AES 561 Homework 4

Mitchell Dodson

March 23, 2023

1 Satellite Image Analysis

1.1 Brightness histograms and equalization

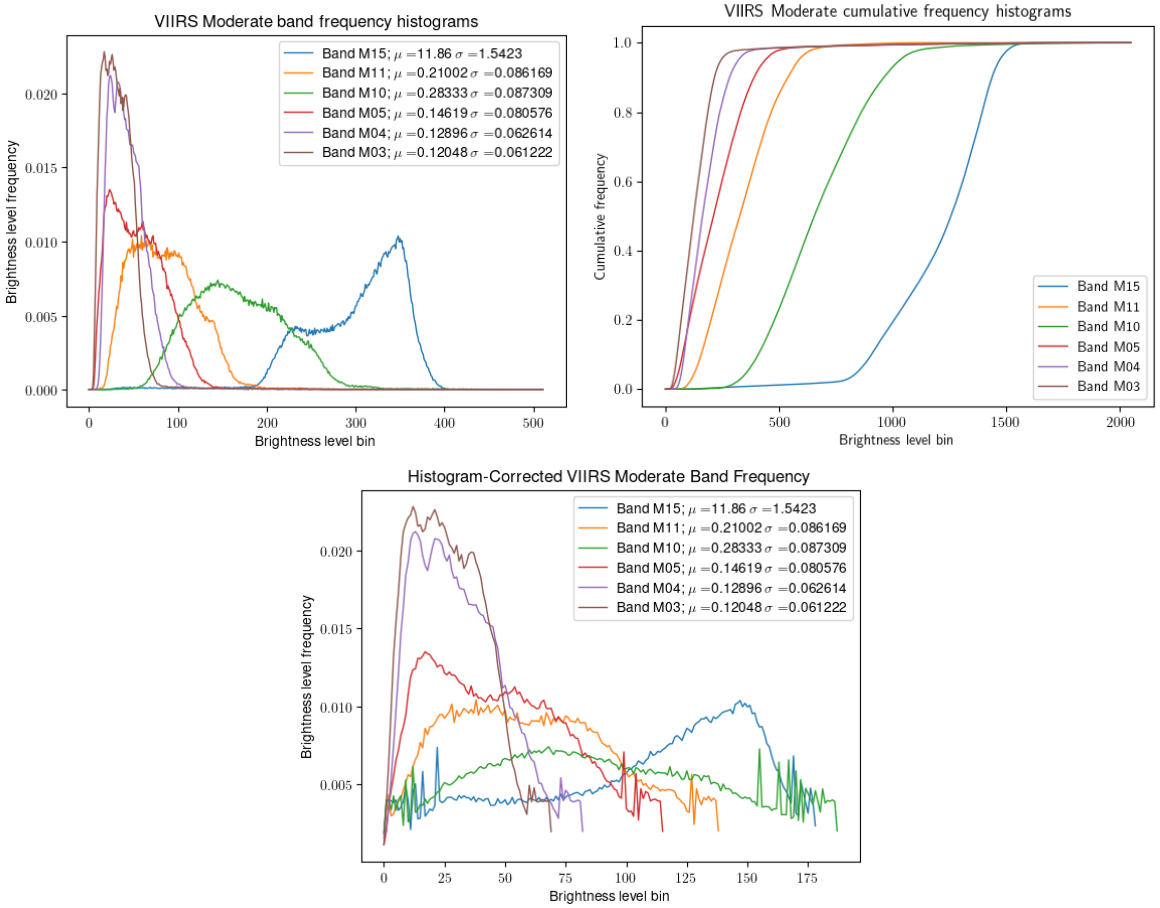

Figure 1: Brightness histograms, cumulative histograms, and discretized histogram-normalized distributions of MODIS bands M15, M11, M10, M05, M04, and M03. Histograms are useful in satellite remote sensing because they characterize the distribution of brightness values for each channel in an image. Subtle features of an image are lost when many brightness values are rounded to the same discrete bin. When enhancement techniques like histogram equalization and matching are applied, these subtle features can be accentuated. Histogram equalization is useful when the probability density of each brightness needs to be as evenly distributed as possible, and histogram equalization is useful for matching the distribution of brightnesses between images, enhancing brightness values in specific subregions, or adapting the brightnesses to a convenient and well-known curve like a gaussian distribution.

1.2 Image enhancement with histogram equalization

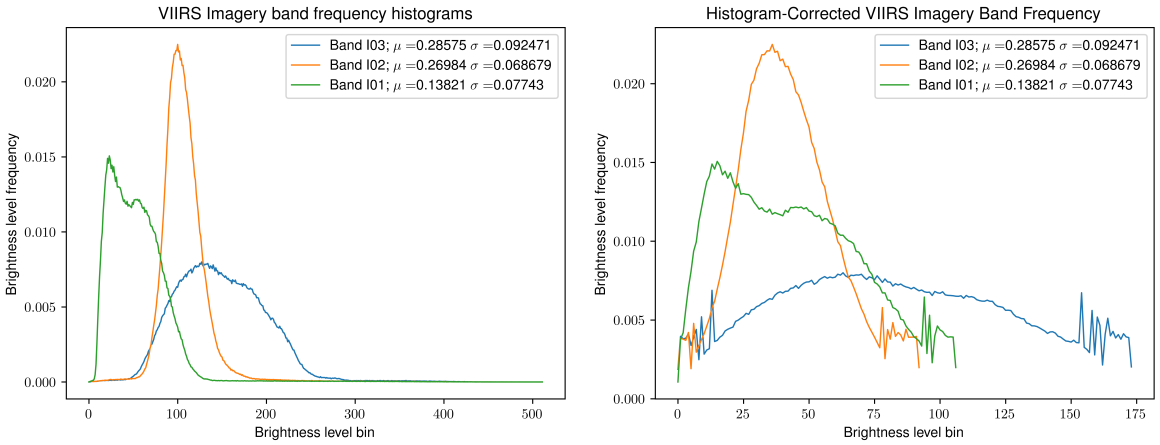

Figure 2: Original and histogram-corrected brightness frequency distribution hisograms for each MODIS imagery band used in the naturalcolor image.

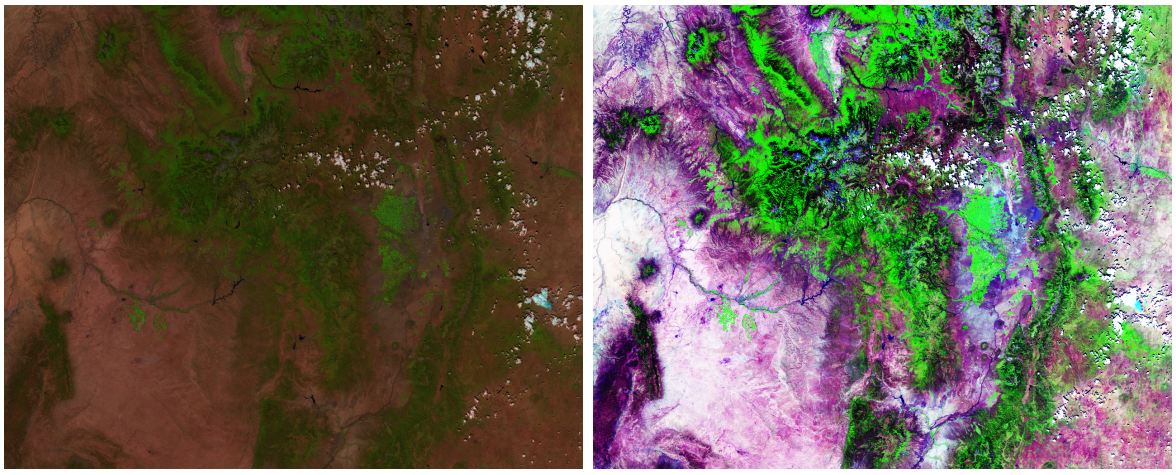

Figure 3: Original MODIS natural color RGB (bands I03, I02, and I01), and histogram-enhanced RGB, both normalized.

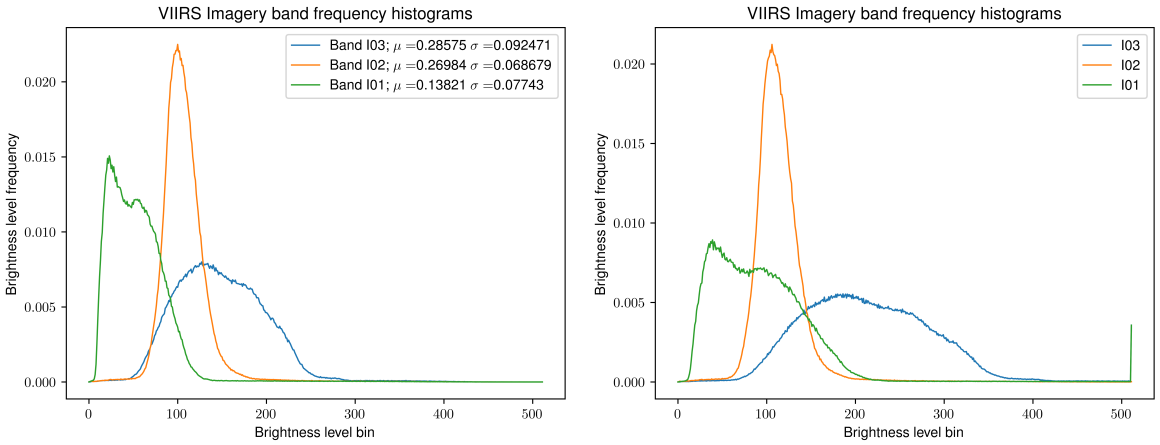

Figure 4: Original and saturating contrast-stretched brightness frequency distribution hisograms for each MODIS imagery band used in the naturalcolor image.

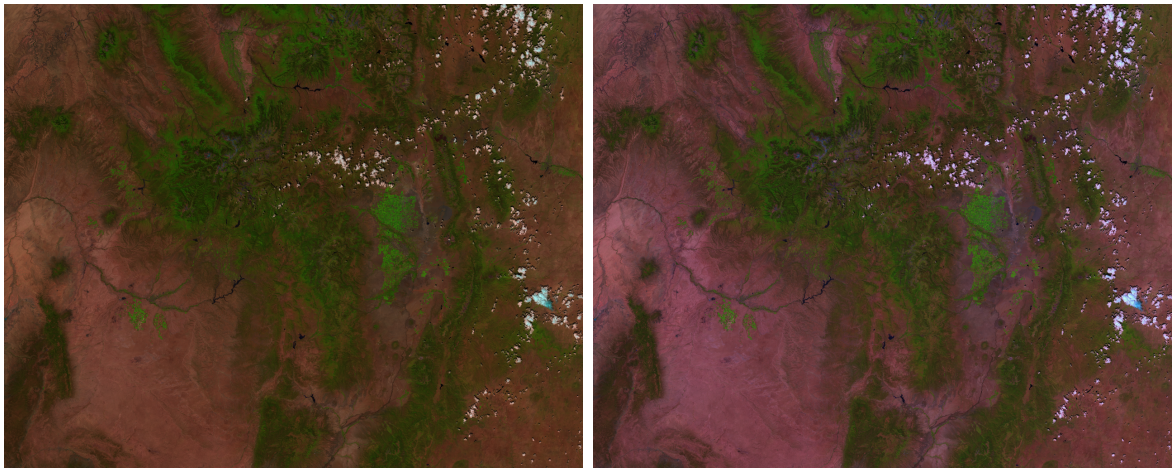

Figure 5: Original natural color RGB and natural color RGB enhanced with the saturating linear contrast method. In my opinion, although the contrast-stretched image is more visually appealing, the histogram-equalized one is much better for identifying pixel classes since subtle features are spread over a greater range of pixel values and aren't rounded to the same brightness level as often.

1.3 Classification candidate selection

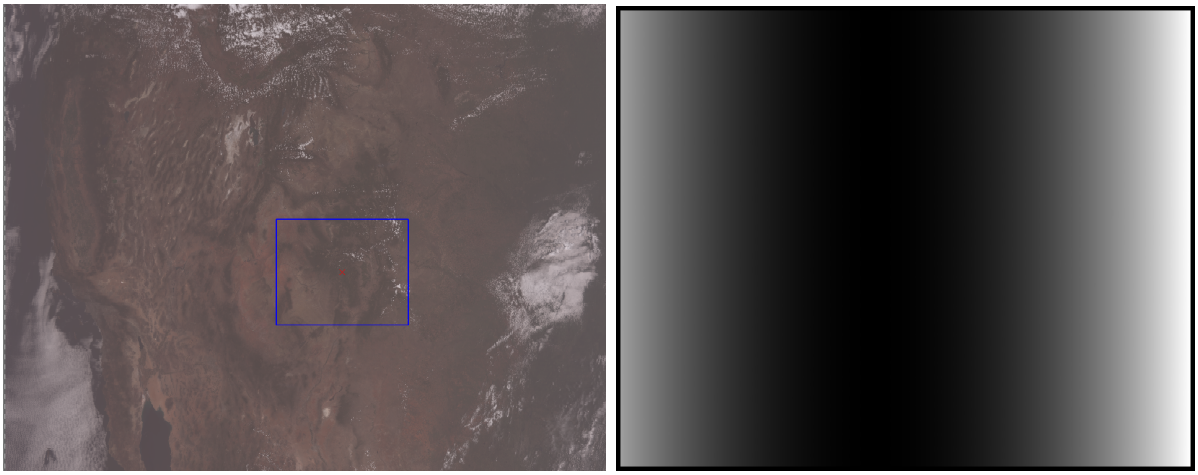

Figure 6: These images give context on the region I selected. The first image indicates my region's extent in the West US with the blue rectangle, and the second shows the pixelwise area distortion in my domain, with white representing greater distortion. The Suomi-NPP MODIS granule I retrieved for my region was captured on July 12, 2016 at 2006z. The center of my image is slightly East of the nadir point, so the East edge is subject to the greatest panoramic distortion. The approximate surface area of my region is $200,691.53\,km^2$.

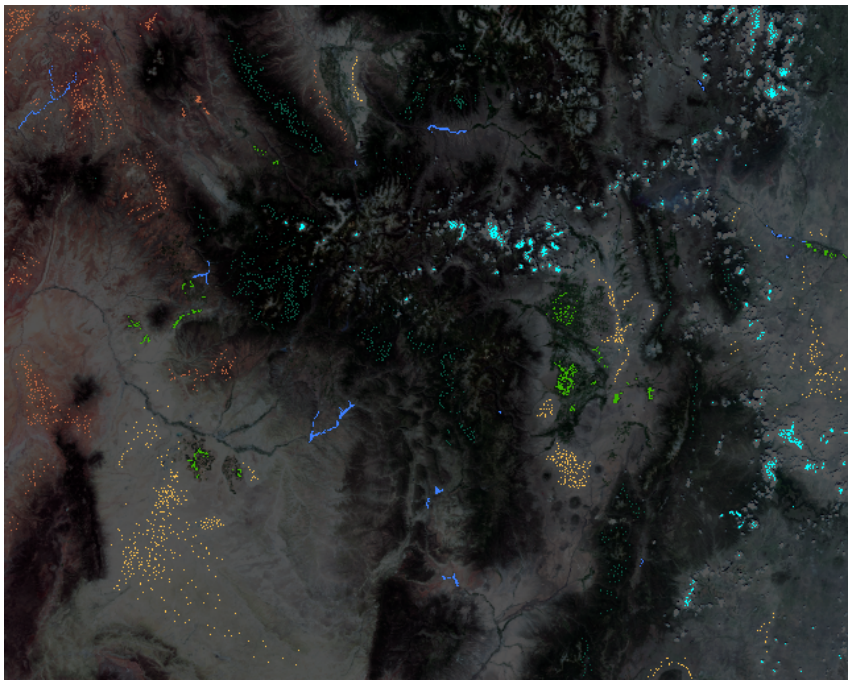

Figure 7: Pixel selections for each class used as training inputs. Cloud samples are shown in teal, crops samples are shown in bright green, mountain vegetation samples are shown in dark green, water samples are shown in blue, grassland samples are shown in yellow, and red sand/clay samples are shown in orange.

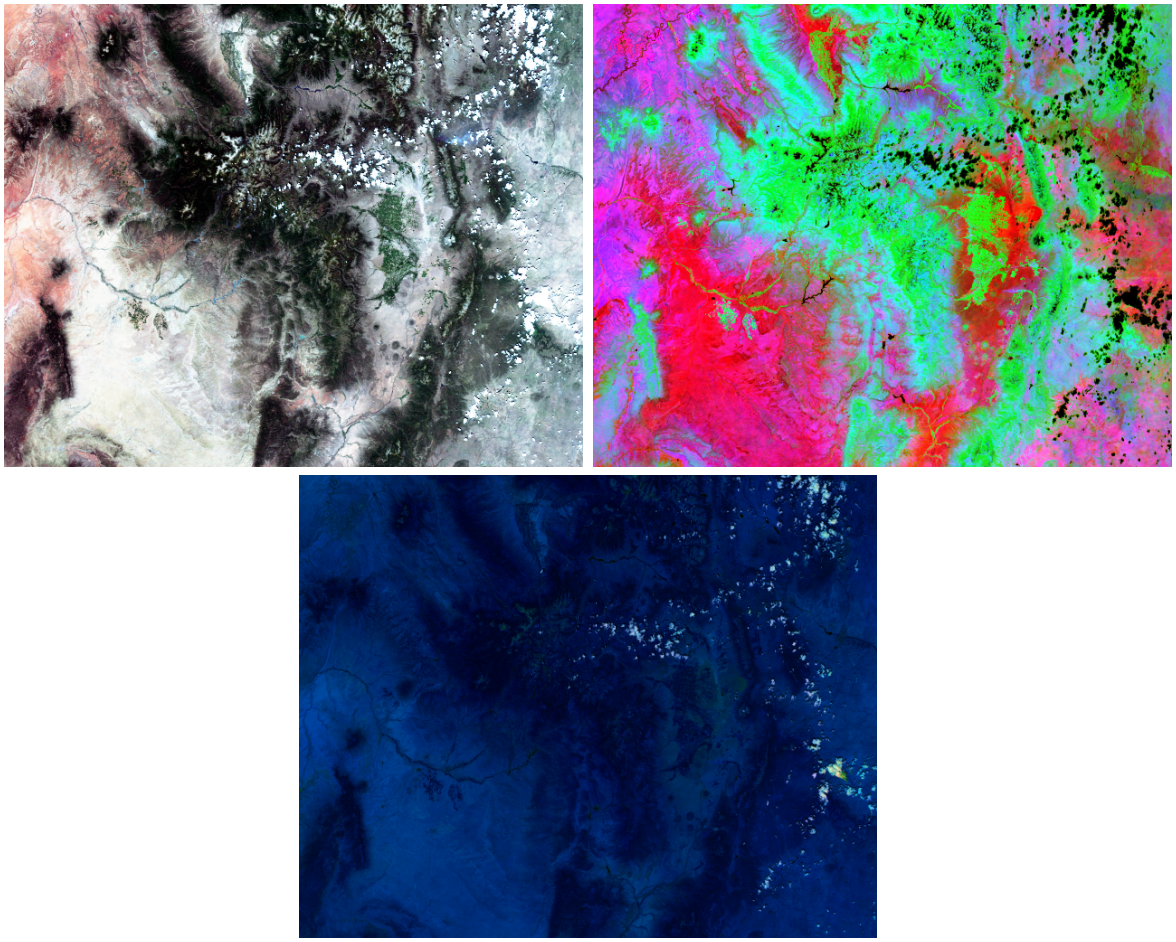

Figure 8: MODIS RGBs used for pixel class selection.

Category	Cloud	Water	Mtn Veg	Clay	Grassland	Crops
Sample Count	874	357	1085	956	974	475

Figure 9: Quantity of unique pixels collected as samples for each category.

Figure 8 displays the RGBs I used to pick samples for pixel classes, the first image is an equalized truecolor image with bands M05, M04, and M03. The second image is a histogram-enhanced custom recipe of band M15, M07-M05 NDVI, and M10-M04 NDSI, which I created in order to easily distinguish clouds and multiple surface types, and the third image is a gamma-enhanced day-cloud phase RGB adaptation of the GOES ABI recipe using bands M09, M05, and M10. The RED band of my custom RGB is M15, which has a peak wavelength near $10.8\mu m$. Band M15 is sensitive to the emissivity properties of surfaces and helps distinguish red clay from arid grassland. The GREEN band is NDVI, which makes vegetation very visible. Last, the BLUE band is NDSI, which separates snow from clouds and vegetation, and incidentally distinguishes exposed and rocky areas from vegetation and grassland better than the truecolor.

In order to choose sample pixels, I developed a GUI tool with OpenCV2 that that allows the user to select and deselect individual pixels for each category, and cycles through provided RGB recipes so that the user can select samples with the RGB that best resolves each category.

	M15		M14		M12		M10		M09	
Category	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3	μ_4	σ_4	μ_5	σ_5
Cloud	5.545	1.539	4.784	1.686	0.913	0.173	0.552	0.151	0.215	0.098
Water	10.220	1.352	9.697	1.429	0.803	0.268	0.147	0.078	0.001	0.001
Mtn Veg	9.683	0.510	9.242	0.584	0.519	0.098	0.155	0.026	0.003	0.001
Clay	13.186	0.415	12.698	0.540	1.479	0.150	0.358	0.063	0.003	0.001
Grassland	13.389	0.511	13.171	0.502	1.500	0.156	0.372	0.061	0.003	0.002
Crops	10.575	0.685	10.188	0.769	0.719	0.161	0.220	0.032	0.002	0.001

	M07		M05		M04		M03	
Category	μ_6	σ_6	μ_7	σ_7	μ_8	σ_8	μ_9	σ_9
Cloud	0.676	0.215	0.611	0.227	0.581	0.215	0.590	0.220
Water	0.141	0.065	0.118	0.041	0.117	0.024	0.118	0.019
Mtn Veg	0.269	0.046	0.049	0.010	0.071	0.008	0.068	0.007
Clay	0.291	0.052	0.204	0.045	0.128	0.018	0.117	0.012
Grassland	0.301	0.031	0.231	0.038	0.186	0.027	0.164	0.019
Crops	0.411	0.065	0.091	0.024	0.107	0.012	0.094	0.012

Figure 10: Means and standard deviations of pixels selected as samples for each surface category.

1.4 Minimum-distance classification

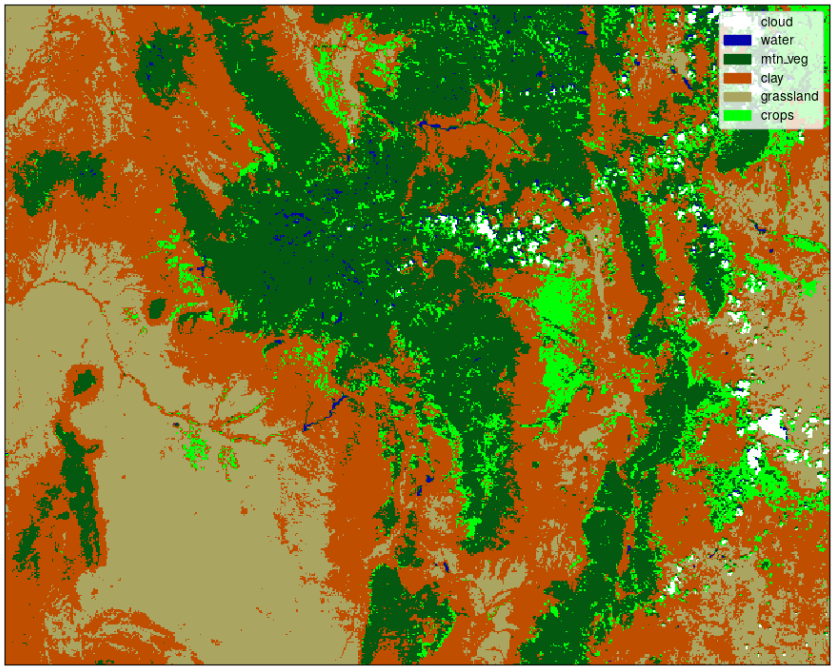

Figure 11: Minimum-distance classification of my region using all bands listed in Fig 10. To perform minimum-distance classification on my image, I extracted and averaged the sample pixels for each class, applied the discriminant function $g_i(\vec{x}) = 2\vec{m}_i \cdot \vec{x} - \vec{m}_i \cdot \vec{m}_i$ to each pixel $\vec{x}$ and class mean $\vec{m}_i$, and assigned each pixel to the class minimizing $g_i(\vec{x})$.

		M15		M14		M12		M10	
Category	Area (km^2)	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3	μ_4	σ_4
Cloud	2,797	5.488	1.516	4.722	1.661	0.911	0.169	0.552	0.148
Water	19,109	10.164	1.359	9.637	1.437	0.795	0.266	0.144	0.077
Mtn Veg	29,993	9.724	0.537	9.291	0.619	0.527	0.104	0.156	0.027
Clay	81,778	13.189	0.429	12.706	0.550	1.478	0.153	0.354	0.063
Grassland	31,163	13.369	0.517	13.154	0.514	1.490	0.159	0.372	0.059
Crops	35,850	10.560	0.675	10.172	0.758	0.715	0.159	0.218	0.032

	M09		M07		M05		M04		M03	
Category	μ_5	σ_5	μ_6	σ_6	μ_7	σ_7	μ_8	σ_8	μ_9	σ_9
Cloud	0.218	0.098	0.411	0.065	0.091	0.023	0.107	0.012	0.094	0.012
Water	0.001	0.001	0.299	0.031	0.228	0.040	0.184	0.028	0.162	0.019
Mtn veg	0.003	0.001	0.289	0.051	0.203	0.044	0.127	0.018	0.116	0.012
Clay	0.003	0.001	0.269	0.044	0.050	0.011	0.071	0.008	0.069	0.007
Grassland	0.003	0.002	0.139	0.065	0.117	0.041	0.116	0.024	0.118	0.019
Crops	0.002	0.001	0.682	0.214	0.619	0.226	0.588	0.214	0.598	0.219

Figure 12: Area coverage and statistics for each category and input band.

1.5 Maximum-likelihood classification

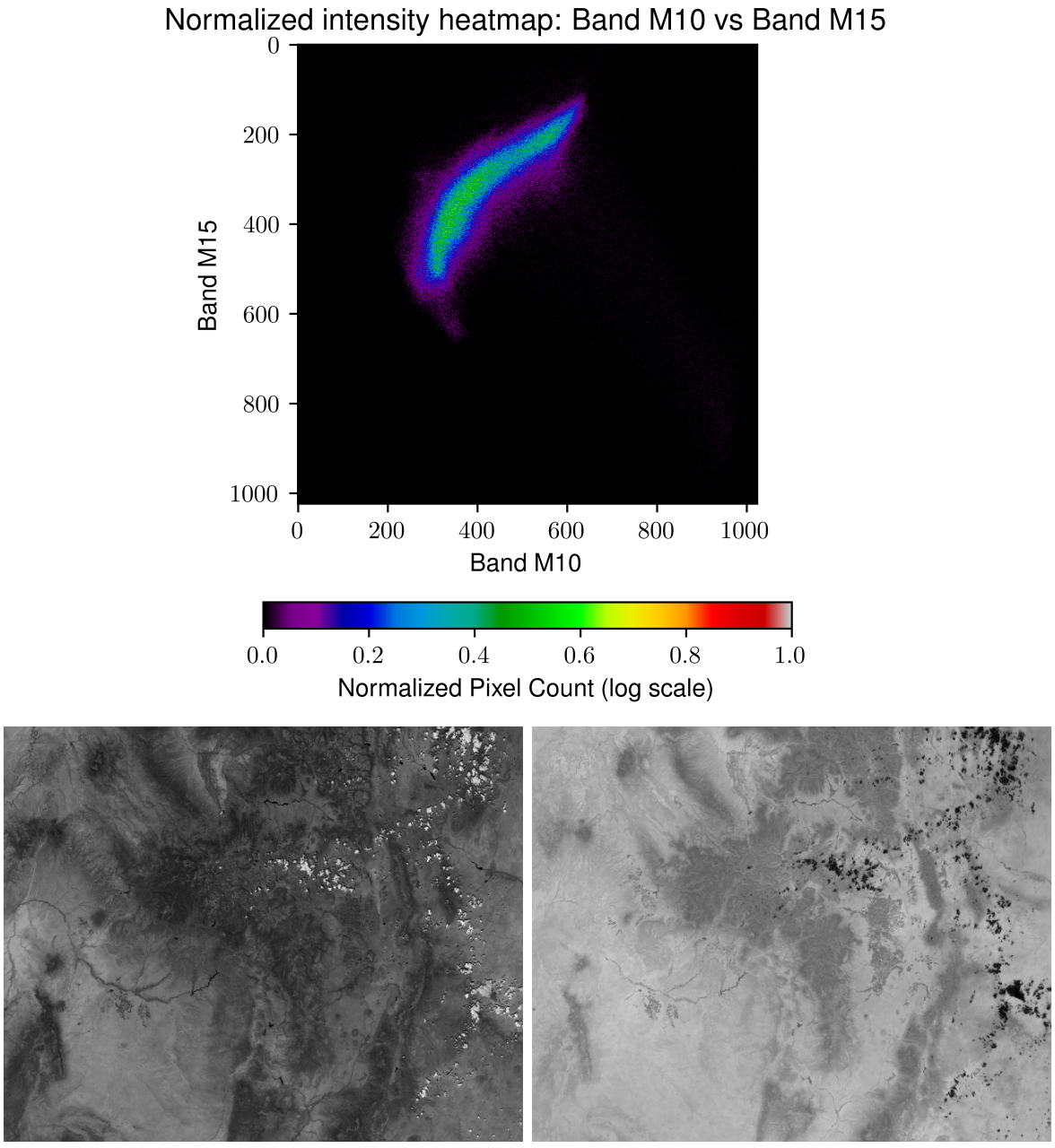

Figure 13: Intensity heat map of bands M10 and M15, and greyscale images of band M10 and M15, respectively. Band M10 peaks at $1.61\mu m$, and band M15 peaks at $10.76\mu m$. I chose these bands because the nonlinear heatmap suggests that the similarities of some surface types are easier to isolate when the surfaces' properties differ between bands.

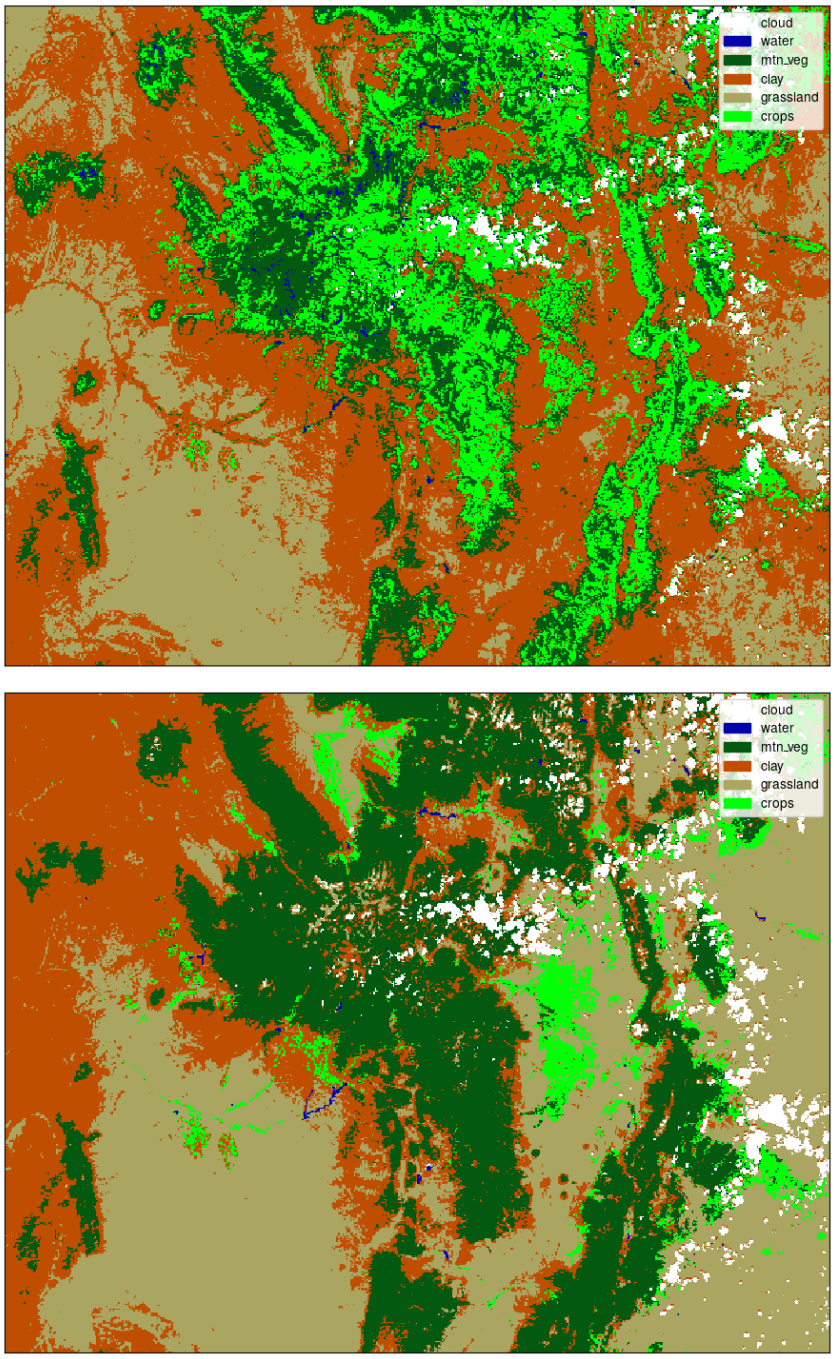

Figure 14: Maximum-likelihood classification of my region. The first image shows classes assigned using only bands M10 and M15, and the second image shows classes assigned using all 9 bands I selected. The 2-band version has high skill in distinguishing clouds and vegetation from other surface types, but misclassified snow on the mountain peaks as water, and misclassified much of the unvegetated ground. The all-band version had much higher skill, and closely emulated the surfaces I selected for my training samples.

		M10		M15	
Category	Area (km^2)	μ_1	σ_1	μ_2	σ_2
Cloud	4,860	0.468	0.139	6.783	1.569
Water	818	0.103	0.023	9.278	0.389
Mtn Veg	34,785	0.183	0.037	10.509	0.917
Clay	83,785	0.288	0.039	12.586	0.683
Grassland	43,843	0.391	0.037	13.233	0.447
Crops	32,601	0.217	0.025	10.460	0.532

Figure 15: Bands M10 and M15 maximum-likelihood mean and standard deviation for each category, and each category's area coverage. Area coverage was calculated by taking the sum of the area of each pixel for each class based on the area distortion map shown in Figure 6.

		M15		M14		M12		M10	
Category	Area (km^2)	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3	μ_4	σ_4
Cloud	8,397	8.204	2.130	7.705	2.334	0.896	0.191	0.378	0.158
Water	246	9.533	0.907	8.986	0.999	0.669	0.200	0.109	0.053
Mtn Veg	52,797	10.512	0.831	10.195	0.961	0.699	0.169	0.195	0.037
Clay	52,412	12.445	0.842	12.144	0.879	1.208	0.243	0.290	0.068
Grassland	77,052	12.897	0.816	12.707	0.899	1.331	0.216	0.339	0.056
Crops	9,785	11.194	0.698	10.892	0.806	0.848	0.163	0.234	0.032

	M09		M07		M05		M04		M03	
Category	μ_5	σ_5	μ_6	σ_6	μ_7	σ_7	μ_8	σ_8	μ_9	σ_9
Cloud	0.096	0.087	0.405	0.194	0.310	0.206	0.296	0.193	0.299	0.198
Water	0.001	0.000	0.108	0.046	0.096	0.031	0.107	0.027	0.111	0.022
Mtn Veg	0.004	0.002	0.256	0.041	0.070	0.017	0.083	0.011	0.079	0.010
Clay	0.003	0.001	0.251	0.047	0.151	0.051	0.117	0.021	0.110	0.016
Grassland	0.003	0.002	0.276	0.031	0.186	0.042	0.155	0.027	0.141	0.019
Crops	0.002	0.001	0.317	0.061	0.101	0.021	0.108	0.011	0.100	0.010

Figure 16: All-band maximum-likelihood mean and standard deviation of each band for each category, and each category's area coverage.

1.6 K-means classification

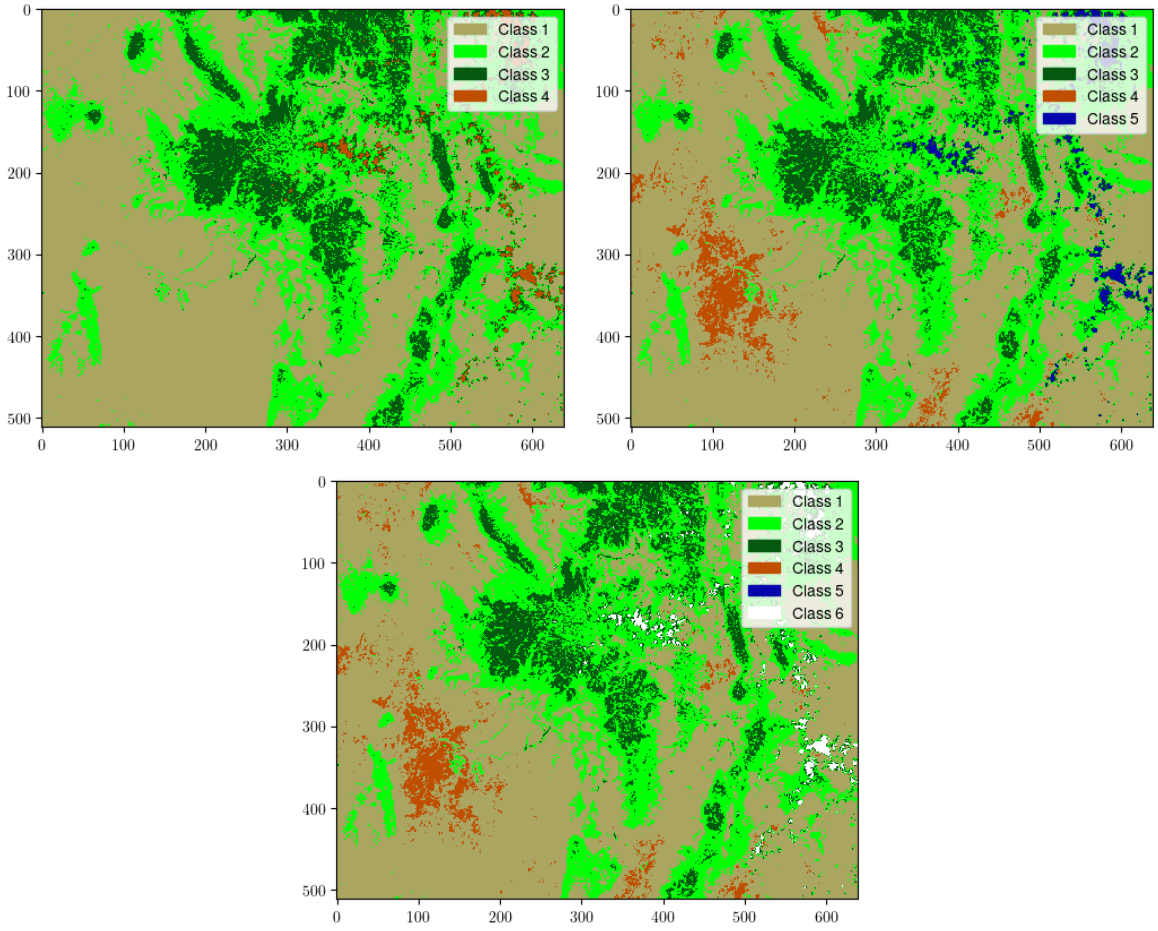

Figure 17: K-means classification using MODIS bands M10 and M15. Images are initialized with 4, 5, and 6 categories, respectively. As shown in Figure 13, MODIS bands M10 and M15 have a nonlinear correlation that makes clouds easy to distinguish, and visibly differentiates vegetated surfaces from arid ground. These affinities are reflected in the K-means classification results, which starkly separate clouds, vegetation, and grasslands/red clay. My K-means algorithm randomly-selects pixels as centroids, then uses euclidean distance to assign each pixel to the closest cluster. Once pixels are collected, the algorithm calculates the new mean for each cluster and repeats at the euclidean distance operation until all the means are equal up to 3 decimal accuracy. I found that the randomly-initialized centroids would sometimes fail to find any nearby pixels, so I implemented a condition that re-initializes a centroid up to 5 times if its cluster has no members. After 5 failures, the cluster is dropped. My results show that 4 classes is too few since no unvegetated surface texture is identified, and 6 classes is too many since no additional surface types were found. I also noticed that K-means classified much of the lower-elevation vegetated surfaces separately from the higher-elevation regions. After looking at locations near the border of the 2 classes with Google street view, I realized that they finely trace the altitude-driven transition from forests containing mostly ponderosa pines and aspen trees to forests dominated by spruce-fir, which is corroborated by Colorado State Forest Service.
HTTPS://CSFS.COLOSTATE.EDU/COLORADO-FORESTS/FOREST-TYPES/

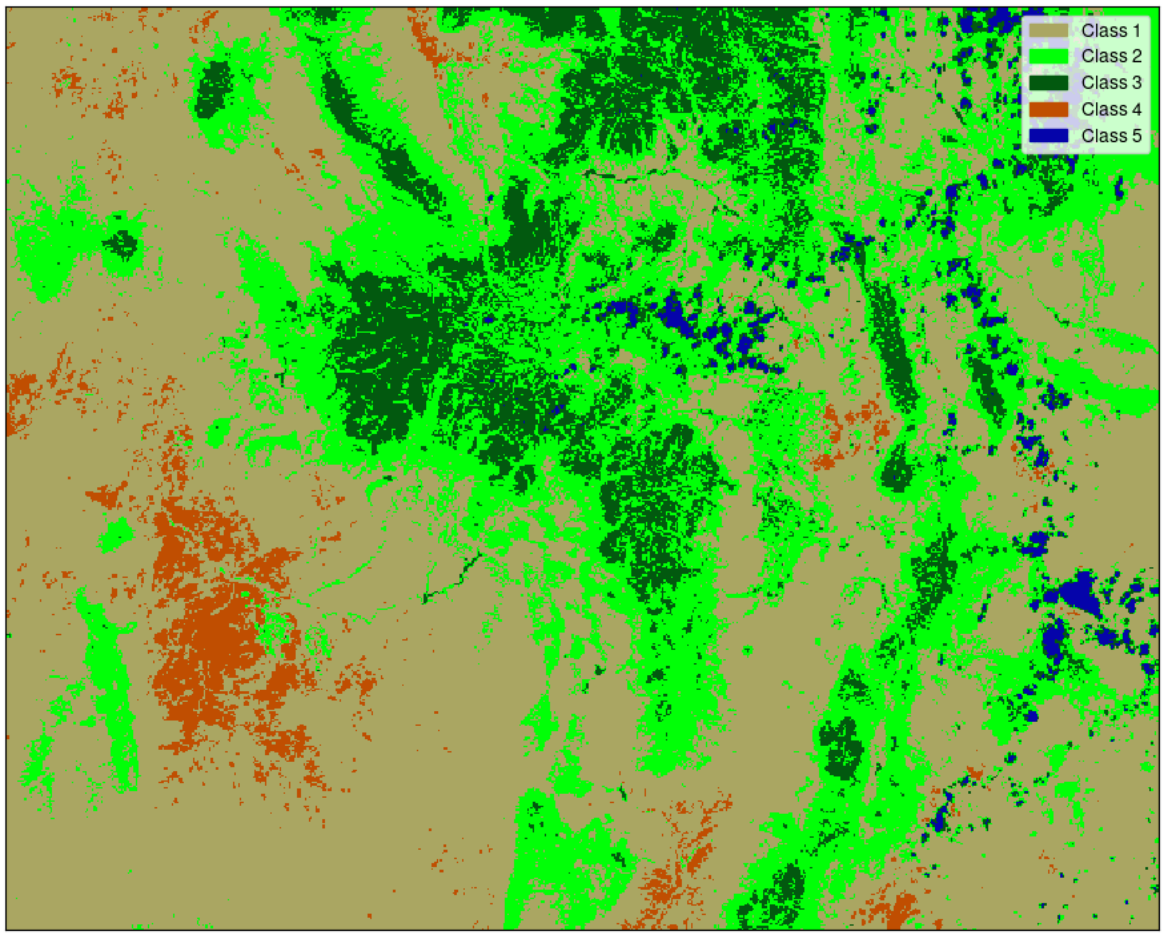

Figure 18: 5-category k-means classification with bands M10 and M15.

Class	Count	M10 μ	M10 σ	M15 μ	M15 σ
Class 1	181779	0.267	0.039	1.306	0.174
Class 2	95648	0.261	0.048	0.799	0.134
Class 3	34619	0.268	0.067	0.565	0.143
Class 4	11051	0.289	0.033	1.651	0.067
Class 5	4583	0.612	0.172	0.933	0.134

Figure 19: Pixel counts, mean, and standard deviation of each pixel class identified by K-means classification with bands M10 and M15.

1.7 Principle component analysis

$$\begin{pmatrix} 0.0065 & 0.0036 & 0.0245 \\ 0.0036 & 0.0041 & -0.0259 \\ 0.0245 & -0.0259 & 2.3843 \end{pmatrix}$$

	λ	$\vec{v}_1$	$\vec{v}_2$	$\vec{v}_3$
$PC1$	2.3848	−0.0103	−0.8063	0.5915
$PC2$	0.0091	0.0109	−0.5915	−0.8062
$PC3$	0.0010	−0.9999	0.0019	−0.0148

Figure 20: Covariance matrix, eigenvalues (λ), and normalized eigenvector components ($\vec{v}$) used for principle component analysis of my region.

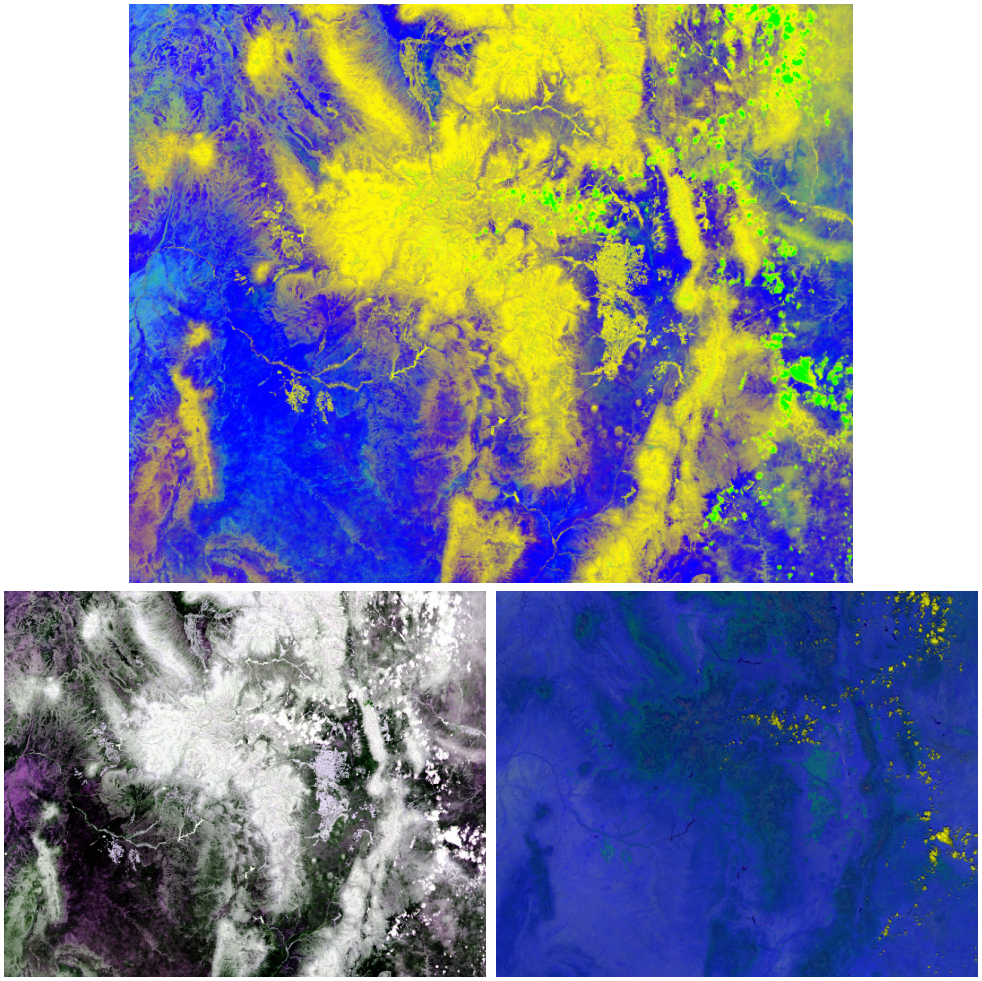

Figure 21: Principle component analysis results. The fist image was generated using only bands M05, M07, and M15, and the second shows 3 of the 9 components found when all 9 bands were used for PCA. The third image is a normalized RGB of bands M05, M07, and M15. Compared to the original composite, the 3-band image is good at resolving clouds and multiple surface types, but still doesn't easily isolate water or vegetation types, and the all-band components I chose are skillful at distinguishing vegetation types and surface texture, but fails to clearly distinguish clouds from mountain vegetation. Judging by the color distribution of the 3-band image, the red component appears to scale positively with highly vegetated surfaces and negatively with clouds, the green band correlates with cloud cover and vegetation, and the blue band correlates with bare ground.

1.8 Edge detection

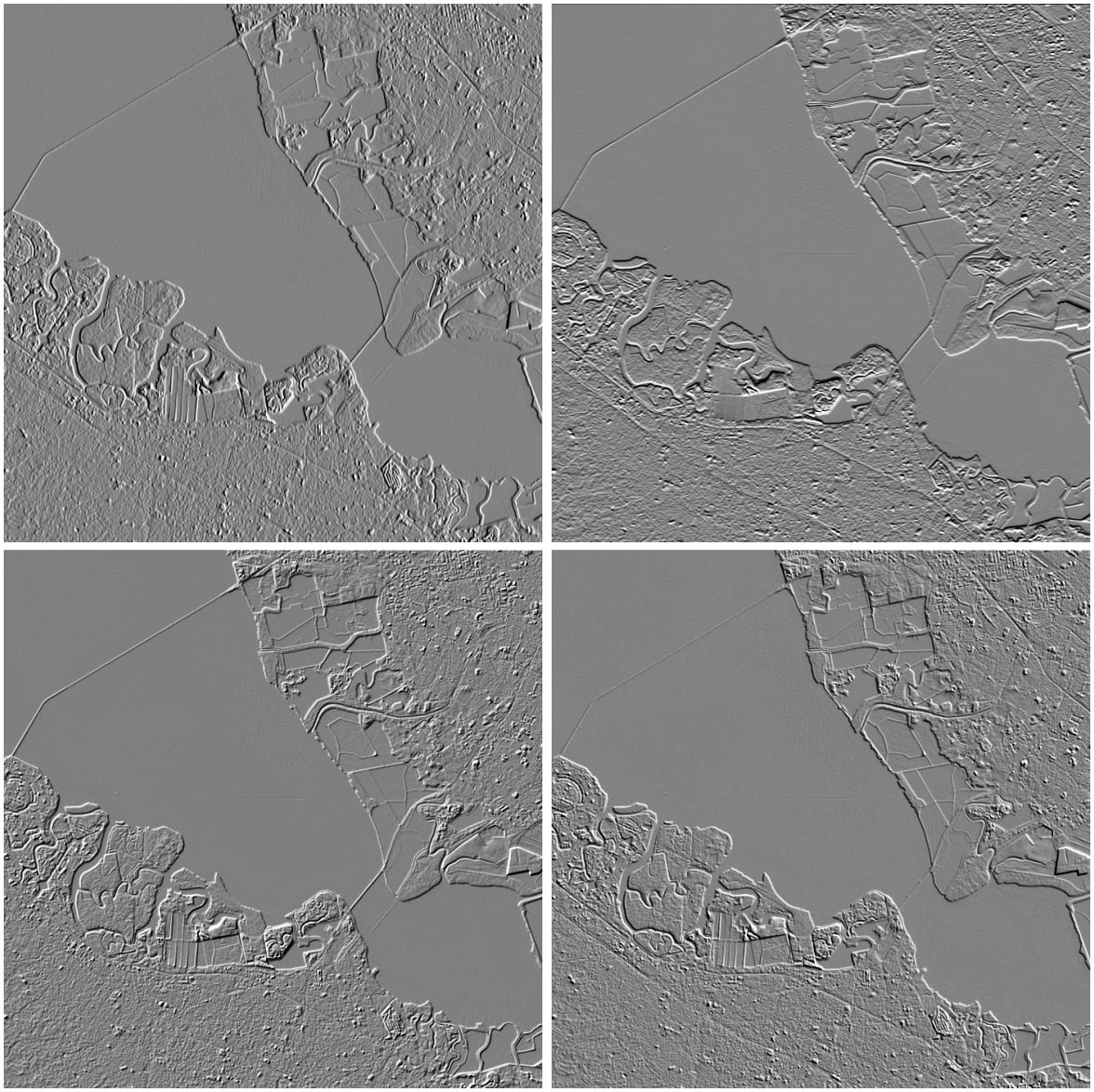

Figure 22: ASTER image after application of vertical, horizontal, forward diagonal, and back diagonal filters, respectively. Fine structures are much easier to identify when they run parallel to an edge filter's kernel. For example, the vertical lines of the harbor are easier to resolve in the vertically-filter image, and the San Mateo-Hayward Bridge is easiest to identify in the images filtered with horizontal and forward-diagonal kernels.

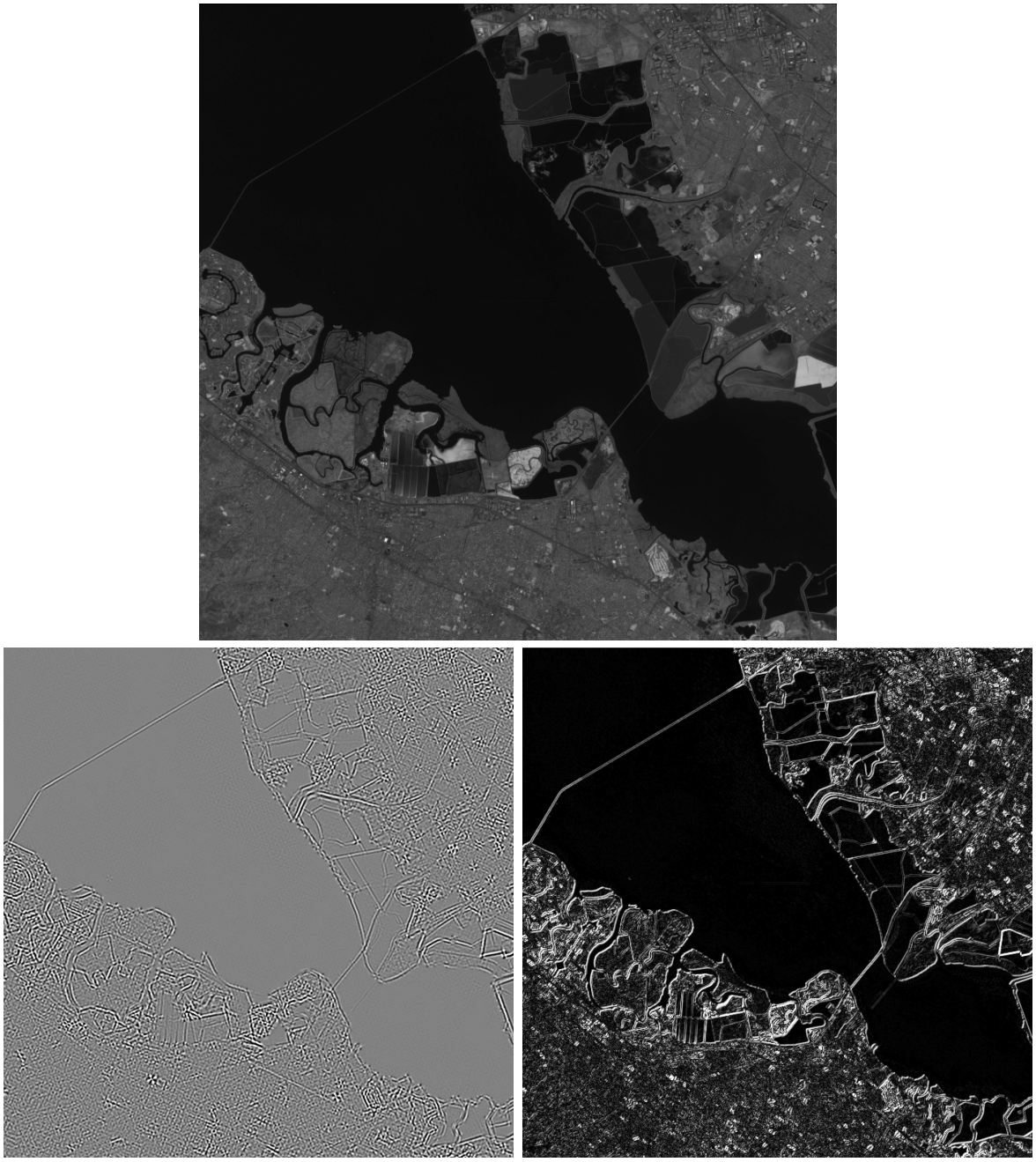

Figure 23: Original ASTER image, edge detection using consecutive application of filters, histogram-equalized edge detection using each filter applied independently, then combined with euclidean distance in the same manner as the Sobel operator. Both edge detection methods appear to be effective since the edges of structures are much easier to resolve than in the original image, but even after histogram equalization the image generated with sequentially-applied filters appears noisier and more difficult to resolve than the euclidean distance scheme.

1.9 Roberts and Sobel operators

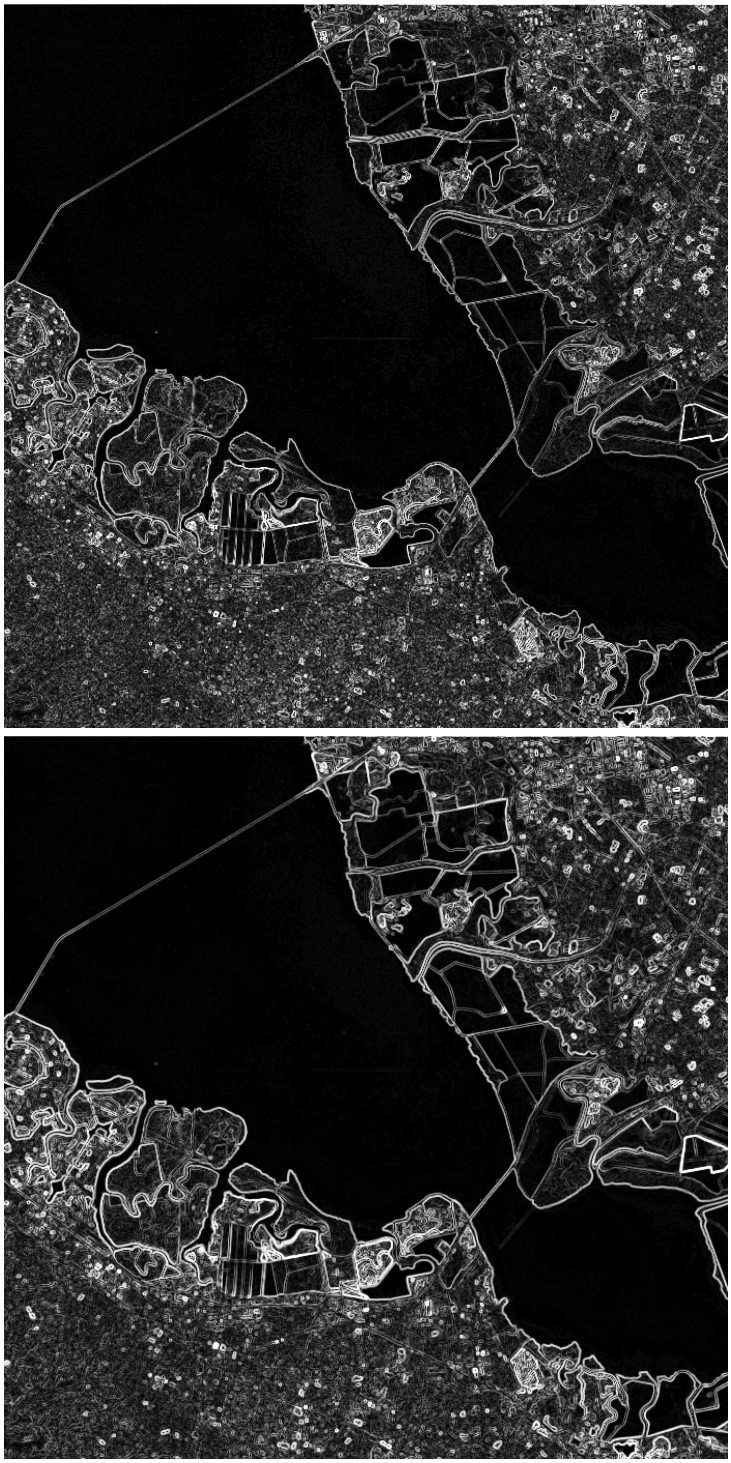

Figure 24: ASTER image after applications of the Roberts and the Sobel operators, respectively. The results for each method are similar overall, but the Sobel operator resolves edges more finely, and doesn't generate as much noise in areas with features that are too small to be reasonably considered edges.

1.10 Landsat MSS cloud and water-masked NDVI

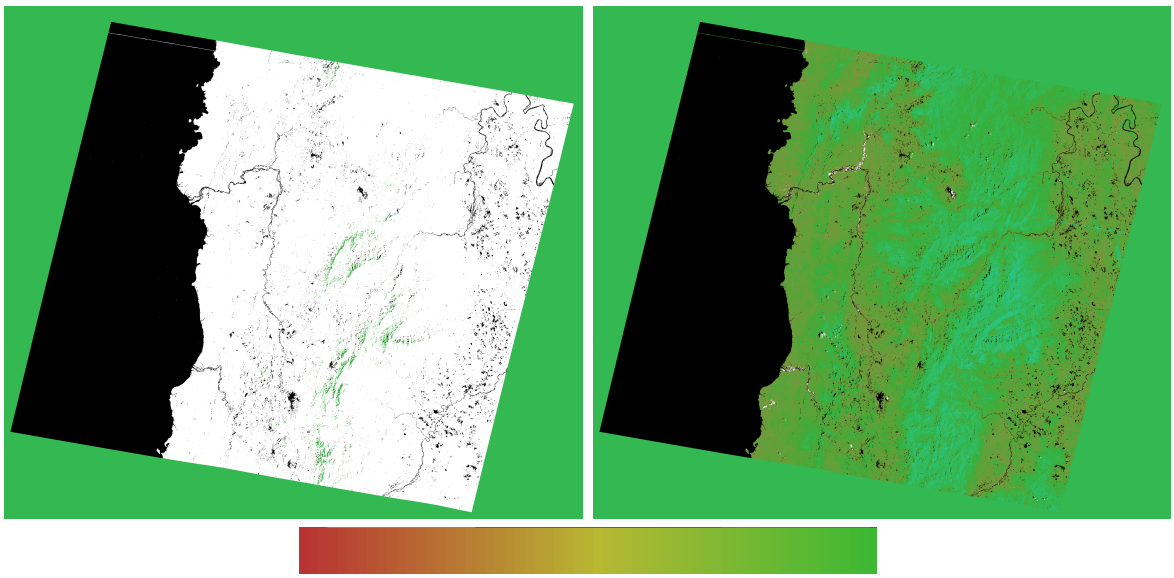

Figure 25: Normalized NDVIs with clouds masked as white, and water masked as black. In both images, water is masked using the NDWI recipe (GREEN-NIR)/(GREEN+NIR) with Landsat MSS bands 1 ($0.5-0.6\mu m$) and 3 ($0.7-0.8\mu m$). The first image applies the Saunders & Kriebel method as closely as possible, substituting AVHRR band 1 ($.62\mu m$) for Landsat band 1, and AVHRR band 2 ($.8\mu m$) for Landsat band 4 ($0.8-1.1\mu m$). The second image is an adaptation of the S&K method using custom thresholds. The colorbar at the bottom is a rendering of the 256 brightness bins used in generating the NDVI RGB.

Maximum	Minimum	Mean	Std. Dev.
0.7125	0.0570	0.5041	0.0079

Figure 26: Statistics for masked NDVI values.

Figure 25 displays my results as an HSV-mapped RGB with gamma enhancement, and Figure 26 shows statistics on the NDVI values that weren't masked as clouds or water. The granule I used was captured by Landsat 3 over the Banaue Rice Terraces on Luzon island in the Phillipines on March 11, 1979. The rice paddy terraces are positioned East of the Lagban river, in the region with the highest NDVI values. In order to mask water in my image, I simply set a lower-bound threshold of .39 on the NDWI recipe specified in Figure 25. Since the MSS instrument on Landsat 3 isn't equipped with any thermal IR bands (except for the one that failed soon after launch), I only performed tests 3 and 4 of the Saunders & Kriebel method. Tests 3 and 4 stipulate that a cloud candidate must have a RED band reflectance greater than 15%, and a NIR to RED band ratio less than 1.6. These thresholds weren't restrictive enough to be effective on my image, probably due to the additional atmospheric absorption in the NIR region covered by MSS band 4. When I increased test 1's threshold to a 38% lower bound for the RED band and decreased test 2's NIR/RED ratio threshold to a .9 lower bound, the test successfully masked the few cumulus clouds in my image, as well as some valley fog on the Lagben river.

1.11 Amazon fire detection

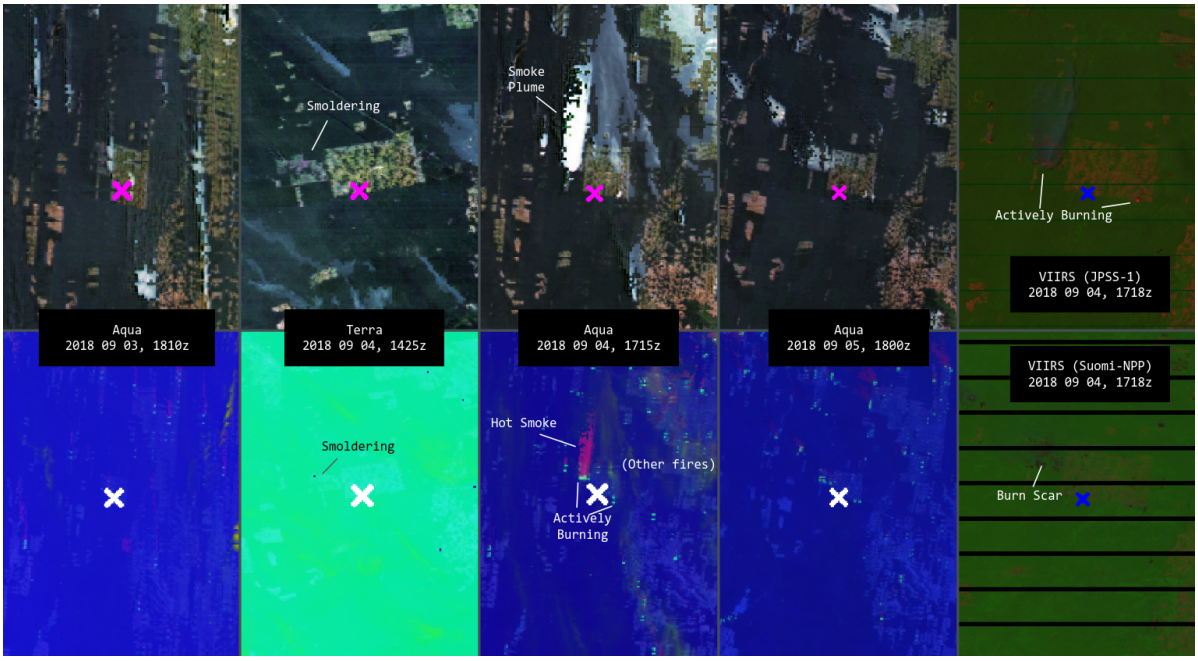

Figure 27: Mosaic of the progression of the field fire I selected, which burned September 3-5, 2018. The x marks the pixel closest to $60°W$, $10°S$. I specifically focus on the image captured by Aqua on September 4 at 1715z. This granule was acquired only 3 minutes prior to the JPSS-1 VIIRS granule shown in the top right.

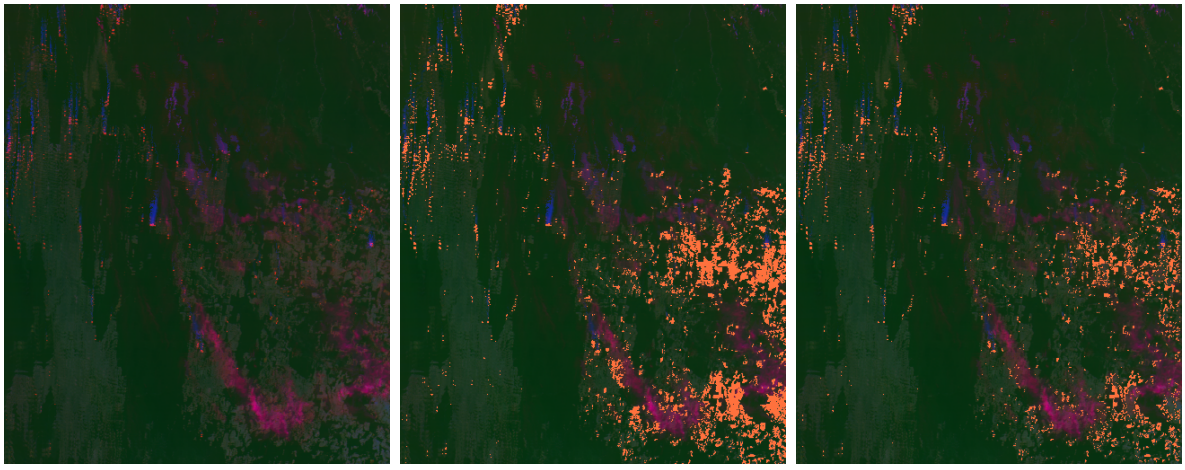

Figure 28: My custom-enhanced fire RGB, pixels identified as fire candidates by the Flasse & Ceccato (F&C) method, and fires confirmed by F&C. 5,982 pixels were positively identified as fires with the standard thresholds. The RGB in the first image is a gamma-stretched composite with RED=SWIR-LWIR, GREEN=LWIR, and BLUE=VIS, where the visible channel is the average reflectance in the $.459 - .670\mu m$ range (see Figure 29 caption). These are the same channels I used as inputs for the F&C procedure.

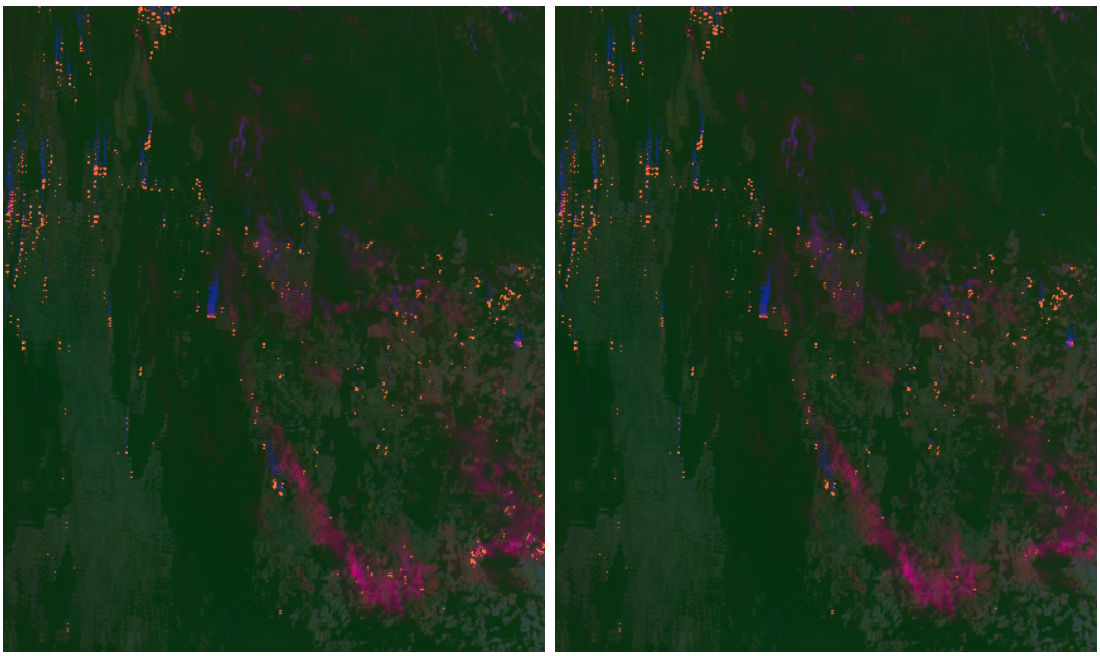

Figure 29: Pixels identified as fire candidates using custom thresholds, and subsequent pixels identified as fires with the F&C confirmation methodology. When selecting pixel candidates, I decreased the SWIR lower bound from $314K$ to $311K$, decreased the SWIR-LWIR difference threshold to $8K$, decreased the upper bound of NIR reflectivity to 10%, and set a new lower bound of $280K$ on the LWIR threshold. I chose these values to better approximate the locations of the relatively isolated red pixels in my fire RGB, as the standard F&C method demonstrated in Figure 30 seemed to classify fires far too generously. The only significant differences between the pixel candidates my threshold selected and the pixels confirmed to be fires are that some groupings of candidate pixels over the low-lying clouds in the bottom right were disqualified as fires by the F&C algorithm. I believe these changes were necessary because I substituted AVHRR band 2 ($.725-1.0\mu m$) for the average reflectance of MODIS bands 4, 3, and 1 ($.459-.670\mu m$ in total) since no NIR bands were available in the L1b granule I used. The $.8-1\mu m$ edge of the blackbody radiance curve for a $1000K$ fire is small, but much greater than in the visible range, and NIR wavelengths are attenuated less by Rayleigh scattering than in the visible spectrum. Either or both of these factors may have contributed to decreasing observed radiance in the MODIS visible bands compared to AVHRR band 2, which warrants decreasing the reflectivity threshold. Ultimately, 928 pixels were positively identified as fires using my custom thresholds.

1.12 FFT noise reduction

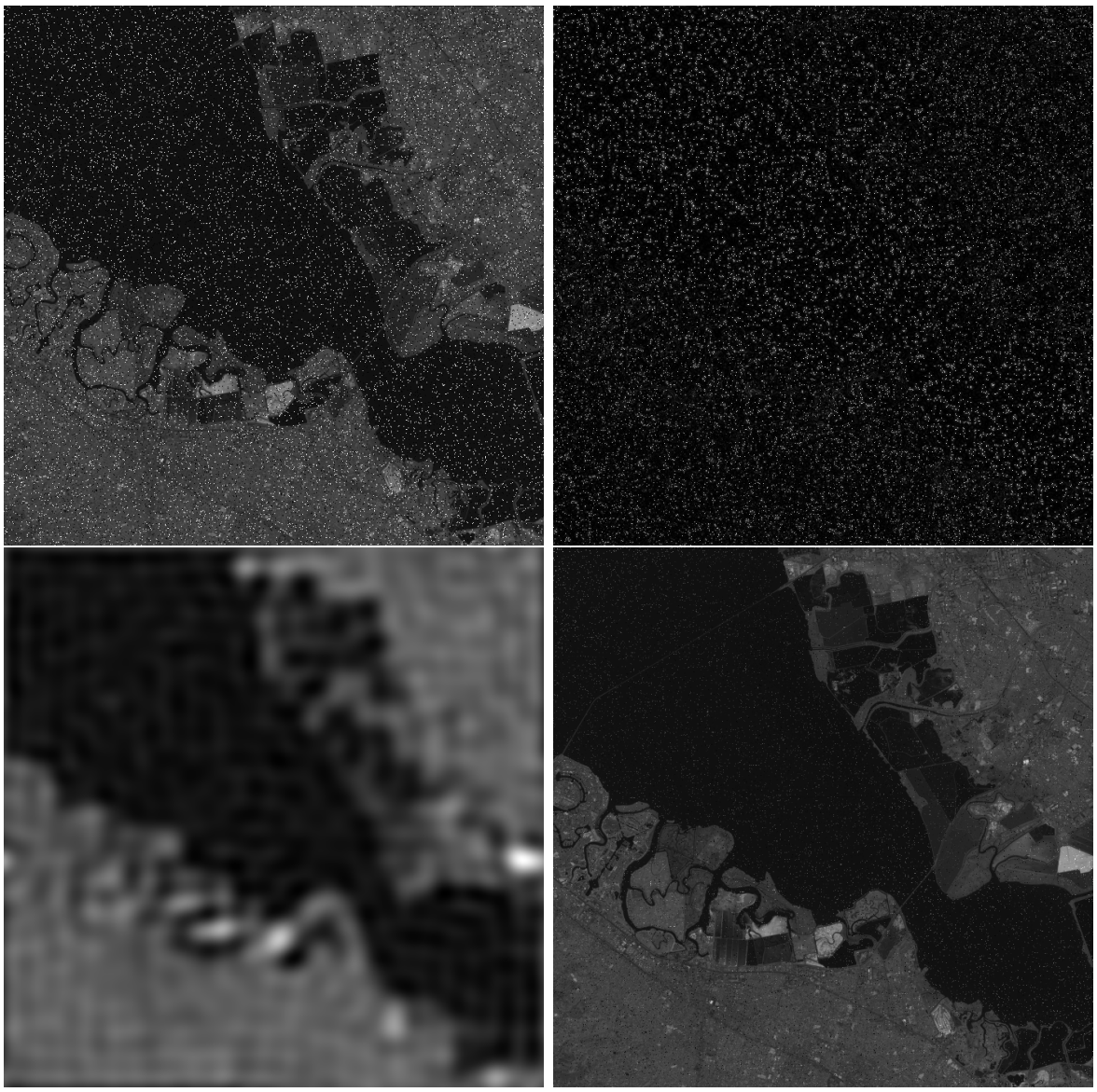

Figure 30: Original noisy image, noisy image with a radius filter (F1) removing middle-frequency values, noisy image with a radius filter removing highest and lowest-frequency values (F2), and a denoised reconstruction. I manually selected the phase space regions to remove for each filter with an OpenCV2-based GUI tool that I built. After the filters were applied, I generated the reconstruction by normalizing the F1 brightness array to $[0, 1]$, then using restricting F1 to a lower-bound threshold of .078 to identify and mask noise pixels. Finally, I replaced the masked pixels in the noisy image with the pixel values smoothed by F2. The resulting reconstruction still has residual noise since F2 only "smears" the noisy pixels, but the denoised image is visibly smoother.

2 Code comment

One of my major goals for graduate school is to develop code I write for homework and research into a highly generalized and modular python codebase. To that end, I wrote more than 3,000 lines of python for this assignment as a series of modules that can be imported as a python package. All the figures in this document were generated with small scripts importing the high-level methods contained in the modules. My code is available on github with the link below.

https://github.com/Mitchell-D/aes670hw2/

Capabilities of my codebase:

- Search the MODAPS-LAADS DAAC (LWS API) for any available product, and narrow search by illumination, geographic range, and a target time or time range, and download products that match the conditions.
- Process MODIS and VIIRS L1b data and georeference arrays into geographically-specified subgrids of a requested set of bands.
- GUI tools for manually selecting sub-regions, picking pixel categories from RGBs, selecting parameter values for arbitrary enhancement functions (linear gamma, contrast stretch, noise filters) using a sliding bar and a live-rendered image, and manually removing specific regions in the phase space of an array with a click-and-drag function.
- All classification and filtering algorithms, with standardized inputs and outputs.
- Methods for procedurally rasterizing markers, boxes, and circles, and text onto an ndarray using geographic or pixel-coordinate reference.
- Destripe and perform bowtie-correction via linear interpolation on arbitrary pixel grids.
- Calculate and print a variety of statistical features for a provided array.