CLASSIFICATION OF ELECTRIC TOWERS AND DETECTION OF LINE'S

Rahul Ramesh Kumar

New York University

ABSTRACT

Classification of Electric Towers has been an area of research for some time and countries now days want to replace manual tower inspection with Automatic tower inspection. Hence such a problem poses lot of interesting research in Computer Vision. Currently there is a lot of literature that is around for classification of Electric Towers, however there is very few work that aim to extract the line segments that make up an Electric tower.

The paper describes a method to classify Electric Towers from Images generated by Aerial inspection vehicles. It also aims at extraction of line segments from these classified images to locate the lines that make up the Electric Tower. The image recognition uses IBM Watson Visual Recognition to train and classify these towers. One area where this can be useful is to make a classifier. The power of IBM Visual Recognition is realized when a large dataset of images are presented to it and mathematical tools such as ROC and Confidence intervals can be plotted to continuously improve the classifier and fine tune it to produce better results. The paper also provides a working implementation that extracts Line-segments that are part of Type 1 Towers .

***Index Terms*—** Recognition, Classification

1. INTRODUCTION

The paper describes a method to classify and detect Type 1, Type 2 and Type 4 towers from Images. The image recognition uses IBM Watson Visual Recognition to train and classify these animals.

A possible usecase of such a detection can be in scenarios where Electric power companies would like to automate detection of a electric tower and find failures /defects without having to manually inspect. Such a software system can greatly benefit by reducing cost and reducing time taken for inspection and most importantly reducing risk of lives of Line men.

Section 2 describes the Selection criteria used to train the Tower images that are fed to the IBM Watson Visual Recognition system . Section 3 describes the Classification types and ther results achieved. Section 4 discusses about the Line detection algorithm used and strategies used to extract the Line segments that make up the Electric Tower. It also shows results and discussion about improvements. Section 5 discusses Gap analysis performed with a similar research paper[3].

2. DATA SET AND SELECTION CRITERIA

Each tower type is defined as a class and each image was selected using features of the tower like Top shape, Bottom base size and external features to allow the classifier to learn as much as possible from each image. Following are the criterias.

- Images of Front, Bottom and Top View.
- Occlusion - Only certain parts of Images are shown.
- 32 Images.
- Illumination - Based on different lighting.
- Based on Image Quality.
- Images where they can interpreted as different structures.
- Images where they are present with a background like Sun, grasses.

3. CLASSIFICATION AND RESULTS

Using IBM Visual Recognition API, an image classifier was created that was trained to classify three classes of images :

- Type 1 Tower (275 kv).
- Type 2 Tower (220 kv).
- Type 4 Tower (415 v).

The positive training set comprised of 25 images for each animal. The negative training set was a bag of 40 images that compromised of random images that were picked up from image-net.org

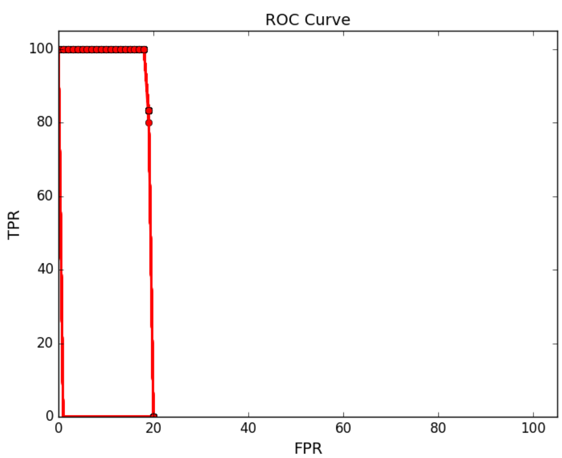
3.1. Test Run

A positive Test was performed by selecting randomly 6 images from the test set from positive samples for Type 1 tower . Similarly a negative test was performed by selecting randomly 6 images from the test set of negative samples that were randommly picked from ImageNet.

In order to obtain confidence intervals, the test was performed 50 times using the same set but by randomly selecting them using Python random api.

3.2. ROC Curve

Following is the image for ROC Curve obtained for the 50th iteration. The ROC curve plot Images for all the iterations are placed in a zip file for viewer who is interested to analyze individually.



(1) ROC Curve- 50th iteration

3.3. Confidence Intervals

The confidence intervals was obtained for each iteration for both False Positive Rate and True Positive Rate. Also the final confidence rate was also calculated and shown below. Iteration is defined as range function from 0 to 1 in steps of 0.5

Confidence Interval TPR = 93.59, 95.28
Confidence Interval FPR = 3.91, 5.60

Table 1. 80 Percent Confidence Interval for all 50 iterations

Threshold	TPR	FPR
0.00	100.0, 100.0	100.0, 100.0
0.05	100.0, 100.0	100.0, 100.0
0.10	100.0, 100.0	0.0, 0.0
0.15	100.0, 100.0	0, 0.0
0.20	100.0, 100.0	0.0, 0.0
0.25	100.0, 100.0	0.0, 0.0
0.30	100.0, 100.0	0.0, 0.0
0.35	100.0, 100.0	0.0, 0.0
0.40	0.0, 0.0	0.0, 0.0
0.45	100.0, 100.0	0.0, 0.0
0.50	100.0, 100.0	0.0, 0.0
0.55	100.0, 100.0	0.0, 0.0
0.60	100.0, 100.0	0.0, 0.0
0.65	100.0, 100.0	0.0, 0.0
0.70	100.0, 100.0	0.0, 0.0
0.75	100.0, 100.0	0.0, 0.0
0.80	100.0, 100.0	0.0, 0.0
0.85	100.0, 100.0	0.0, 0.0
0.90	100.0, 100.0	0.0, 0.0
0.95	83.18, 83.35	0.0, 0.0
1.00	0.0, 0.0	0.0, 0.0

4. LINE DETECTION

After the Images are classified , its possible to extract features such as Line Segments from the Image. The Line segments are important here to find since they make up the tower. Once the line segments are extracted , its easy to find features that are common to a Type 1, Type 2 or Type 4 Tower. The Line Segment Detector that is used in this implementation is from [4]. The implementation is written in C and works well on a Linux Environment with no dependency on OpenCV libraries. The Line Segment Detector outputs all LineSegments detected with their points on the image and their length. After than an algorithm is employed to detect parallel lines or almost collinear lines. Below section covers the Algorithm, Results and Improvements.

4.1. Problem

The aim of the paper involves in identifying the Line segments that make up the Electric Tower. Each type of Electric tower have their own structural characteristics and hence we develop one that is aimed at detecting Type 1 towers. One of the sub problem's that is identified here is to extract parallel line segments. Once the parallel lines are extracted, then we apply a threshold based checking to filter out parallel lines that are close to each other. Another sub problem is to merge line segments that are very close such that they can be classified into one line. Here we apply suitable thresholds in angle of inclination to counteract for any noise developed during image analysis by LSD [4]. Below section deals with these problems in more detail.

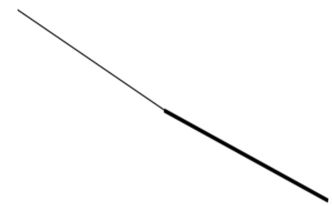
4.2. Algorithm

- Run the LSD detector and calculate length of each Line Segment detected .
- Sort all line segments based on the length (A quick sort implementation in C was used).
- Find All Parallel Lines (By Theta Binning) for each Line Segment.
- Calculate Slope of each Line using Slope Formula and binning it to a 5 degree variance.
- Calculate Perpendicular distance between each pair of Line Segment.
- Find OverLap Distance - Finds if two parallel line segments have any overlap and by how much. Using slope and a reference Axis, the distance is computed by projecting to X or Y axis.
- Now based on OverLap Distance and Perpendicular distance , The line segments can be classified into two types
  - **Class 1** All most similar line which are collinear and parallel.
  - **Class 2** All most parallel and close.
  - if ((absolute difference in slope(LS1) and slope(LS2)) less than t1) and (distance between them less than t2) and (overlap between them less than t3) then it can be classified as one single line segment. (Class 1)
  - if ((absolute difference in slope(LS1) and slope(LS2)) less than t1) and length of line is not less than t4) then it can be classified as parallel line segment. (Class 2)
- Merge all Class 1 lines that have a very less perpendicular distance and overlap distance with in a threshold t3. The Merge is done such that once a new merged line is created , then every other line that is parallel to this line needs to check if it can be grouped together.

4.3. Results

Once the algorithm runs, it outputs for each Line Segment Pair its classified Line Segments into a .eps file which can be converted to .pdf file. We present some of the results here.

Figure 2 shows how lines are classified correctly since both are almost collinear. Using the coordinates of both lines



(2) Rightly Classified into Class 1

we can draw a line over it to produce a bigger line segment thus avoid segregation.

Figure 3 shows how lines are classified correctly since both are parallel and have large overlap. By manipualating the distance , features can be produced that is distinctive to each Tower design. The program must be able to take this configuration in run time and accordingly run the algorithm to produce effective results

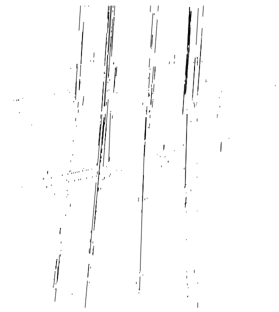
(3) Rightly Classified into Class 2

Figure 4 shows the output after running the merge on the Class 1 lines.

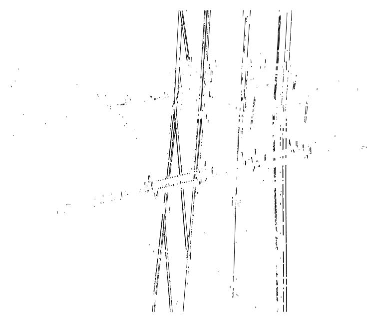
Figure 5 shows the output after running class 2 based thresholding. Here we have obtained the entire tower line segments in one image. This is one of the main advantages of slope binning.

4.4. Performance

For the test image we found 10504 line segments with a resolution of 7152 * 5368. The entire program took around 10



(4) Merge output of entire tower



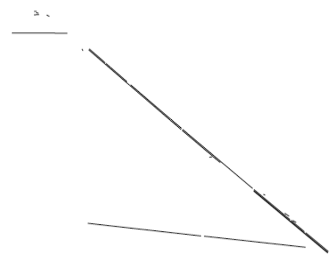
(5) Output after class 2 thresholding, we can see the entire tower

seconds to execute, this also includes the time taken to create the eps file. Let n be the number of line segments found, then the time complexity of the program is O(n*n). Extra memory used to Hold overlap distance, perpendicular distance And slopes hence the space complexity is O(n*n)

4.5. Discussions and Improvements

One of the approaches to solve the slope isolation issue was Slope binning and it has helped a long way in the further part of the algorithm that depends on threshold checking. Below Image 6 shows one issue if binning is not performed.

Another issue is with respect to the inbuilt Guassian filtering that is used by [4] and by modifying the sigma value we can obtain better results since such calculations can be affected by noise/blur in image.



(6) Too much seperation - we can avoid by binning

5. GAP ANALYSIS

A literature heavily surved was [3] which is neural net based classifier. Here the aim is to compare between an existing or somewhat similar implementation and discuss about potential drawbacks in the current implementation and how it can be improved.

- A HOG descriptor is used to describe shape of the electric towers and then binned for 9 orientations. Each bin stores the gradient at that angle which is more powerful than calculating individual slopes since it can produce more false positives. A somewhat similar implementation is underway to bin based on the slopes that will reduce the false positives.
- One more step is that all oriented histograms are normalized thereby it allows invariance to any photometric effects which clearly had effects in our algorithm that resulted in slope calculation issues.
- A 3 layer Neural Net is employed here to classify the towers, similary an approach such as this could have been employed to classify the line segments into Class1 and Class2. However it would require training the classifier which is an added effort towards development.

6. CONCLUSIONS

Overall the detection system is able to identify the target segments of the Electric Tower. The algorithm can be fine tuned further as we discussed in the improvements section and gap analysis section. We have shown here that a basic understanding of geometry can be used to solve a Computer Vision problem without using too much OpenCV library functions.

The main reason to implement such as system is to promote more research in this area and also help reduce false positives in Classification based systems. The extracted line segments can be used to feed the Cloud based Classifier to produce better results. The hope is that this paper can make way to automated Electric tower and Wire inspection system and help reduce costs and make it safer for Line men who risk their life for us. The entire program is hosted on github [5] which is written in C.

I would like to thank Dr. Sharath Pankanti and Dr. Nalini K. Ratha for providing feed back during the course of the project and also introducing me to various topics in Computer Vision.

7. REFERENCES

1. http://visual-recognition-demo.mybluemix.net/
2. image-net.org
3. A Supervised Approach to Electric Tower Detection and Classification for power Line Inspection
4. LSD: a Line Segment Detector 2012
5. https://github.com/rahulkalpsts107/cv-LinesExtraction