Final Report: AI in Geoscience: Blur Detection in Images

Zhang Boen BS in Computer Science
Zachary Golden BS in Computer Science
Faculty Advisors Dr. Renzhi Cao, Dr. Tarka Wilcox, and Dr. Yajun An

Spring Semester 2018

Abstract

Currently, in the field of Geoscience, numerous images of topography are captured via drone imaging and manually sorted for the purpose of monitoring changes in topographical structure over time. Utilizing machine learning for blur identification of images, our project aims to assist in the creation of 3D maps from remote drone imaging in geoscience. With images provided by the Department of Geoscience at PLU, by calculating the Laplacian variance of a single channel of an image, we trained a support vector machine (SVM) with more than 2000 images and used our model trained from the SVM to classify images on the level of blurriness. Using a React based web application, users can submit images and manage projects by storing multiple datasets and set parameters for the amount of blurriness deemed acceptable for any given project. The blurriness of the images is rated (0 to 9) to determine if images are in the acceptable range for the modeling datasets (currently we find out score of 4 is the most reasonable acceptable range). Implications of this project could be used to efficiently sort or classify large image datasets.

Github repository: https://github.com/zachgolden/geoai

Table of Contents

Abstract	**2**
Introduction	**5**
Requirements	**6**
Design	**8**
Implementation	**13**
Future Work	**16**
References	**17**
Glossary	**18**

List of Figures

Figure 1: Domain Model
Figure 1.1: Image Prediction SSD
Figure 2: SVM Batch Processing
Figure 3: Login Implementation Diagram
Figure 3.1: Login Update Diagram
Figure 4: Image Upload Diagram
Figure 5: SVM Training
Figure 5.1: SVM Prediction

Introduction

With the use of drone imaging and remote sensing data the Geoscience field is utilizing the growing access to the technology to track land movement through creation of three-dimensional maps to help recognize potential land hazards. However, the process of collecting large amounts of images of a site over time produces an issue as many images taken by are unusable as they have been blurred during the capture process. The purpose of our project is then to utilize machine learning to help classify and detect blurring of the captured images from any given site and help to sort unusable images out of the datasets.

In this project we decide to use Support Vector Machine(SVM), a machine learning algorithm to help us accomplish the blur detection program. SVM is a supervised learning algorithm that analyze data used for classification; if the training data is carefully analyzed and the amount of training data is huge enough. SVM could help us get a model which could process the picture efficiently.

This project is different than our regular class projects as we don't have clear guidelines to follow in regards to how we can detect or classify images as blurred or how to use machine learning. Because this is a complex task we searched through published papers which cover the similar topic. We read through Blurred image detection and classification in which they utilized an SVM in order to classify and estimate the extent of blur in an image (Hsu and Chen). Another resource we used was Analysis of focus measure operators for shape-from-focus which provided many options of techniques to classify the focus or lack thereof in an image and advised us to use the laplacian method (Pertuz et al. 2013). A helpful resource was the CERTH image dataset, No-Reference blur assessment in natural images using Fourier transform and spatial pyramids, which provided a comprehensive external data set for image testing (Pertuz et al. 2013; Mavridaki and Mezaris 2014). Other research we reviewed helped to find focus in our project and provide insight to the topics of image detection and classification of blur (Wei Xu et al. 2013; Chen et al. 2010; Renting Liu et al. 2008).These papers provided us with a basis for our project and helped us to learn more information about image processing and machine learning technique. Which are important for us to accomplish this project.

Our educational objectives for this project were to:
Understand the algorithms and theories of machine learning and artificial intelligence, remote sensing, imaging processing concepts, and web development skills.

Our functional objectives for this project were to:

- Utilize machine learning to classify and *detect/predict* blurring in images.
- Allow users to upload images to run the prediction on non-locally.
- Once uploaded allow users to select images to predict.
- Process the images and run the prediction then return the *quality/assessment of blur* of the image to user.

- The program should process the image on a server, so it should have:
  - Be accessible on any Mac, Windows, or Linux machine
  - A website for user to upload images, get results and manage projects
  - A database to store users uploaded images and information
  - Provide an estimated time of processing image to users

Requirements

The requirements for this project were based on our connection with the Department of Geoscience at PLU. This provided our primary user profile based of off Dr. Tarka Wilcox.

Dr. Tarka Wilcox, Assistant Professor of Geosciences at PLU.

Dr. Wilcox started research last summer using consumer-based drone technology to collect images of land to use photogrammetry and create three-dimensional maps. By creating such maps Dr. Wilcox and his student researchers can track areas with potential for landslides or other geological hazards. However, Dr. Wilcox and his student researchers want to access multiple sites and visit each site on multiple occasions producing large amounts of image data that must be manually sorted before they can start the process of creating the maps. With busy schedules during the school year it can be a time-consuming process to sort and find the best data that can be used to create the desired maps.

User Stories:

Image Analysis:
A user wants to be able to upload and image and be returned a numeric result of the quality of the image in regards to its blurriness.
A user wants to be able to detect blurring issues of movement during capture, the analysis should be able to detect cases of radial, zoom, and vertical/horizontal motion blurring.

Training:
A user wants to see what data does the machine learning program take in order to generate the machine learning model.

Website:
A user should have login credentials provided in order to limit access and prevent overload on the server.
A user should have to login in order to access files and information.
If a user is logged in they should be able to upload to the database on the server as well as be able to view and remove files from the database if they are logged in.

Design

Our final design is use support vector machine(SVM) to detect the image is blur or not. SVM is a supervised learning model so we have to 'supervise' the result to the SVM. SVM will take the training data from the Laplacian algorithm and human will supervise the result(score 0-9 to each picture) then let SVM come up an model. To predict the result, testing image's Laplacian algorithm score will be put into the SVM and to get the best fit.

Domain model diagram:

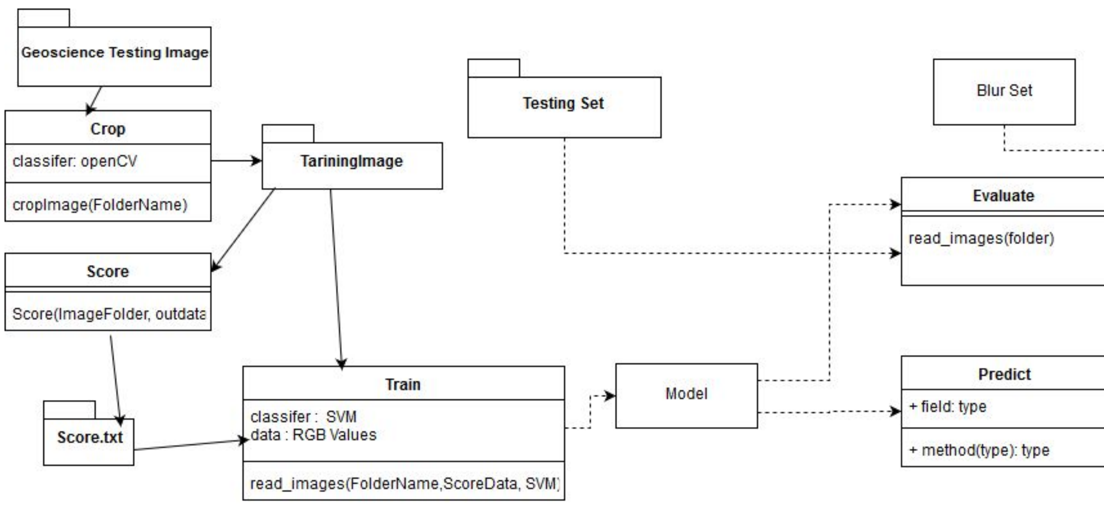

Figure 1: UML class diagram

SSD for user to predict image:

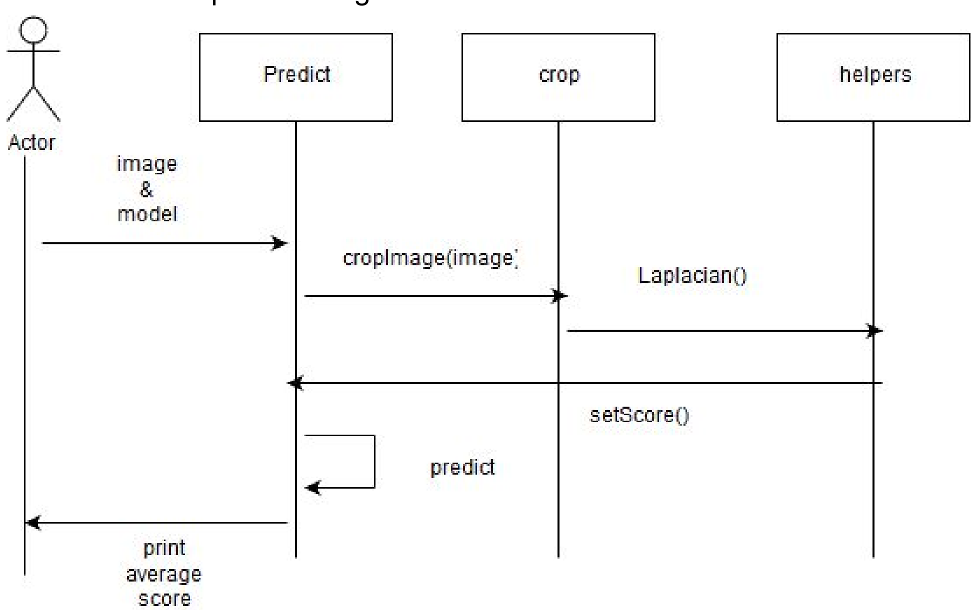

Figure 1.1: SSD

To test the accuracy of the SVM model, we manually scored a set of testing images; run SVM prediction program to get the prediction result for the same set of images; then compare the difference.

Use SVM model to batch process a testing images folder:

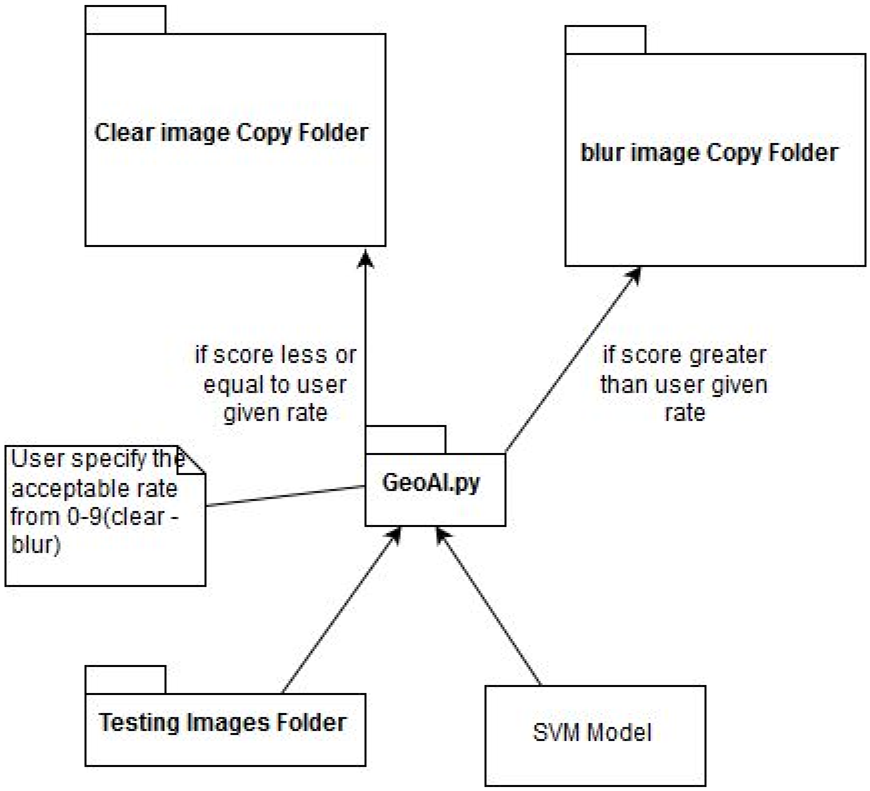

Figure 2: batch processing

The design of the website was to create something that was simple and easy to use. The functionality of the website was limited to authenticating users, allowing uploading and removal of images to the database, and running of the prediction with a returned result. During the design phase of the website we wanted to work with the MERN (MongoDB, Express, React, and Node.js) stack to create the web application.

To handle our authentication/login process we used an external user authentication product called Auth0 which allowed us to create a custom API that we could call to authenticate as well as manage and store externally. Auth0 uses JSON Web Tokens (JWT) which was similar to our initial approach using passport to authenticate, but Auth0 also provides ease of access for us to use google accounts to authenticate and promotes single sign on for future implementation of other software or sites.

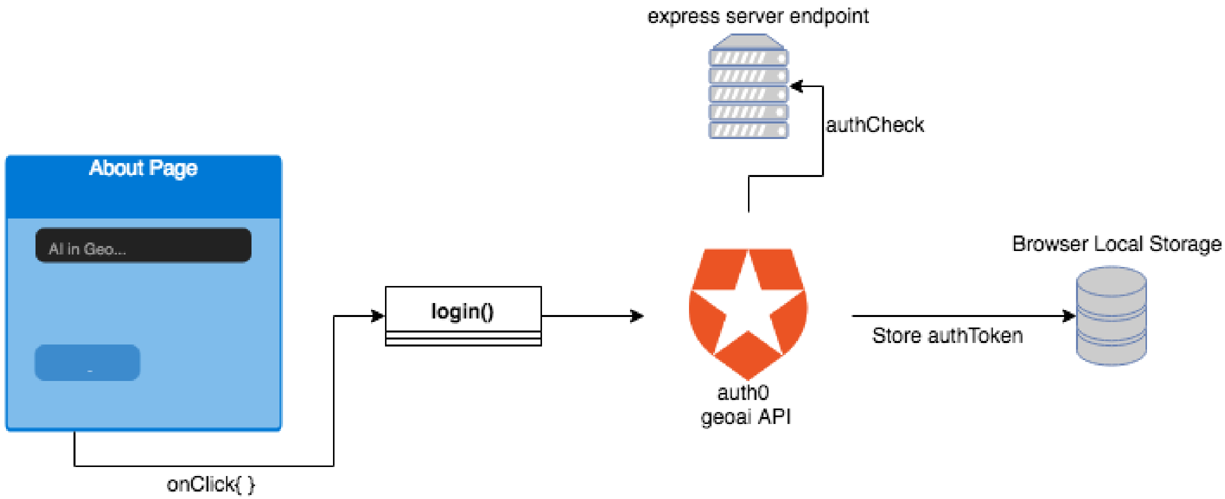

Figure 3: Login Implementation

This diagram represents the action when a user attempts to login. On click the login function is called from our authUtilities class and begins the process of calling the Auth0 API. Once the API is called routes to the backend express server which handles the authCheck to the geoai API. This routes to a default login which can be managed to allow for login or signup from Auth0 and if a user is authenticated it stores the returned token in the browsers local storage. If a user is already logged in this process is avoided as the next figure shows.

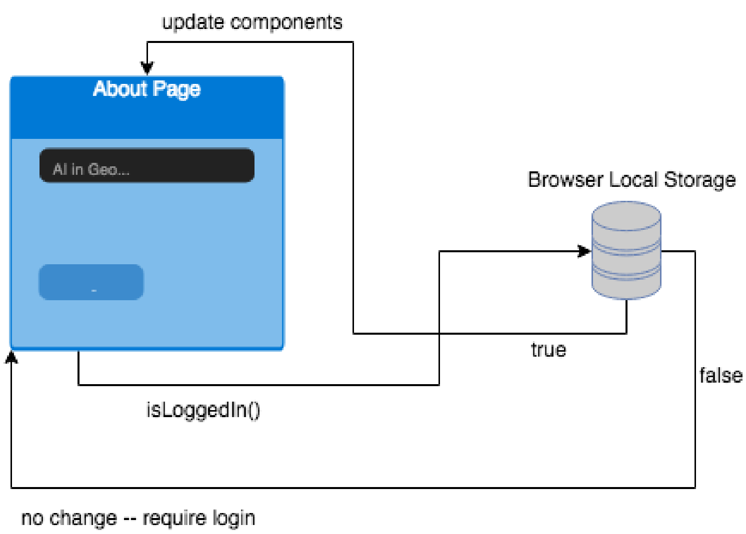

Figure 3.1: Login Update

If a user is already logged in then an authentication token will already exist in the browsers local storage. If a user has just logged in the isLoggedIn function will update the component removing the login button and replacing it with a link to the file directory and upload page and providing a logout button on the navbar.

Once a user is logged the link to the file directory and upload is shown to the user. Here a user can upload a file or remove a file from the MongoDB database which uses GridFS in order to handle the large file size. The database connection is made through mongoose on the backend express server and then when the user select a file to upload or delete the appropriate methods call and fetch through the file api to the backend express server to upload or remove the files.

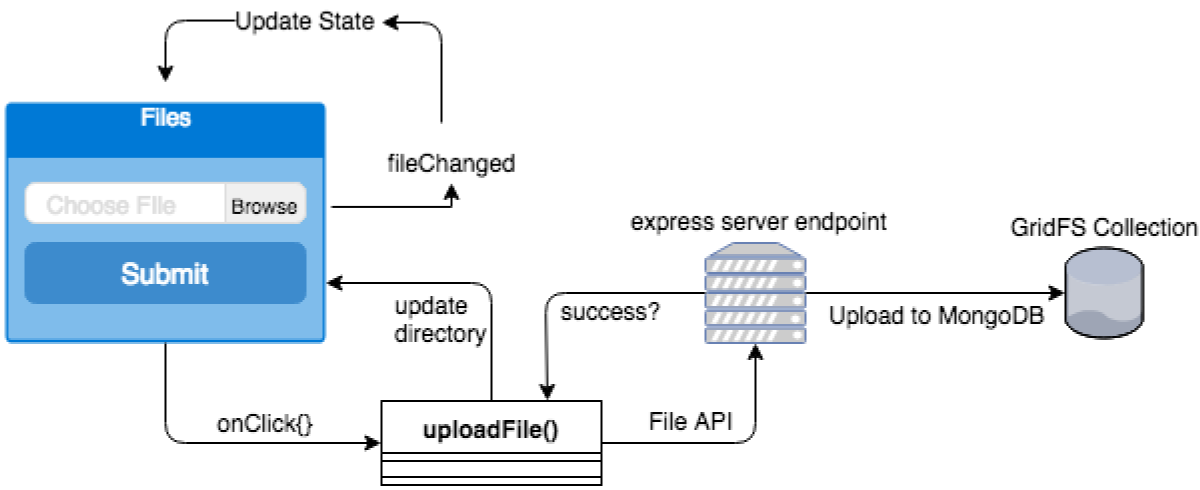

Figure 4: Upload Process

This diagram represents when a user uploads a file on the site. Once a file is selected the state of the component is updated and when upload is clicked the upload is routed as above.

Implementation

1) First model:

Using the CERTH image data set, read in the image and resize the image to 800X600, since this is the blur detection project, we also convert all the images to grayscale. The score we use to train is achieve by Laplacian algorithm. In this model, we only set the image is either blur or not blur. So all the score given by Laplacian algorithm will be set to a score 1 or -1 depends on the result given by CERTH image data set, then push to the SVM.

Evaluate the testing image: before program input the testing image into the SVM, the testing image also get resized and grayscale. Then get a score by Laplacian algorithm. The score given by Laplacian algorithm will be send into the SVM and SVM will try to find the best fit for this score and return either 1 or -1 as result. If the result is 1, means the image is a clear image, -1 as the blur image.

2) Second model:

Switch the training image from CERTH image data set to the image set provided by geoscience department.

3) Third model:

Use the same image set from geoscience department, before the image get score form Laplacian, every single image will be crop to a set of 200X200 image, then human give this image a score between 0-9 and store the image score in a separate file.for scoring: 0 is very clear image and 9 means very blur image, this score combine with the score given by Laplacian algorithm will be push into the SVM and gets a model from it.

Evaluate the testing image: testing will also be crop to a set of 200X200 images and each image get a score by Laplacian, SVM use this score and return an int between 0-9 for each small image. The result of the whole picture is the average score for all the small images generated from the testing image

4) Fourth & final model:

In this model, instead crop one picture to a set of 200X200 image, we crop them to 300X300 so that each image could include more information in it so it's easier for human to score each image.

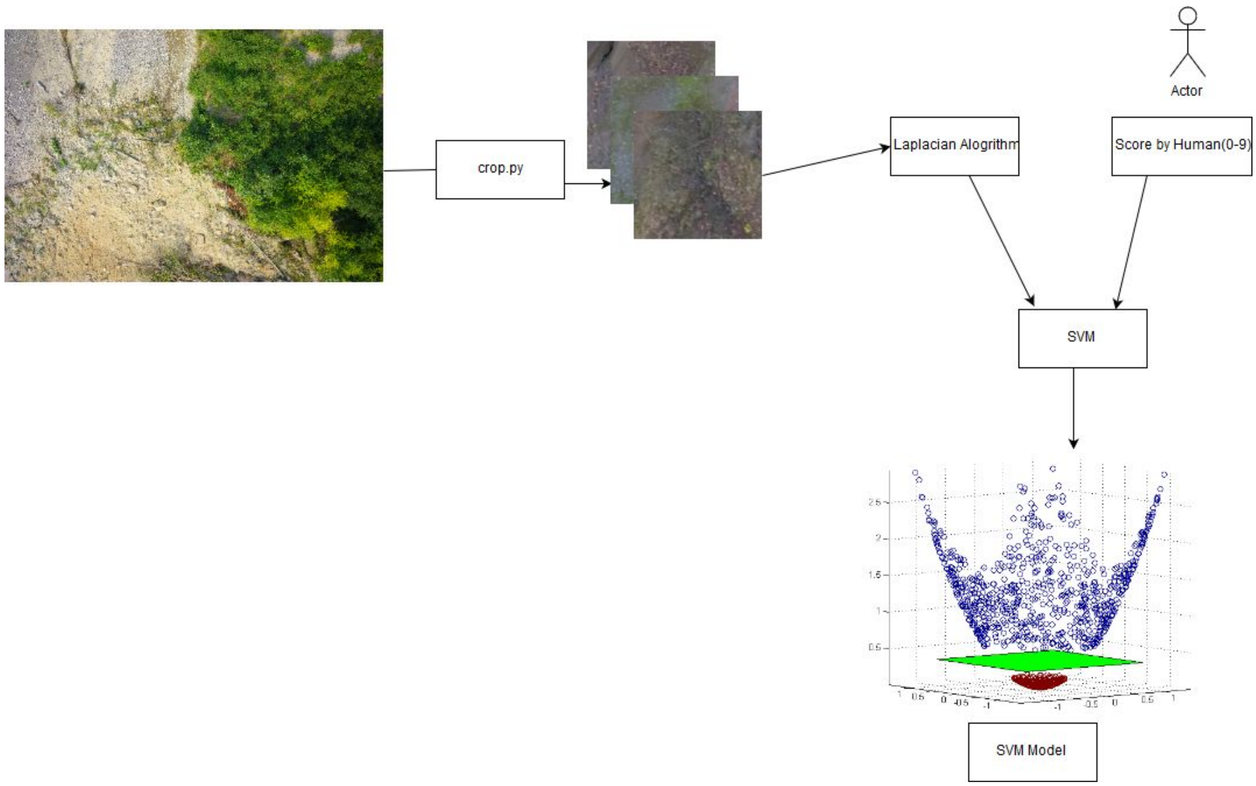

Figure 5: Overall pipeline for training SVM model

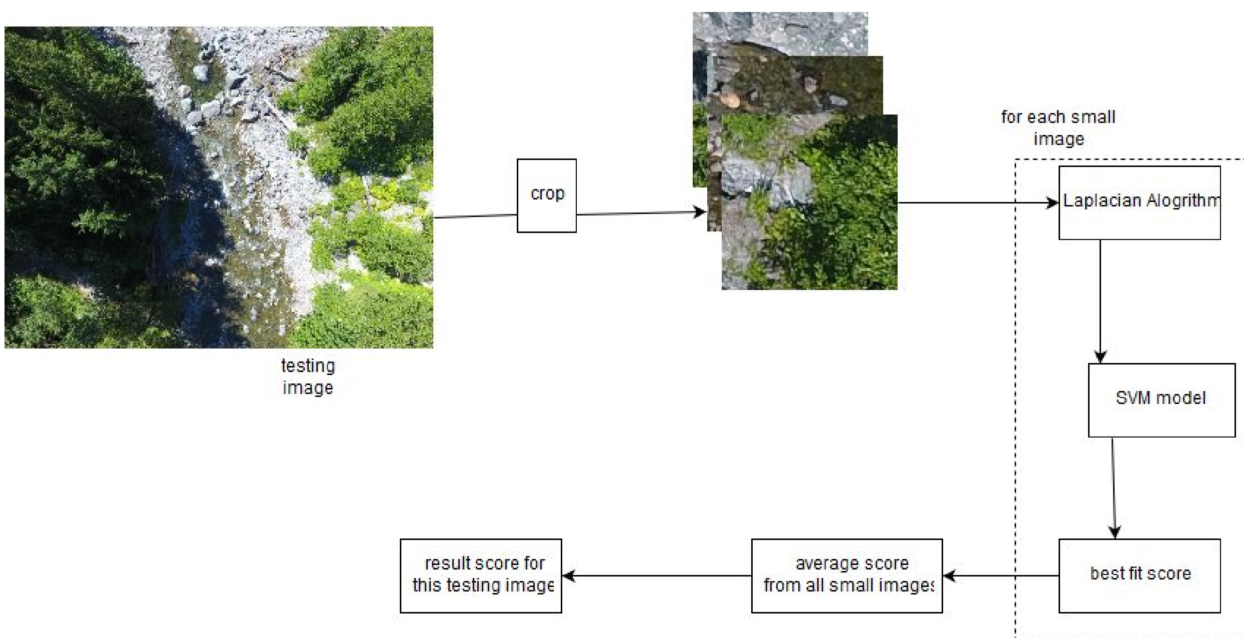

Figure 5.1: Overall pipeline of using SVM model to predict testing images

The process involved in creating our website was to first build our front-end portion. By choosing to use React we utilized the create-react-app boilerplate to jumpstart our project and build a basic page structure for our web application. After completing the basic front-end structure, we then began working on user authentication which required the use of Node.js and Express to setup a backend endpoint. We created the backend endpoint and then connected to our custom API created through Auth0. The process of connecting required changing some elements of the front-end but provided a method to easily connect and restrict access to portions of the web application.

With the user authentication and React site routing restricted we then moved on to connecting the site to a database. Utilizing MongoDB, we worked with the existing GridFS utility provided by MongoDB to handle the storage of our large data files. We first connected to a local mongo instance and tested uploading and removing images from the database. After we had authentication and file uploading working properly we then pushed our site onto the department server simon. This took some adapting from working on a local instance, but after changing addresses from local and addressing the routing and callback issues we had our site functional and ready to connect the prediction to the web application.

In implementing our project, we communicated primarily through Slack and Email and communicated via text messages and phone calls for more urgent matters. This helped us to schedule our meetings and work around dates that may have later conflicted with our school and work schedules.

We utilized our workroom and met at a minimum of twice a week either during or directly after our capstone class period dependent on our class meetings. We attempted to use the agile methodologies and worked on two-week sprints meeting with our advisor Dr. Cao every other Thursday during the Fall semester and every other Tuesday during the Spring semester.

For our image data and python code we used Google drive to share information among both partners and advisors. When we went to implement our website, we used GitHub to manage changes and commits as well as providing access to quickly clone the repository and run tests.

Our main issues that we discovered while implementing the project were that initially we didn’t understand how to address detecting blur. This required a lot of research as well as insight from Dr. Wilcox as to what would be most beneficial for detecting blur in the images provided by the Department of Geoscience. Another primary issue that we faced was after we had the website functioning on the server we did not account for issues with running python on the server. This halted our progress and prevented us from connecting to the website as we didn’t have the proper permissions on the server and were unfamiliar with the tools we were using more specifically flask.

Future Work

The requirements that we did not get to fully implement were to connect the prediction function to the web application, to use a Convolutional neural network to solve the problem of blur detection, and to add a GUI or desktop app for the program's offline version.

If we were starting over in our project we would begin by working with existing attempts at blur detection. We should look more into the picture processing technique: currently most of our picture processing function is from the openCV library, and sometime openCV perform differently than what we expect. So if we could start over our project, it would be more beneficial to spend some time to find or even develop our own image processing technique which would have benefit the performance of our project.

In the future we plan to continue our work by adding Dr. An's gradient feature recognition in addition to our current SVM implementation to produce higher accuracy in detecting blur. Separate from our current SVM we would also like to adapt the current process for use with a CNN to handle the blur detection. An addition that we discovered later in our process was possibly providing a python application in place of our website that a user can run locally instead of being dependent on the external service of the website. Also we would like to work with new research using Super Resolution to test for higher accuracy for prediction of blur detection.[1]

References

Ping Hsu and Bing-Yu Chen. 2008. Blurred image detection and classification. In Proceedings of the 14th international conference on Advances in multimedia modeling (MMM'08), Shin'ichi Satoh, Frank Nack, and Minoru Etoh (Eds.). Springer-Verlag, Berlin, Heidelberg, 277-286.

Said Pertuz, Domenec Puig, and Miguel Angel Garcia. 2013. Analysis of focus measure operators for shape-from-focus. Pattern Recogn. 46, 5 (May 2013), 1415-1432. DOI=http://dx.doi.org/10.1016/j.patcog.2012.11.011

Hashim Mir, Peter Xu and Peter van Beek. 2014. An extensive empirical evaluation of focus measures for digital photography. Proc. SPIE 9023 Digital Photography X, 90230I (May 2014). DOI=https://doi.org/10.1117/12.2042350

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. https://arxiv.org/abs/1603.08155

E. Mavridaki, V. Mezaris, "No-Reference blur assessment in natural images using Fourier transform and spatial pyramids", Proc. IEEE International Conference on Image Processing (ICIP 2014), Paris, France, October 2014.

Wei Xu, J. Mulligan, Di Xu and Xiaoping Chen, "Detecting and classifying blurred image regions," *2013 IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, 2013, pp. 1-6.

X. Chen, J. Yang, Q. Wu and J. Zhao, "Motion blur detection based on lowest directional high-frequency energy," *2010 IEEE International Conference on Image Processing*, Hong Kong, 2010, pp. 2533-2536.

Renting Liu, Zhaorong Li and Jiaya Jia, "Image partial blur detection and classification," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1-8.

Glossary

Topography - The arrangement of natural and artificial physical features of an area of land.

Drone Imaging - The capture of photos (still images) or video from a remotely operated drone. Commonly referred to as drone photography.

Photogrammetry - The science and technology of gathering information about the environment/topography through recording, measuring, and interpreting images and patterns.

SVM (Support Vector Machine) - Given labeled training data (from supervised learning) the algorithm produces an optimal plane to classify new examples.

Blur / Blurriness - Defined as being clear or indistinct. For the purpose of our project blurriness represents images that are unclear and cannot be used in the photogrammetry process. The distinct types of blur that we focused on are listed below.

Radial Blur - This is a type of blur that is distorted around a single focal point. In our case this is typically the center of an image, which is the case in the generated blurred data that we created in photoshop, and is a common side-effect of the instability of drone imaging.

Zoom Blur - This type of blur results from adjustment in the zoom of the lens while the shutter is open resulting in distortion

Motion Blur - This is the most common type of blur in relation to drone imaging due to the nature of the drones constant movement. The blur produced appears to be streaked and can be further classified in to horizontal motion and vertical motion in the direction of blurring due to the movement of the camera.

Remote Sensing Data - Data or information gathered from satellite or aircraft based sensor technology to classify objects or features on earth.

Land Hazards - Any hazard that may result in loss or injury, damage property, or degrade the environment such as lands prone to erosion, flooding, or instability of slope.

Machine Learning - Application of artificial intelligence and statistical techniques to provide systems with the ability to “learn” and improve without explicit manipulation/programming.

Model - An artifact resulting from training of a machine learning algorithm with provided data.

Laplacian - A differential operator that results the left member of Laplace’s equation. A scalar function.

CERTH Image Blur Dataset - This is a photo dataset providing naturally and artificially blurred, and undistorted images to test and evaluate image quality.