Satellite Image Preprocessing for Enhanced Edge Detection

Joshua Abraham
Lassonde School of Engineering
York University, Toronto, Canada
December 13, 2019

EECS 4422 – Final Project
Scientific Stream
Code available at: **https://github.com/Josh-Abraham/EECS4422-SatelliteImagePreprocessing**

Introduction

Within this technical report, the effects of preprocessing satellite images are evaluated with the aim of improving Canny edge detection. Satellite images are currently being used within many industries that rely on high accuracy data to provide products and services to their customers. This type of image capture is used in Cartography like Google Maps, industrial planning done by governmental bodies, or IBM's PAIRS Geoscope used within the agricultural and meteorological sectors. However, even with its widespread usage in industry, satellite imaging poses many security and privacy concerns for the general public. Due to these concerns, there is currently a limit on the resolution of satellite images imposed by the government at 50cm x 50cm pixel resolution. However, there is constant demand from industry to increase the allowable resolution in order to gain more insightful information from an image for their products. This technical report aims to prove that increasing resolution is not the only means of extracting more semantically meaningful information from a satellite image. Specifically, this report will be investigating the effects of various preprocessing techniques and how they enhance or detriment the application of Canny edge detector. Edge detection will be the focus of this experiment because it has significant uses within satellite images with respect to detecting buildings, roads, farmland divisions and other meaningful data.

Preprocessing Techniques

The preprocessing techniques used within the pipeline were hand tested to determine which would benefit the images the best most consistently. After testing various subsets of the images, 7 preprocessing steps were used to create the pipeline, however some of these steps are conditionally applied to the images based off specific factors. The techniques are listed in order of application below and their intended benefits.

White Balancing

The purpose of a white balancing filter is used to remove the coloured haze on an image, and make it look as if it is under white light. This may not seem like a significant step with respect to edge detection, however the effect of haze on an image can greatly reduce relative contrast and by doing so reduce the output of the edge detection. Since the light source in satellite images are not constant, and weather conditions can reduce the overall condition of an image by white balancing all images the output will be more consistent to what would be seen under normal conditions. This type of algorithm becomes especially useful when looking at images with a slight fog, at dusk or dawn. This algorithm was done first in the pipeline because it is the only filter that is colour dependant as it is transferred to the CIELAB colour space, then once it is finished it is converted to greyscale.

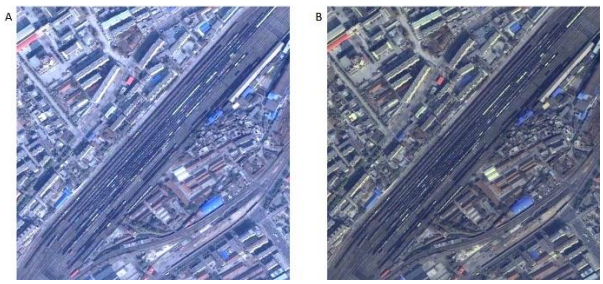

Figure 1: Demonstration of the affects of white balancing on an image. Image A shows the original satellite image, and image B shows the white balanced version.

Fuzzy Histogram Hyperbolization

This type of filter is a specialized subset of the classical histogram Hyperbolization. The goal of a histogram Hyperbolization is to take an image and adjust the brightness of the image with respect to the probability density found from the histogram. Originally when brightening a photo, effectively the histogram is being flattened into the higher pixel value buckets. Whereas the fuzzy histogram approach uses a logarithmic scale when applying the brightness increase, as this is how humans interpret light. Even though the goal of these preprocessing techniques is not to improve the quality for a human, the fuzzy histogram Hyperbolization is very effective at increasing localized contrast. This is very helpful when looking at the edges as localized contrast changes can help to distinguish edges that may have previously been missed. The following calculations are applied for each pixel within the image where β is the fuzzy factor, g represents the original image, g' is the output of the processed image.

$$\mu(g_{ij}) = \frac{g_{ij} - g_{min}}{g_{max} - g_{min}}$$

$$\mu'^{(g_{ij})} = [\mu(g_{ij})]^{\beta}$$

$$g'_{ij} = \left(\frac{L-1}{e^{-1}-1}\right) * \left(e^{-\mu(g_{ij})^{\beta}} - 1\right)$$

After applying the fuzzification to each pixel the resulting image has a lot more significant information retained. In the picture provided, the red region highlighted in both images shows the importance of applying a fuzzy histogram as the localized contrast in that region is accentuated. This is helpful as this to a human would be considered a significant edge but may not have been recognized within the Canny edge detection of the original image.

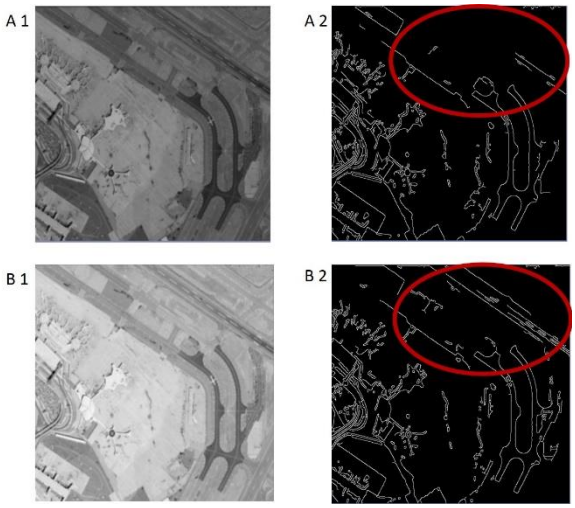

Figure 2: Demonstrates the affect of Fuzzy Histogram Hyperbolization. Image A1 is the original image, and A2 is the resultant Canny edge detector. Image B1 is the image after applying the fuzzification, and B2 is the resultant Canny

Median Blur

Satellite images are usually fine with respect to salt and pepper noise, at least within the dataset that this pipeline was built for. However, when looking at a ground truth of significant edges in an image, small pixel variations would not be a significant edge. In the example image below, thee effect of median blurring removes small variances within the image that would not be significant edges based off the HED algorithm. This is also a logical operation because a human ground truth would probably also ignore the minor variances within a satellite image and focus on the more obvious edges. The image set below shows the original image (A), the

Canny output of the image without the median blur (B), and the Canny output of the image with the median blur(C). It is evident that this filter helps to reduce the noise within the image with respect to small localized variances.

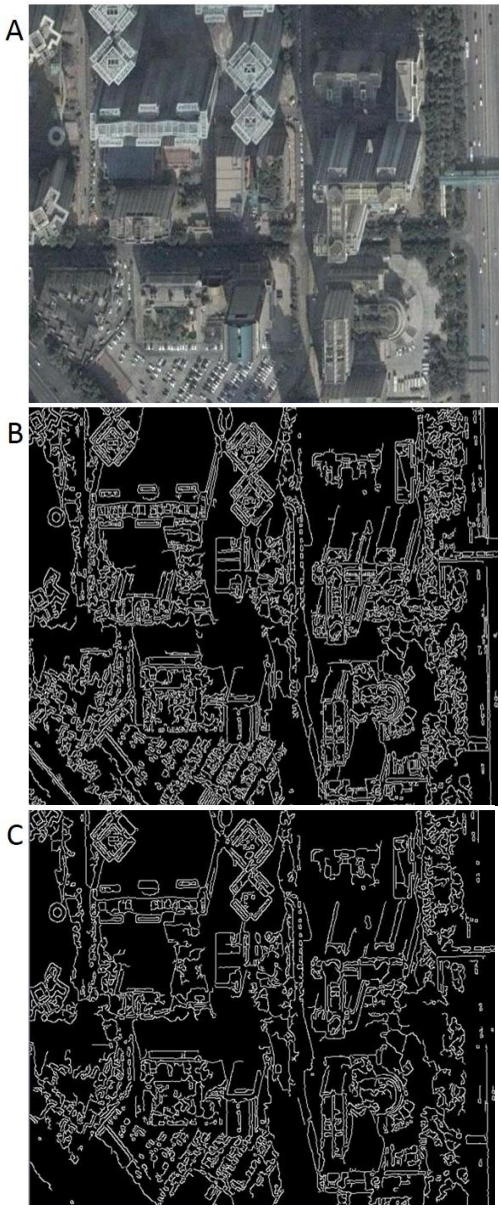

Figure 3: Image A shows the original satellite image, image B shows the original Canny, and C shows the Canny after applying a median blur. It is evident that median blurring reduces noise in Canny by removing small image variances.

Anisotropic Diffusion

This type of filtering technique is intended to maintain significant parts of an image, specifically lines and edges while reducing the overall noise in the image. Conceptually the filter convolves an isotropic kernel with the original image. The logic of an anisotropic diffusion is quite similar to a standard Gaussian blur; however, the variation comes from the diffusion constant, which is effectively an edge detector, which will be ignored by the blurring. The equation of the output image is given below, with *c (x, y, t)* being the diffusion rate equation and *I* represents the image matrix in greyscale, and K is a function of the noise of the image.

$$Anisotrpoic\ Equation = \nabla c * \nabla I + c * \nabla^2 I$$

$$Diffusion\ Coefficient = c(||\nabla I||) = e^{-\left(\frac{||\nabla I||}{K}\right)^2}$$

Blurring

Blurring the image is a helpful step within the preprocessing pipeline to help achieve a similar step as with median blurring. The only difference here is this is not a rule-based filter and so will equally blur all aspects of the image. This step is helpful in reducing minor variance within the image. When blurring it is significant that this happens later on within the pipeline. Multiple of the filters listed previously have the intent of reducing various types of noise within the image. Blurring prior to these steps would result in spreading the noise within the image, and by doing so would make the other filters less efficient at their respective tasks.

Blur (Conditional Secondary Blur)

This filter is identical to the blurring filter above, however this secondary blurring is not a mandatory step within the pipeline. Instead

it is a conditional blur based off the preprocessed image's histogram at this point in the pipeline. After splitting the image into a histogram by pixel value, the algorithm does a calculation to see how many bins are within the range $0 < X < 36$, which is 0.01% of the image. If there are more than 25 instances of this, then a secondary blurring should be run. The logic of this is that if there are at least 10% of pixels (255/10) holding a small number of pixel values, it is likely that the image still has a lot of small details that would negatively affect the edge detection.

Dual Sided Contrast Normalization (Conditional Step)

The dual sided contrast normalization is two forms of contrast adjustment within one portion of the pipeline. This is a conditional step within the pipeline that uses a 25-bin histogram to determine if the image is skewed, and if so to correct this skew so that the pixel values have a better spread. This is dual sided as it looks at both left sided and right sided skews in the histogram and can normalize in both directions. This is the final step of the pipeline, as it also includes a normalization to ensure the image does not become saturated. This normalization will benefit not only the contrasting but any of the above image mutations that may have led to saturation. Below is an image with a heavy right skew (A), the image after going through a contrast normalization (B).

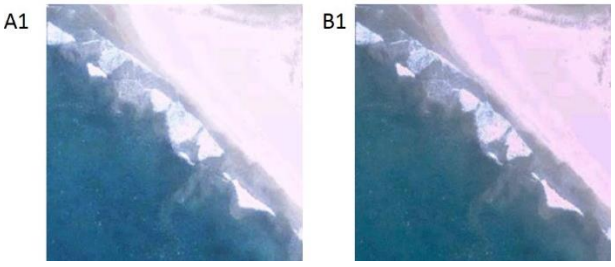

Figure 4: A1 is the original image, and B1 is the contrasted version

Results

In this section, the pipeline for comparing the unprocessed and processed version of the images is discussed. The data for this experiment will be the 19-class satellite imagery dataset which offers 19 unique regions with a total of over 900 images to test with. The images will first go through a Holistically-Nested Edge Detection (HED) algorithm to generate a base truth of the edges within the image as a benchmark to make comparisons against. The HED algorithm was created by members of the Department of CSE from the University of California.

After the HED algorithm is performed on every image they will go through additional processing to make their outputs more like a Canny edge detection output. The reason for this, is that the edge detected images will be compared based on their Structural Similarity Index (SSIM) and so having the images in similar visual representations will create more meaningful SSIM results. SSIM is used to detect the similarity between two images, where one of the images is the ideal version of the image. In this experiment, the processed HED output will be the ideal image, compared against the original Canny and the preprocessed Canny.

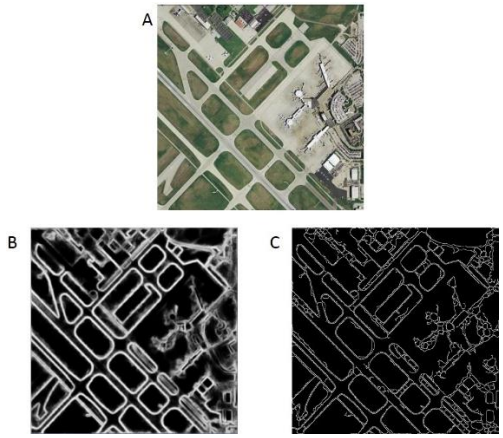

Figure 5: Image A shows the original image, B shows the HED output, and C shows the HED in a way that is similar to a Canny detection

The plot (see Figure 7) shows the outputs of the SSIM for 36 random images within the dataset with two images coming from every subclass. The two bars show the preprocessed SSIM index in blue and the original image in purple. The data shows that for all subclasses of the dataset, the preprocessing pipeline does in fact improve the edge detection similarity. For these 36 images there was an overall improvement of quality by a factor of 27.87%.

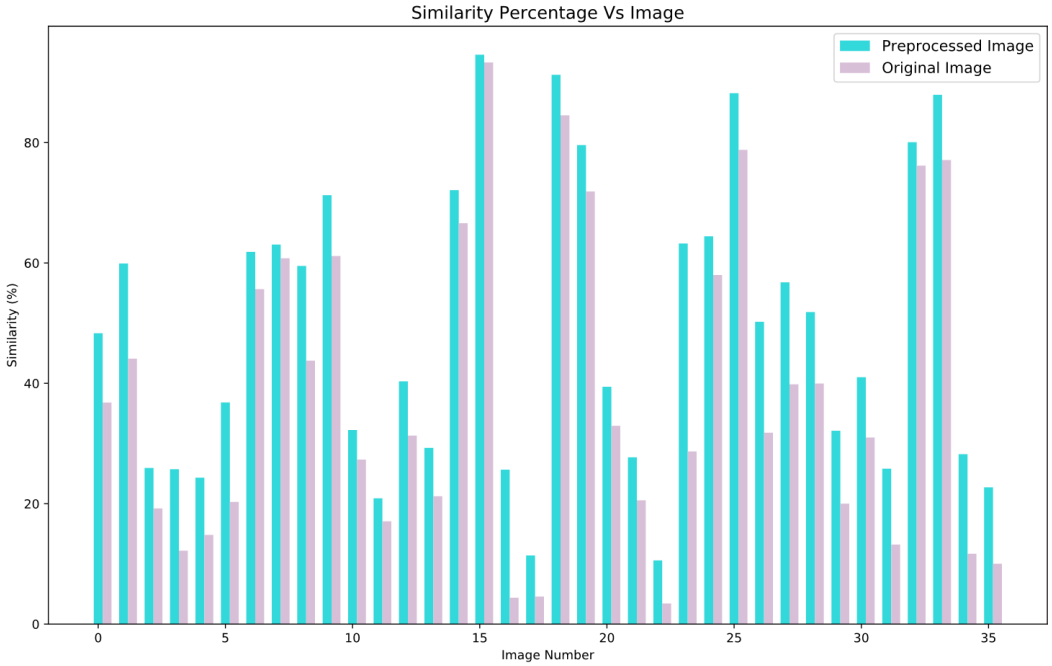

Figure 6: SSIM index results for 36 images from 19-class satellite imagery dataset. Purple represents the original images against the HED output, and the purple represent the preprocessed against the HED output.

This pipeline was also further tested on the entire dataset of 908 which resulted in an overall improvement of 34.20% with respect to the original images. However, when run all the images, there were 41 images that became worse after the preprocessing, with an average decrease of 3.77%. So even though there is a decrease, with respect to the overall improvement seen by all images it is small. Below are some outputs of the original image, the Canny of the original image, and the Canny of the preprocessed image. It can be seen that the preprocessing reduces less significant edges and helps to create a more cohesive understanding of semantically important information.

Conclusion

Overall, the pipeline demonstrates that preprocessing does in fact have a very significant impact on the quality of Canny edge detection. Moreover, further optimization of the algorithm may in fact lead to further quality enhancements that make it so that all images are positively affected. This research was a success in attempting to prove that increasing satellite image resolution is in fact not the only way to enhance the quality of edge detection algorithms. Preprocessing pipelines can be used to drastically improve results, without needing to infringe on the privacy of the general public. This further shows that preprocessing techniques may be a cost-efficient way to extra more meaningful data without the need to put higher resolution image probing into space.