INTRODUCTION TO COMPUTER VISION

AUGMENTED REALITY PEAK IDENTIFICATION

December 15, 2020

Carson Stevens, Ben Brokaw, Nam Do, Brooke Suesser

Colorado School of Mines

INTRODUCTION

The goal of the project is to take a photo as input and label the mountain peaks that are present in the image. Using EXIF data extracted from the photo, the team recovers important information such as latitude, longitude, and heading, as well as focal length, sensor dimensions, camera orientation, alongside other important features. Given this data, the angle of view (AOV) and the homography (relative to the topographic map) can be calculated. The team has compiled a database of thousands of peaks around the world, with more focus on peaks in Colorado. We aimed to compare the scraped GPS data to the data in the database for peaks. Once the nearby peaks are identified, the team will use the AOV and homography to filter the peaks that are actually in view. Once correctly identified, the team aims to overlay the peak name and an arrow pointing to it on the image.

PREVIOUS WORK

Several groups have studied and solved this problem using various methods of peak determination. Using pixel-wise skyline detection to match mountain shapes has been done to identify peaks [2]. However, this method can be unreliable due to differences in weather, cloud coverage, lighting, and perspective the image is taken (E.G. A more aerial view where the peaks are not located on the skyline). These variables can make it more difficult to isolate and threshold out the important regions in the image. Another flaw in pixel-wise detection is that not all peaks have a specific highest point and can come in many different shapes. For that reason, the group decided not to move forward with this option. Elevation maps can be used to identify local maxima in an area to identify peaks [4]. Additionally, the latitude and longitude of many peaks is known and has been compiled in databases. Some groups used this data to identify peaks in an image, and developed an app to identify peaks in real time. This can be done completely offline, which is essential for an app that can be used in remote

locations. Phone service is often poor in the mountains. We drew inspiration from the work of these groups.

Peak Classification Background

Through our research, the team realized the definition of a peak varies depending on the organization and geographic location. Many start classification with the peak's isolation and prominence to other nearby peaks, but others add complementary criteria that are more subjective and could change with the evolution of alpinism. These include features from the mountaineering perspectives such as:

> "the qualities of the routes reaching the peak, the historical significance, and how frequently it is climbed" [1].

This is important to note because the dataset was subjectively created and labeled by the team, so not all "peaks" may be included in the dataset.

TECHNICAL APPROACH

The team approached this problem from several angles because of the limitations of the previous works. Instead of using pixel-wise skyline detection to find peaks, the team aimed to create a Faster-RCNN that would identify mountain peaks. The reasoning behind this architecture lies behind its speed and the type of detection that is done. Instead of identifying single points, the network is trained to draw a bounding box which eliminates the problem of differently shaped peaks.This is combined with the concept of using a homography to filter nearby peaks and transforming their 3D coordinates (latitude, longitude, elevation) into the perspective of the camera. This method yields relative pixel locations in the input photo where peaks in the database are found.

0.0.1 Dataset

Image Dataset

Collection, Cleaning, and Labeling To compile the dataset of mountain peak images, the team came up with a list of search terms that covered mountain ranges in every continent. The hope of doing this was to make the model as inclusive and robust as possible. Using these search terms, the first 150 google images were scraped for each term. Once collected, the team entered the data cleaning and labeling process. Not every image that was scraped was a good example, so those pictures were excluded from the dataset. If the picture met the teams quality standards, bounding box coordinates for all the peaks in each image were recorded. As previously mentioned, this process is highly subjective, but the team focused on the primary goal of making sure that the center of the bounding box was at the highest point on the peak in the image.

Data Augmentation After the data was cleaned and labeled, the team decided to use image augmentation techniques to make the dataset larger as well as more robust. The augmentations also warped the bounding boxes, so their new positions were recorded with the new photo. The different augmentations included a mix of horizontal and vertical flipping, random rotations, random translations, random scaling, random shearing, and finally all images were sent through a re-sizer that formatted all the input into the size of 516x516px.

Mountain Peak Dataset

Using data scraped from KML files, the team created a database filled with mountain peak latitudes, longitudes, elevations, and names. The team is able to query the peak database for the nearest mountain peaks of where a picture is taken. While the KML dataset collect for Colorado is very comprehensive, the team wanted to expand the capability of the system beyond the KML data for just Colorado. To do this, the Overpass API is used to query for

additional nearby peaks. This uses the Open-Street Database to give the team mountain peak data from anywhere in the world.

0.0.2 Faster-RCNN

At this point, the team entered the training stage for the Faster-RCNN. The implementation was based off the main paper [7]. The team concluded that the network would benefit from transfer learning and thus tried to create a model transferred from Resnet50. After the transfer learning blocks, a Region Pooling Network (RPN) is used to evaluate different regions of the picture for probabilities that they are a peak, background, or neither. Theses probability regions are referred to as Regions of Interest or ROIs. Given a probability higher than the peak labeling threshold, the region is sent to the Classification block of the network that tries the classify the region as either a mountain peak or background. If classified as a peak, the bounding box for the region is re-scaled to the original image size. In line with our labeling strategy, the center of the bounding box should represent the highest point on the peak.

0.0.3 Peak Filtering

This allowed us to find the peaks in the image, but the labels for each peak still needed to be identified. The photo's location can be used to query the peak database for nearby peaks. Those peaks locations enter a filtering process to find out which peaks are actually in view. The basic process uses a homography to transform points into the camera view, but then filters points by whether they are between the two vector edges created by the periphery of the field of view. After the peaks go through this filtering, a list a possible peaks is found and continues through the filtering process. Using Google's Elevation API, elevation samples between two locations can be queried which gives the information needed to calculate whether the peak is in the photo's line of sight or not.

0.0.3.1 Filtering Problems

Haversine Problem and Solution The first problem is that the Earth is not flat, and mountains are often viewed from large enough distances to effect the perspective. To rectify this issue, the spherical latitude, longitude, and elevation coordinates need to be transformed into a Cartesian perspective. Using the Haversine equation, the greater circle distance between two points is calculated and simple geometry allows for the coordinates to be transformed. What this looks like in practice is adding and subtracting values from elevation to get the data into the perspective of where the photo is taken. The Haversine Equation is as follows where φ stands for latitude and λ longitude.

$$d = 2R \arcsin \sqrt{\sin^2 \Delta\varphi 2 + \cos\varphi_1 \cos\varphi_2 \sin^2 \Delta\lambda 2}$$

Line of Sight Issues When determining the line of sight, the sampling rate between points becomes an issue. Spacing between samples does not guarantee that the elevations being sampled are actually the maximum values in the line of sight vector. This means that a point could be mis-classified as in-view because an intercepting object was not sampled in the query. The other big problem encounter was caused by perspective. A photo may appear to be taking a picture of the peak, yet the actually peak may be located out of view. An example of this would be taking a photo from the base of a mountain where the real peak is located just on the other side of the mountain. From the photo perspective, there is a 'peak', but the real peak location is not is visible line of sight. This was corrected by adding a small threshold that allowed the line of sight to be accepted if the interception height was less than the threshold. This value is highly depended on the picture. Using this method, the current list of peaks in the photo is further filtered to exclude peaks not in line of sight.

0.0.4 Homography Transformation

This final list of filtered peaks then goes through the homography process to transform the 3D coordinates into the camera's perspective. The points that are sent into the homography are first converted into Cartesian (x,y,z) values. The elevation coordinate again has to be adjusted using the Haversine equation. This process yields the approximate pixel location of the mountain peaks within frame. Another important aspect to note is that the team was unable to reliably scrape the camera orientation besides the heading which means that the mapped y values will be inaccurate, but the x values should at least be relative to each other. Another factor is simply the lack of precision in the camera's orientation and compass data. Because the collected data is not precise enough, the accuracy of mountain peak's reported pixel locations will vary using this method. While this doesn't map the pixels to the peak locations with high precision, crucial information can still be gathered. Most importantly, this gives the general x and y positions which can be used to give the order of peaks in the photo from right to left and top to bottom. This data is then compared to the results from the Faster-RCNN where the order of peaks can be mapped to the correct bounding locations in the original photo. After this, a label is generated and placed to represent the peak name and location for each peak identified in the picture.

0.0.5 Final Pipeline

The final pipeline for project follows the following process: Validate the photo for the EXIF information needed. If the information is present, the needed data in then scraped and stored for calculations later. The first step involves sending the photo through the Faster-RCNN which outputs bounding boxes and probabilities. Next, nearby peaks are queried and filtered using the AOV and line of sight functions. Peaks that are found as visible then go through a homography transformation to the camera's perspective. The pixel coordinates are then ordered from left to right and matched with the bounding boxes ordered left to right.

EXPERIMENTAL RESULTS

For quality annotations, our program needs to be able to accurately identify the pixel location of a mountain peak within an image. Our initial approach involved finding edges and corners in the image and then identifying the peaks. Unfortunately, after many hours of trial and error we were unable to produce satisfying results. To rectify our peak identification problem, the image is sent through a Faster-RCNN architecture with transfer learning from the Resnet50 model to place bounding boxes on the mountain peaks within the photo. The final model used the parameters from the original Faster-RCNN paper (300 ROIs image size of 516x516px) [6]. The final model will train up to the due date as the models loss is still declining indicating we can achieve better accuracy given more time. The model has currently undergone over 250000 train steps and the loss between the RPN Regression layer, RPN CLS layer, Classifier Regression layer, and Classifier CLS layer was 0.369. Given more time, the team theorizes the model would preform better. If we had the chance to train on a higher number of ROIs, a shorter RPN stride, and smaller area for the ROIs, we think results would improve [5]. This reasoning comes down to the fact that the peak objects we are identifying are a lot smaller than the objects being identified in the original paper.

With this said, the intermediate models have preformed well, classifying peaks with a high (>99.9%) accuracy. This metric is slightly tainted though. Because the peaks in the image don't take up very much of the image, any region not classified as a peak is classified by the Faster-RCNN as background by default. What this ended up doing was created a large class imbalance of background to peak samples. So every time the model classified a region as background that was, the accuracy went up. This made the accuracy metric shoot through the roof. So this meant that the model was okay at identifying peaks, but regions classified as backgrounds with significantly lower probabilities were also peaks. More training to help the classifiers loss decrease further would help the model become more consistent.

Another thing that the team experimented a lot with was graphing and determining line

of sight. The current method only uses a 2D perspective, but when collecting elevation data for a bunch of peaks, a more accurate mesh of the landscape can be created which would help alleviate some of the problems we had with sampling rates. Below is sample of a of the 2D line of sight graph.Note the Haversine adjustment can be see at the bottom.

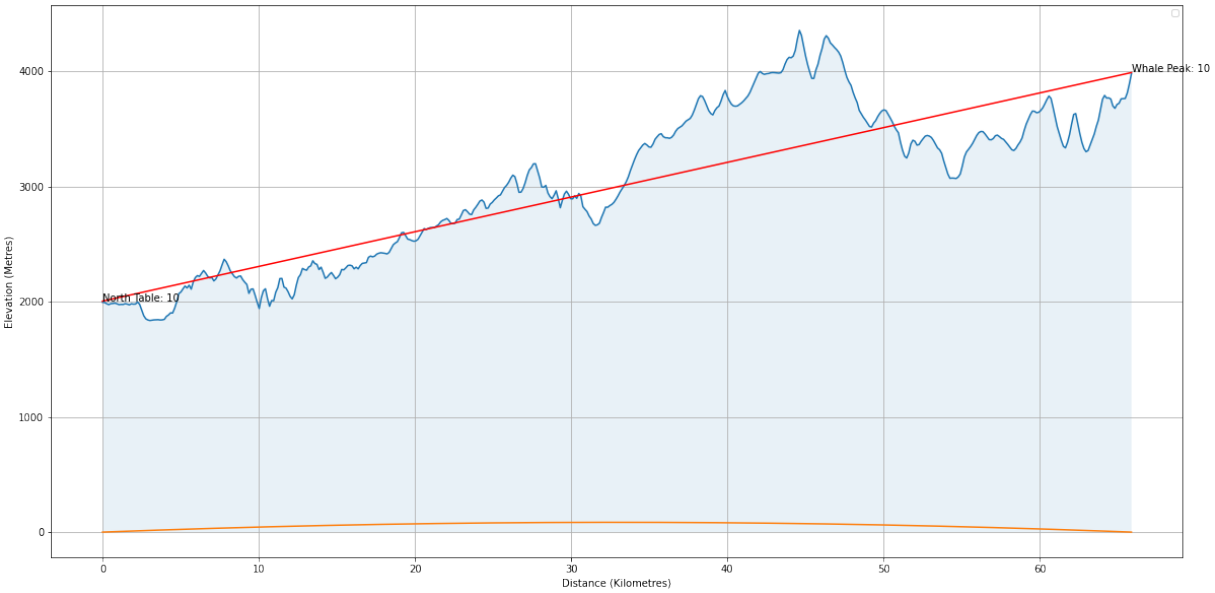

Figure 1: Line of sight to Whale Peak

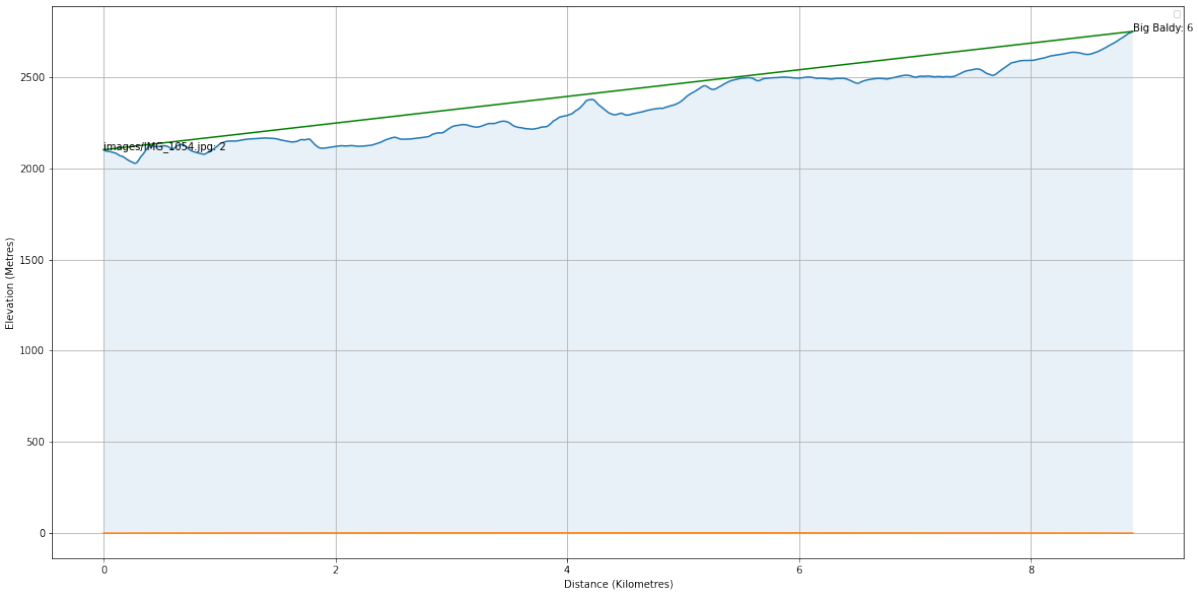

Figure 2: Line of sight to Big Baldy

Here is a topographical map with lines of sight plotted *figure 3* and a 3D mesh as described above sampling over Golden, CO. *figure 4*

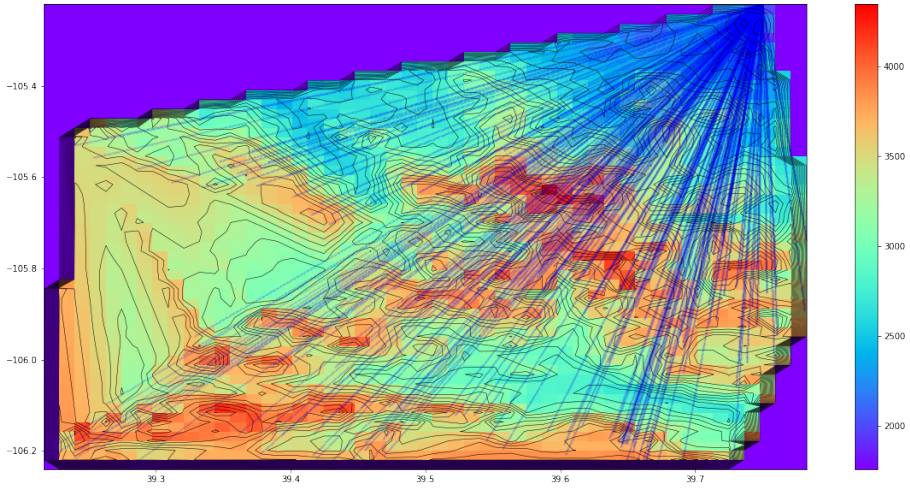

Figure 3: Line of sight on height map

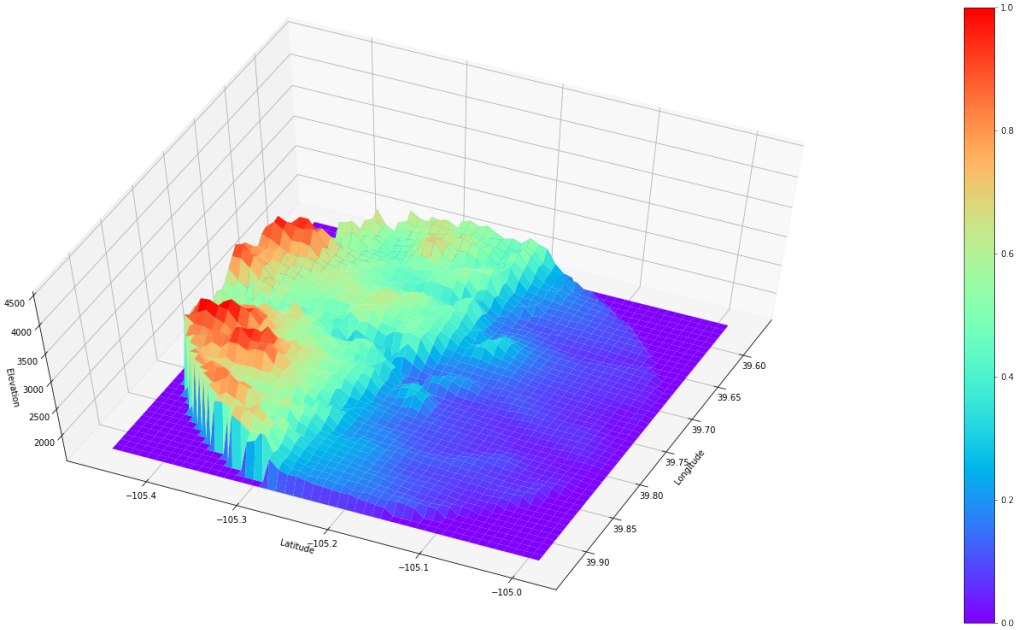

Figure 4: 3D line sight projection

When trying to project real world coordinates to the camera's perspective, there was a lot of trial and error before the team settle on precision being the problem. The first mistake was not adjusting the coordinate system with the Haversine equation. That improved the results, but the coordinates still weren't projecting correctly. The team tried 4 different

implementations taking different routes for projection, but the general formula was always the same and always ended up yielding the same answers which were all spaced within a few pixels of each other. In the end, this mattered less as the actual placement is done with the Faster-RCNN.

Examples of peak identification can be seen in 5 and 6.

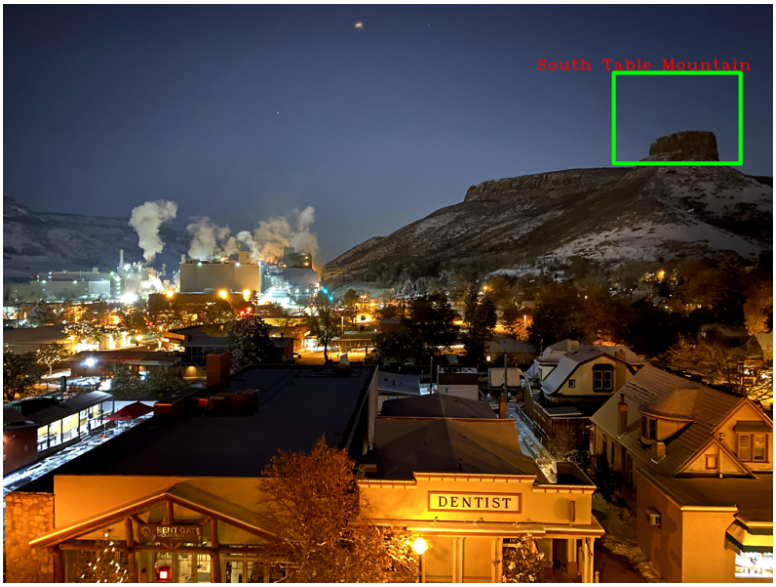

Figure 5: Successful identification of South Table Mountain using the Faster RCNN architecture

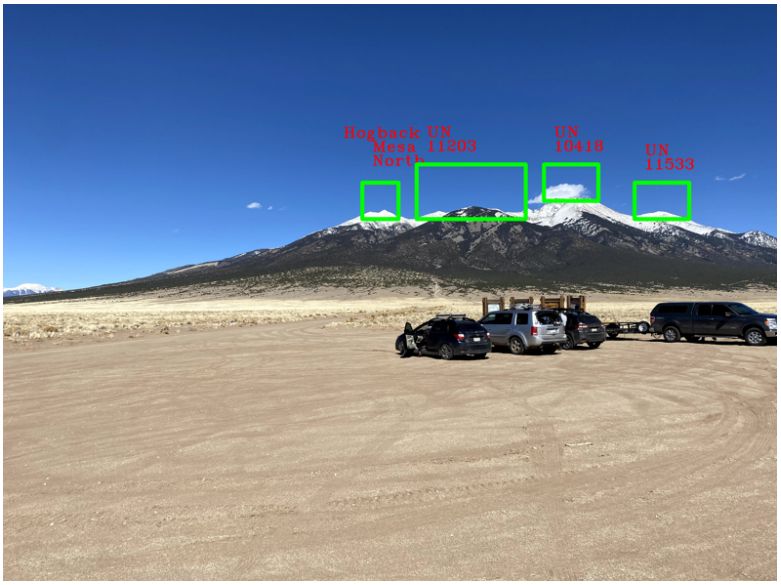

Figure 6: Successful identification of multiple peaks in one image

DISCUSSION

0.0.6 What is a Peak?

A big question is the definition of a mountain peak. Below are some simple definitions for terms most might consider homonyms.

Mount A mountain with exceptional prominence and isolation. Mounts are often found standing alone or are considerably taller than neighboring mountains.

Summit The Highest point of elevation on a mountain. A mountain can only contain a single summit.

Peak A local summit on a mountain. This is classified as a sharp point where the elevation is descending all around it. A single mountain can have several peaks.

The exact qualifications for these definitions are highly variable, and requirements change depending on the organization you ask. This means that some data has some peaks labeled and others didn't classify that peak as a peak. So with the definition being highly subjective, labeling the data is a little subjective. The primary solution was to normalize our labeling technique to have the center of the bounding box be the very tip of the peak. With this method, the threshold of peak detection probability can be tuned to accept mountain 'peaks' that are less commonly classified as peaks.

0.0.7 Ethics

Data ethics is a big concern in our approach because we need to extract the GPS data from each photo, so a user's privacy is something we need to keep in mind when we develop our product both in terms of getting the data, and handling the data after the application is done processing. The team wanted to ensure that the location data wasn't collected or stored so that a user's position could be tracked [3]. Within the collection process, the team tried

to include mountains from all around the world to try and make this application work for every population across the globe. One barrier that the team see as a problem is that not all photographs have the necessary information. In reality, more expensive equipment is needed to take a picture that works which makes the application not available to as many 'poorer' people and lends favor to the 'richer' audience.

0.0.8 Future Work

Improving the Faster-RCNN As the loss from our model continues to decrease, it is only evident that more training needs to be done. On top of that, more data high quality data would lend itself to further improving the model. As briefly mentioned earlier, the team thinks that increasing the number of ROIs, decreasing each ROI's area, and using smaller RPN strides are parameters that the team think will yield improvements.To add on, the team also thinks that transfer learning with VGG instead of Resnet50 might also given better results, but also ran out of time to test that.

More Precise Data The line of sight technique falls victim to the sampling rate. Increasing sampling increases the cost simply because it means more queries to API. This is also an overhead problem since all that extra data has to be processed. As mentioned previously, instead of using a single vector for line of sight, using a mesh created by querying all nearby peaks creates a more accurate 3D topographical mesh to check for interception.Besides that, as phones and cameras improve, more EXIF information will be available to be scraped (and will hopefully continue to increase in precision) which will help fix some of the homography problems.

Accounting for Missing Information In most cases, photos don't have all the information needed, but models like Google's PlaNet are aiming to solve this geolocation problem. The model is able to guess a location where a photo was taken without any other information

which could resolve the lack of available GPS information [**?**]

CONCLUSION

0.0.9 Our Approach

We decided to approach this problem by using two completely different methods than previous researchers. The first method was to use deep learning to train a Faster-RCNN model to identify the peak bounding boxes. The second method was to use homography and warping perspectives in order to identify the peaks. This combination of the two allowed us to label peaks in images if given enough information.

0.0.10 Strengths

We had a very creative way to train our machine learning model. When we used one image for training, we would warp that image into 10 different images in order to increase the robustness of our model during training.The model overall performed well with clear images and even images where the peaks were not on the horizon, covered with weather, or didn't conform to the 'ideal' peak shape. When tested, the pipeline was able to correctly predicted the location and names of these peaks.

These examples show the Faster-RCNN on challenging photos where peaks are covered in weather or not on the horizon line.

0.0.11 Weaknesses

When we used the machine learning method, we didn't have enough data in order to adequately train the model because, in order to generate data, we had to label every peak by hand. The current model has been training on 12000 labelled images. It is not a very efficient process, but the results from the current dataset the team has built have yielded very

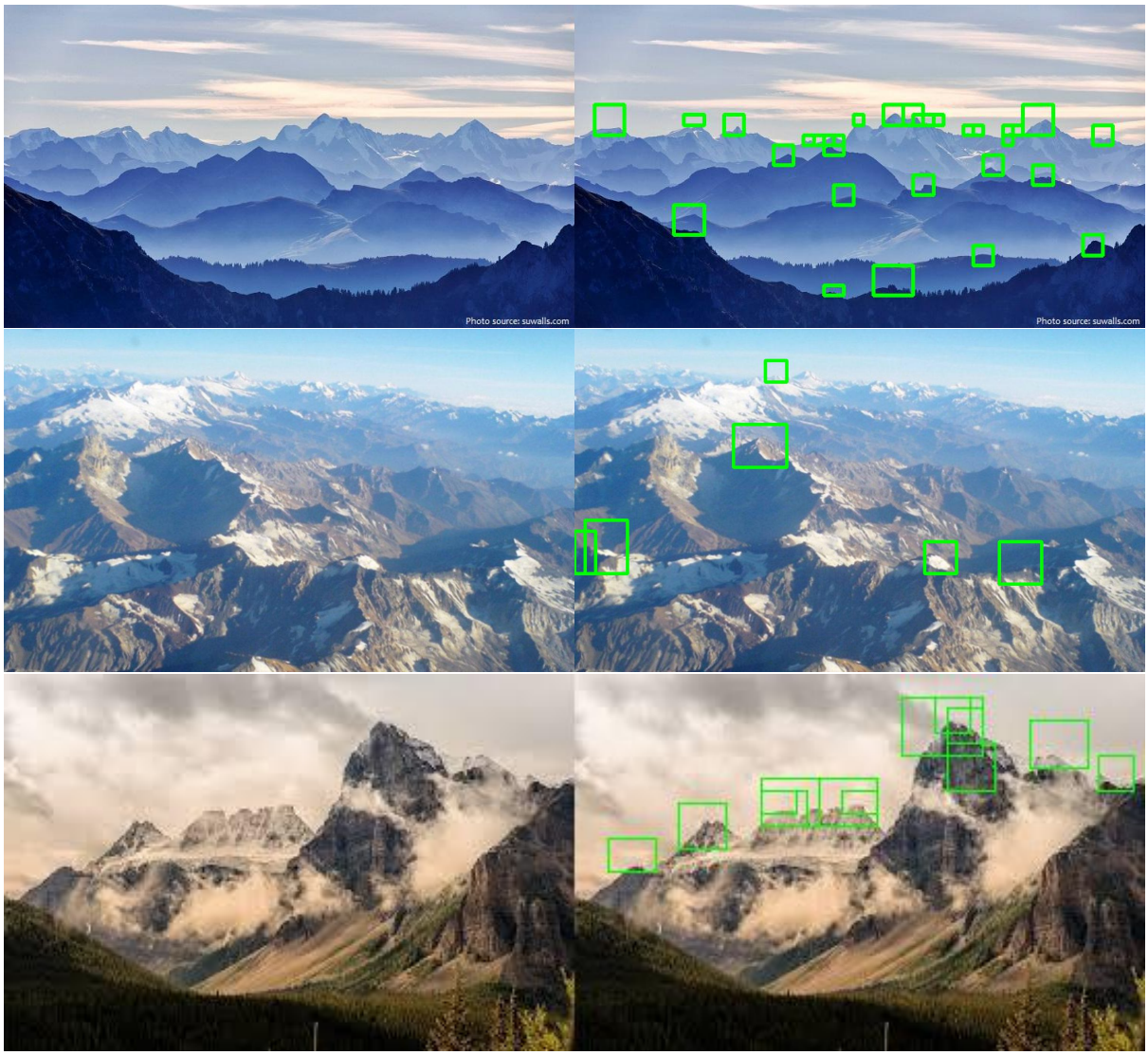
Photo source: suwalls.com
Photo source: suwalls.com

promising results. We also faced scaling issues when training the data. Not enough full sized images could be found, so the team resulted to images with less quality that didn't scale to the 516x516 input size very well. With different data, the class imbalance problem discuss could also be fixed which would help with better classification of the ROIs. With a better model, the false positives in the images could be corrected allowing for easier matching of labels to peaks.

Bibliography

[1] Mountain classification, the uiaa is everything mountaineering.

[2] Rocio Nahime Torres Darian Frajberg, Piero Fraternali. Convolutional neural network for pixel-wise skyline detection.

[3] Cheryl Baker Lionel Prat, NhienAn LeKhac. Mapexif: an image scanning and mapping tool for investigators.

[4] Tomaž Podobnikar. Detecting mountain peaks and delineating their shapes using digital elevation models, remote sensing and geographic information systems using autometric methodological procedures.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks, 2015.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[7] you359. you359/keras-fasterrcnn.