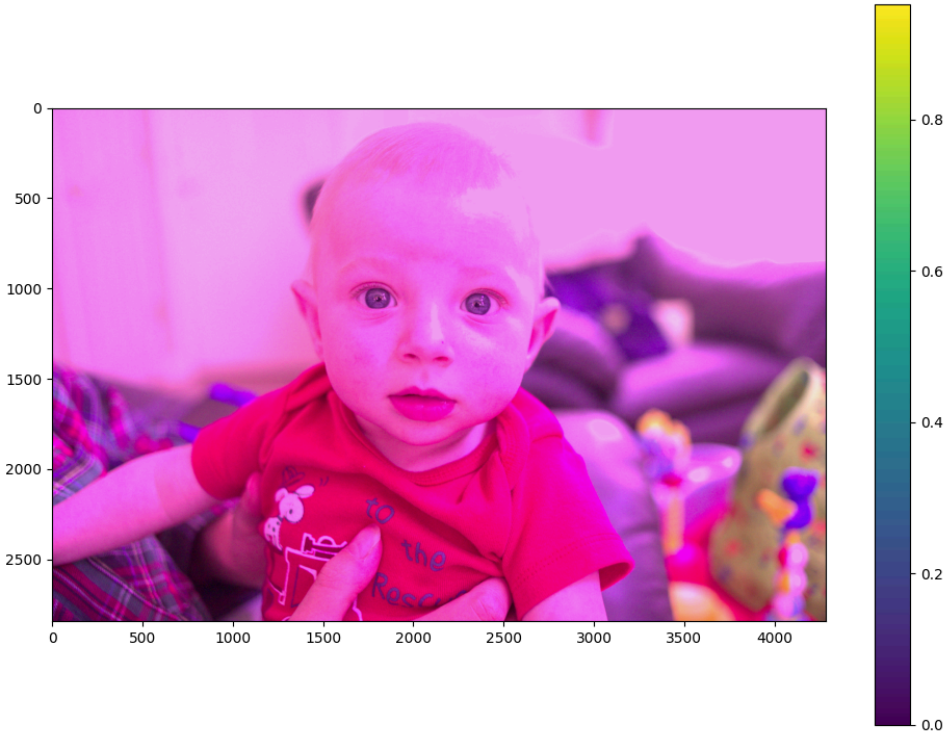
Bright and Gamma Adjusted Gray World:



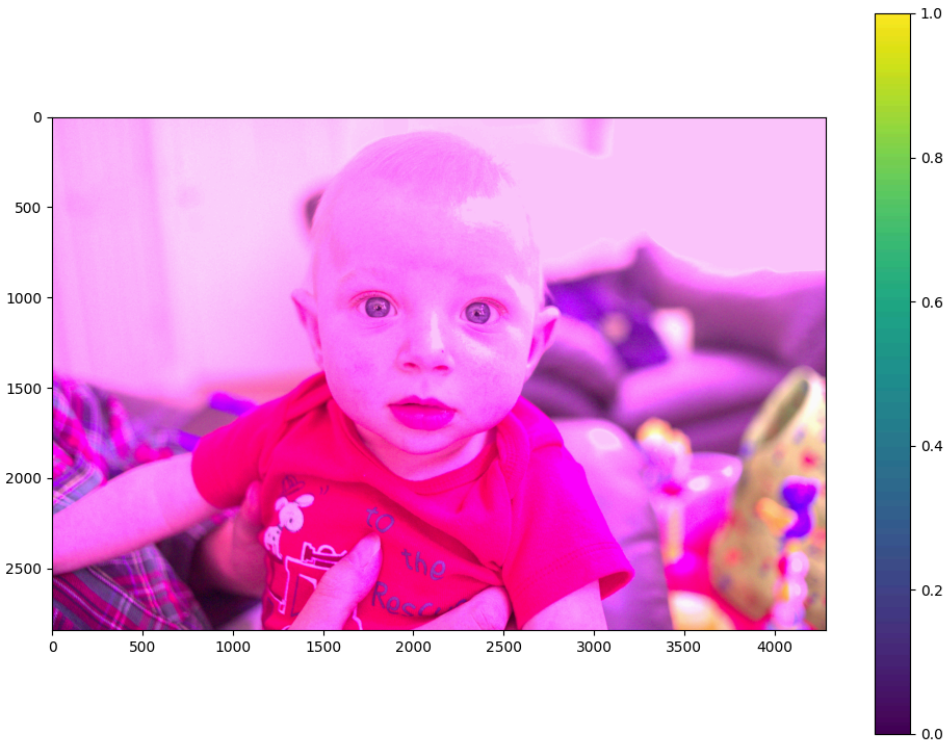
Bright and Gamma Adjusted Preset:



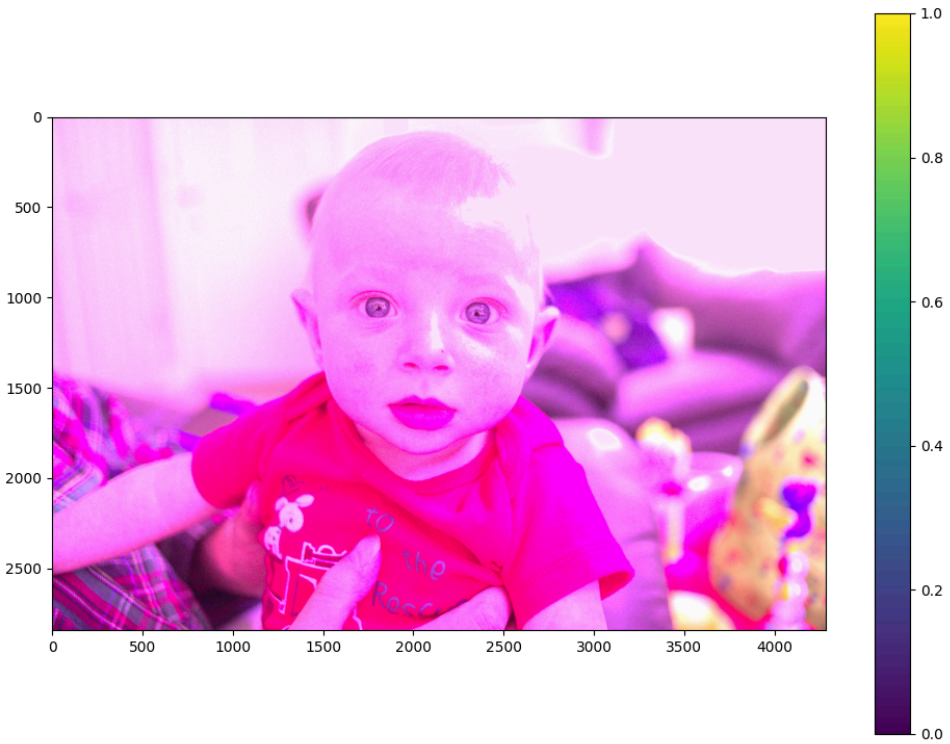
Bright & Gamma Adjusted (mean = 0.25):



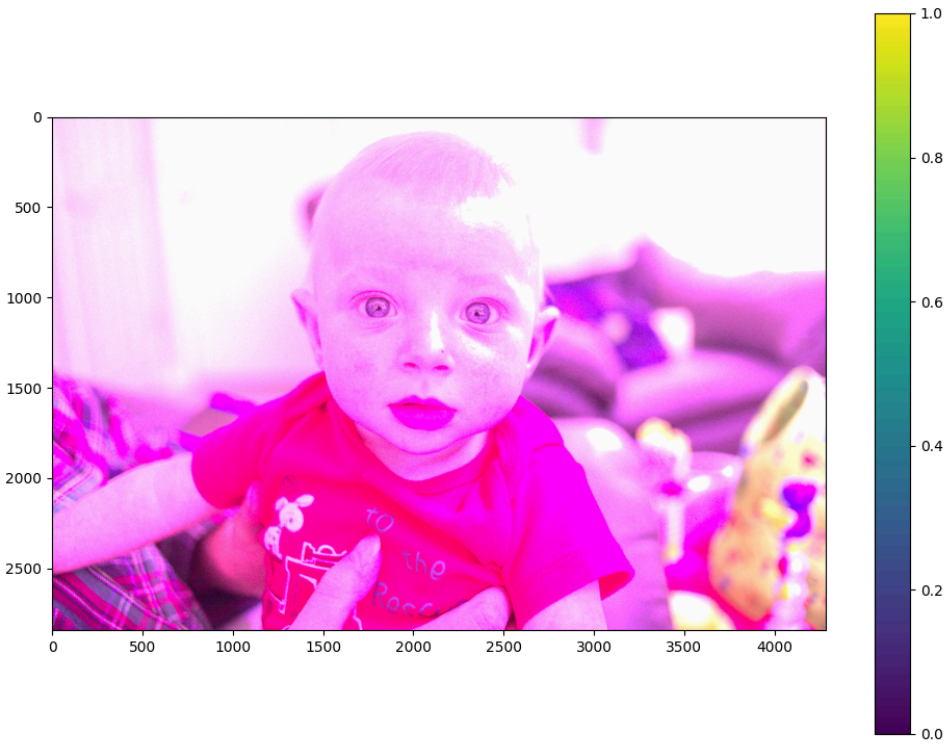
Bright & Gamma Adjusted (mean = 0.50):



Bright & Gamma Adjusted (mean = 0.75):



Bright & Gamma Adjusted (mean = 0.95):



1.1.9: Compression

I was unable to see a difference in quality for the .PNG image and the .JPEG format with the quality parameter set to .95. The compression ratio between the .PNG and .JPEG was approximately **1.25:1**. After changing the .JPEG quality settings, the lowest setting in which I could not distinguish it from the original was at **quality=50**. For this, the **compression ratio was 2.79**.

```
Compression ratio (quality=95) for Final Image: 1.18
Compression ratio (quality=90) for Final Image: 1.18
Compression ratio (quality=85) for Final Image: 1.36
Compression ratio (quality=80) for Final Image: 1.59
Compression ratio (quality=75) for Final Image: 1.88
Compression ratio (quality=70) for Final Image: 2.13
Compression ratio (quality=65) for Final Image: 2.25
Compression ratio (quality=60) for Final Image: 2.40
Compression ratio (quality=55) for Final Image: 2.66
Compression ratio (quality=50) for Final Image: 2.79
Compression ratio (quality=45) for Final Image: 3.05
Compression ratio (quality=40) for Final Image: 3.27
Compression ratio (quality=35) for Final Image: 3.51
Compression ratio (quality=30) for Final Image: 3.77
Compression ratio (quality=25) for Final Image: 3.99
Compression ratio (quality=20) for Final Image: 4.44
Compression ratio (quality=15) for Final Image: 5.39
Compression ratio (quality=10) for Final Image: 6.25
Compression ratio (quality=5) for Final Image: 6.17
```

1.2: Perform manual white balancing

After testing several patches, image (1) had the best results. The image, even though the white balance wasn't properly applied prior to selecting the patch, balanced it very well. However, most patches made the white balancing worse, so they would have to be very carefully selected, and often, it would be luck if the image turned out better or worse after the patch selection.

(1)

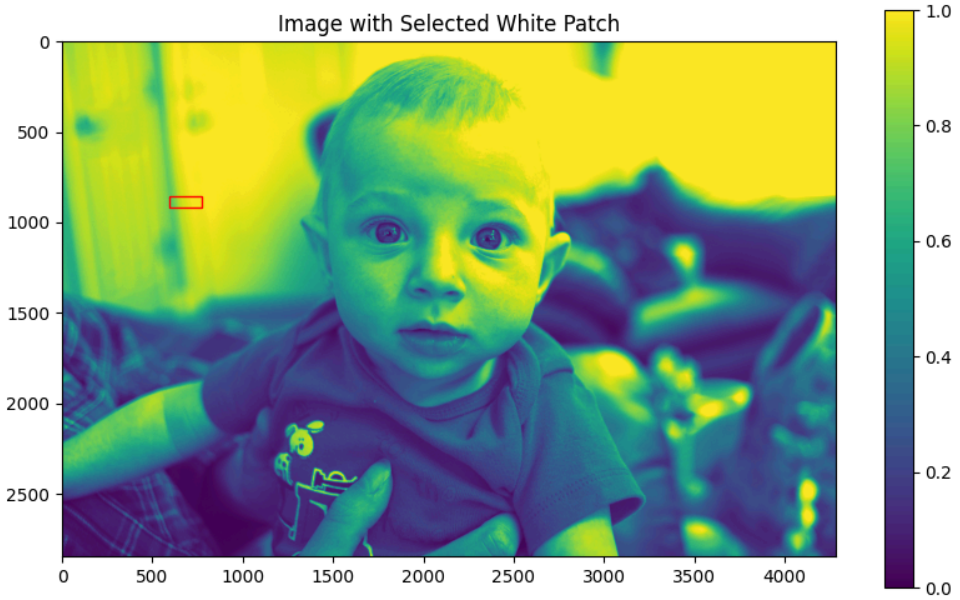
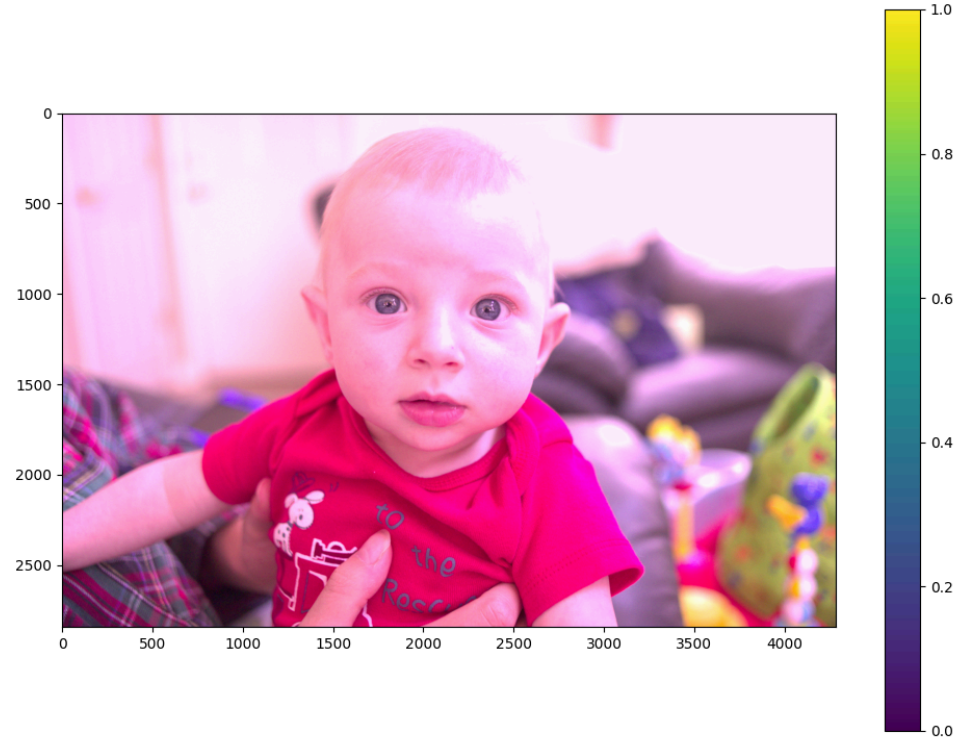


Image with Selected White Patch
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
1.0
0.8
0.6
0.4
0.2
0.0


0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
1.0
0.8
0.6
0.4
0.2
0.0

(2)


0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
1.0
0.8
0.6
0.4
0.2
0.0

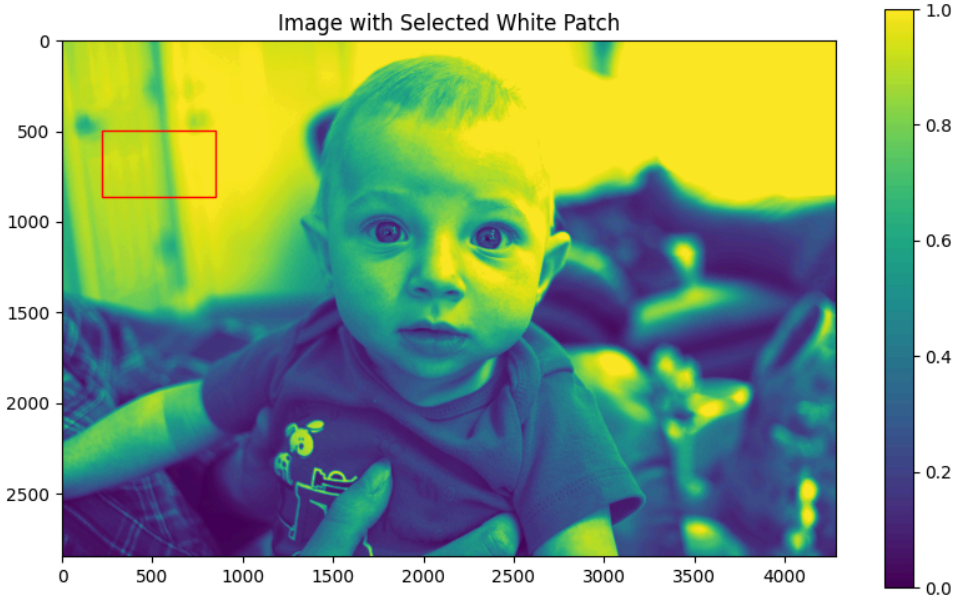


Image with Selected White Patch
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
3500
4000
1.0
0.8
0.6
0.4
0.2
0.0

2.1: Build the pinhole camera

Design Decisions: Screen size ~7inx9in, focal length ~8in, the field of view (FOV) for the pinhole camera is approximately 47.26 degrees horizontally (based on the 7-inch width) and 58.72 degrees vertically (based on the 9-inch height). The average FOV for this pinhole camera setup is about 52.99 degrees.

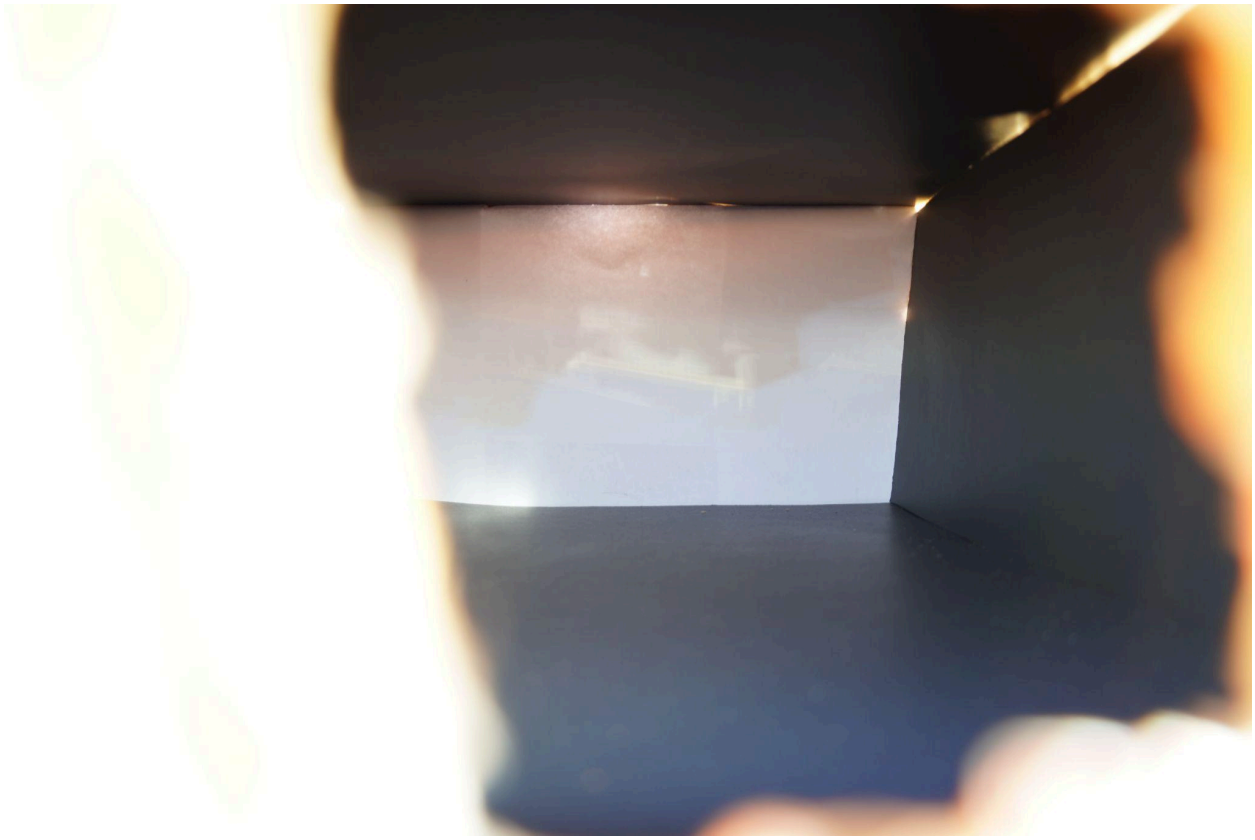
All of the edges of the box were sealed off with thick black paper taped tightly to the box, and all of the inside of the box was covered in the thick black non-reflective paper as well, ensuring no light enters other than through the pinhole.The cutout for the camera was a ~1x1in-1.2in square, and the edges around the camera were covered with a thick black shirt when placed against the camera hole so that no additional light enters.

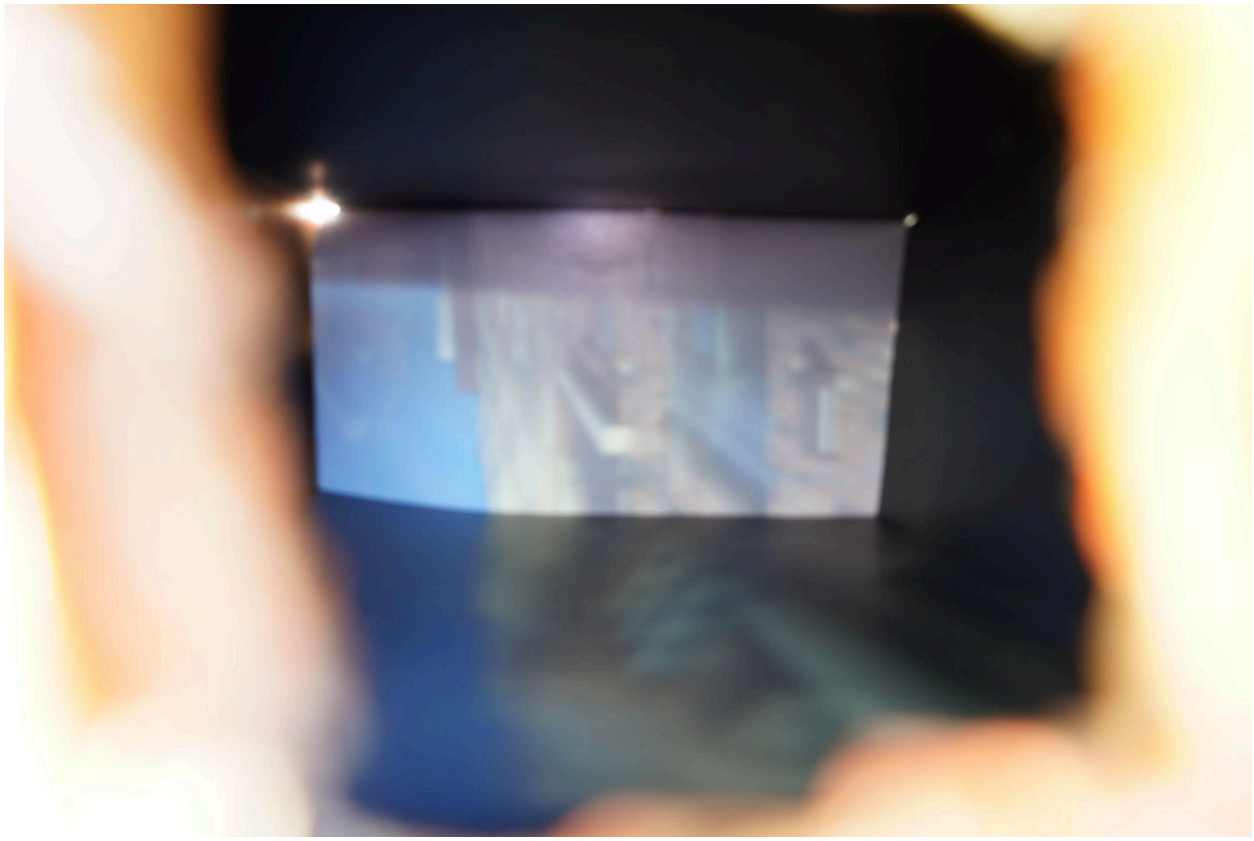
2.2: Use your pinhole camera

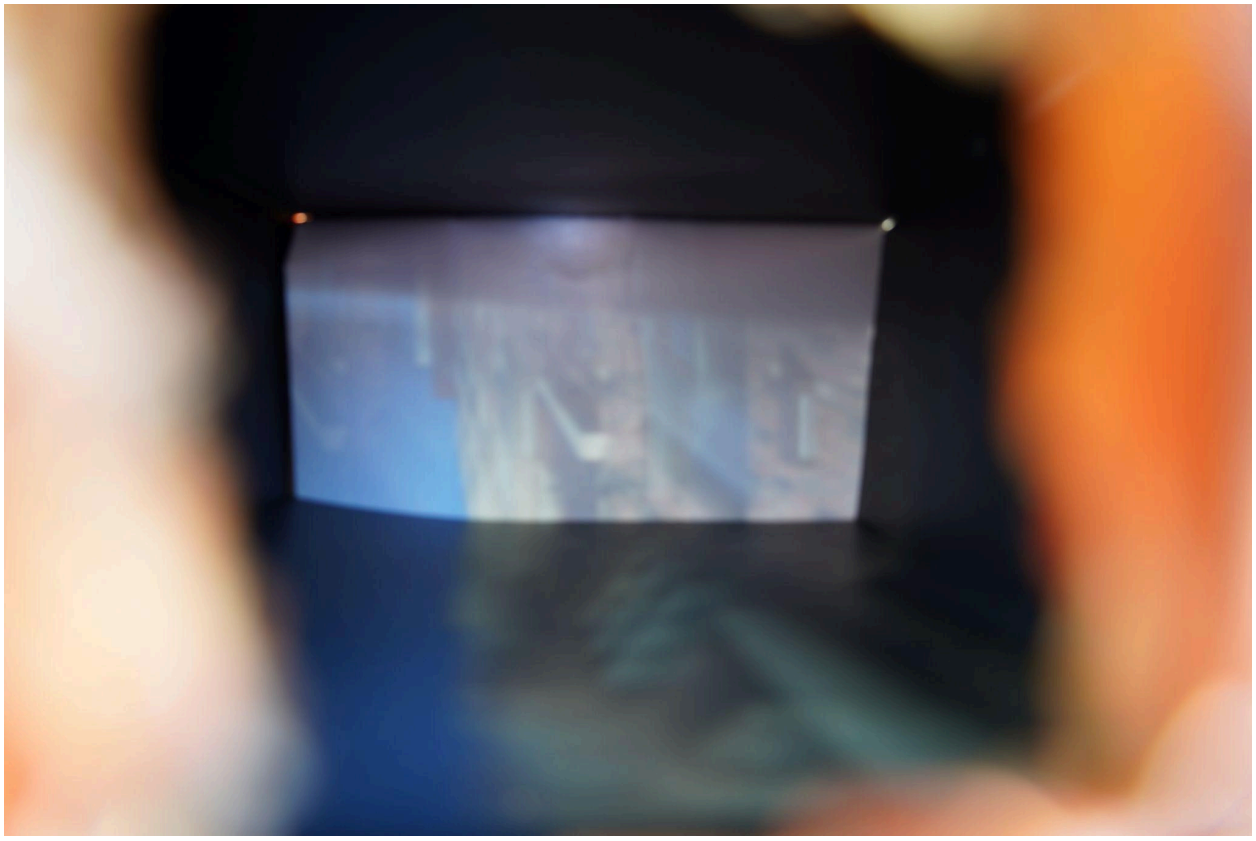
The pinhole diameters used were .1mm, .5mm, and 2mm. Overall, the 2mm let in a sufficient amount of light to the point where the images were being displayed relatively clearly. The smaller pinholes simply did not let enough light through to form bright enough images, even with a 30 second shutter time. It took ~35 images to get 9 usable ones, encountering several errors in many of them. Additionally, the first pinhole camera I built let too much light bleed in, so I built a new one that functioned significantly better.

Picture of Building Outside

These pictures were taken outside of the Carolina Square Apartment Complex in Chapel Hill, NC. These were taken with the first pinhole camera that I had made, as seen by the light bleeding in. The pinhole diameters were 0.1mm, 0.5mm, and 2mm in these pictures (in order).



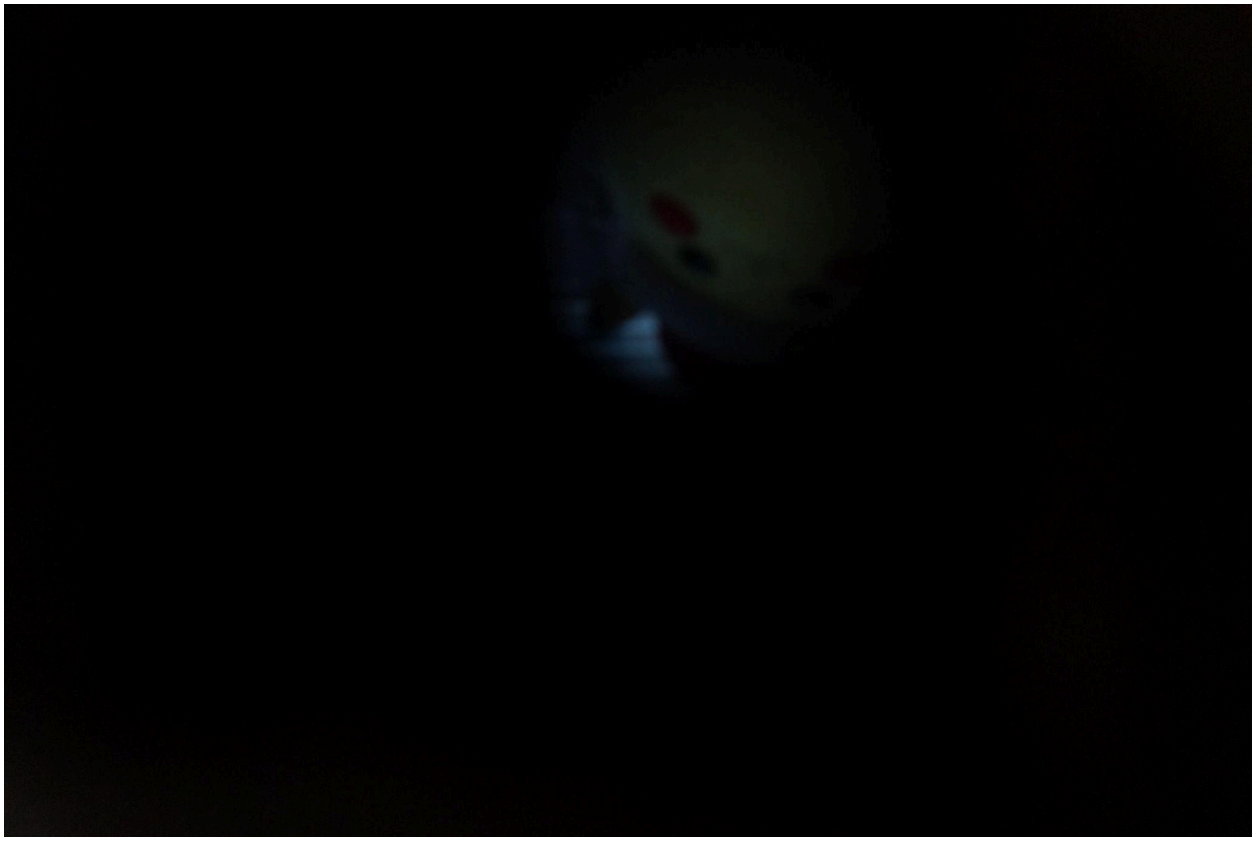


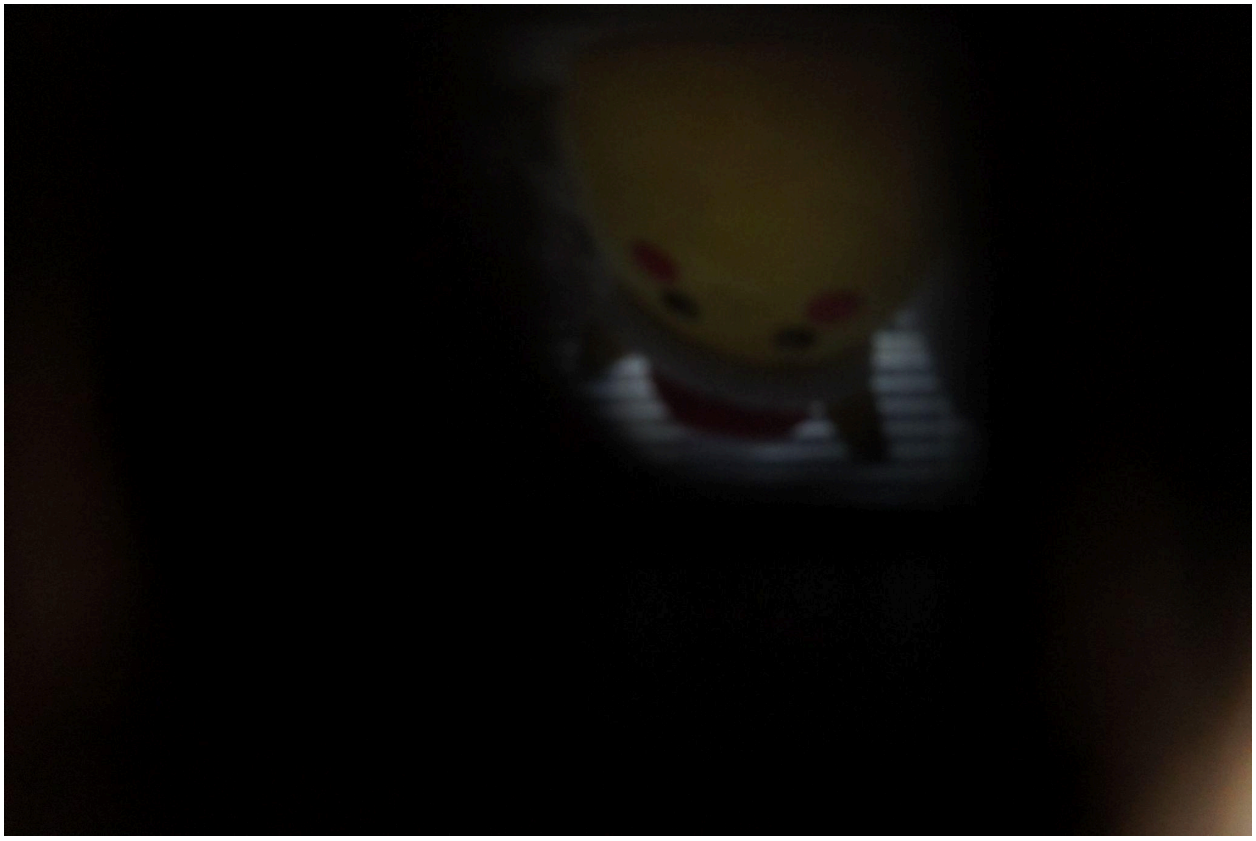


Picture of Pikachu

These pictures were taken inside my apartment. These pictures are of my Pikachu Christmas stuffed animal. I placed Pikachu, as well as the pinhole camera, on my couch with a window in the background and some lights on in the apartment. These were taken with the second pinhole camera that I had made, fixing the issues presented in the first one. The pinhole diameters were 0.1mm, 0.5mm, and 2mm in these pictures (in order).

Here is a picture taken from Target.com for reference:





Picture of Gengar

These pictures were taken inside my apartment. These pictures are of my Gengar stuffed animal. I placed Gengar on my office chair, and the pinhole camera on my bed, again with a window in the background and some lights on in the apartment. These were also taken with the second pinhole camera that I had made. The pinhole diameters were 0.1mm, 0.5mm, and 2mm in these pictures (in order).

Here is a picture taken from Target.com for reference: