The Ohio State University

HackOHI/O 2020

American Electric Power Challenge: **Use of computer vision to better map and understand the distribution network and the status of the equipment.**

AEP has begun using ground-based and drone photography to inspect and evaluate our equipment. Currently we own approximately 90 drones and have 120+ pilots. Computer vision offers an opportunity to scan enormous amounts of data and alert personnel to issues requiring attention.

Computer vision is an interdisciplinary scientific field that deals with how computers can gain a high-level understanding of digital images or videos. From the engineering perspective, it seeks to understand and automate tasks that the human visual system can do. Computer vision tasks include methods for acquiring, processing, analyzing, and understanding digital images and extraction of high-dimensional data from the real world to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means transforming visual images (the input of the retina) into descriptions of the world that make sense to thought processes and can elicit appropriate action.

Challenge: AEP provides power to millions of customers every day, with a network of underground lines and above ground distribution poles stretching over 200,000 miles long. Maintaining such a vast network is no small effort, and AEP has been experimenting with computer vision to better map and understand the network. Build an application that can yield meaningful results using the provided images, metadata and utility pole object detection model.

Challenge Details: In this challenge you will be given access to around 130 images of utility poles with accompanying metadata, as well as access to a utility pole object detection model via an API endpoint hosted in a cloud environment. The challenge is to use the data and the model to build an application that can take in the data and leverage the model to create meaningful results. Ideas might include a catalog of poles and accompanying data, a map of all poles and images, or anything else you feel could be useful.

For bonus points, we will provide the gps coordinates of the poles and the azimuth of the camera to the detected pole. These pieces of date are calculated from the other provided metadata provided with the images. The bonus challenge is to not use the provided azimuth and or pole coordinates, but calculated ones instead for your application.

Prize: $200 Amazon Gift Card

Judging Criteria:

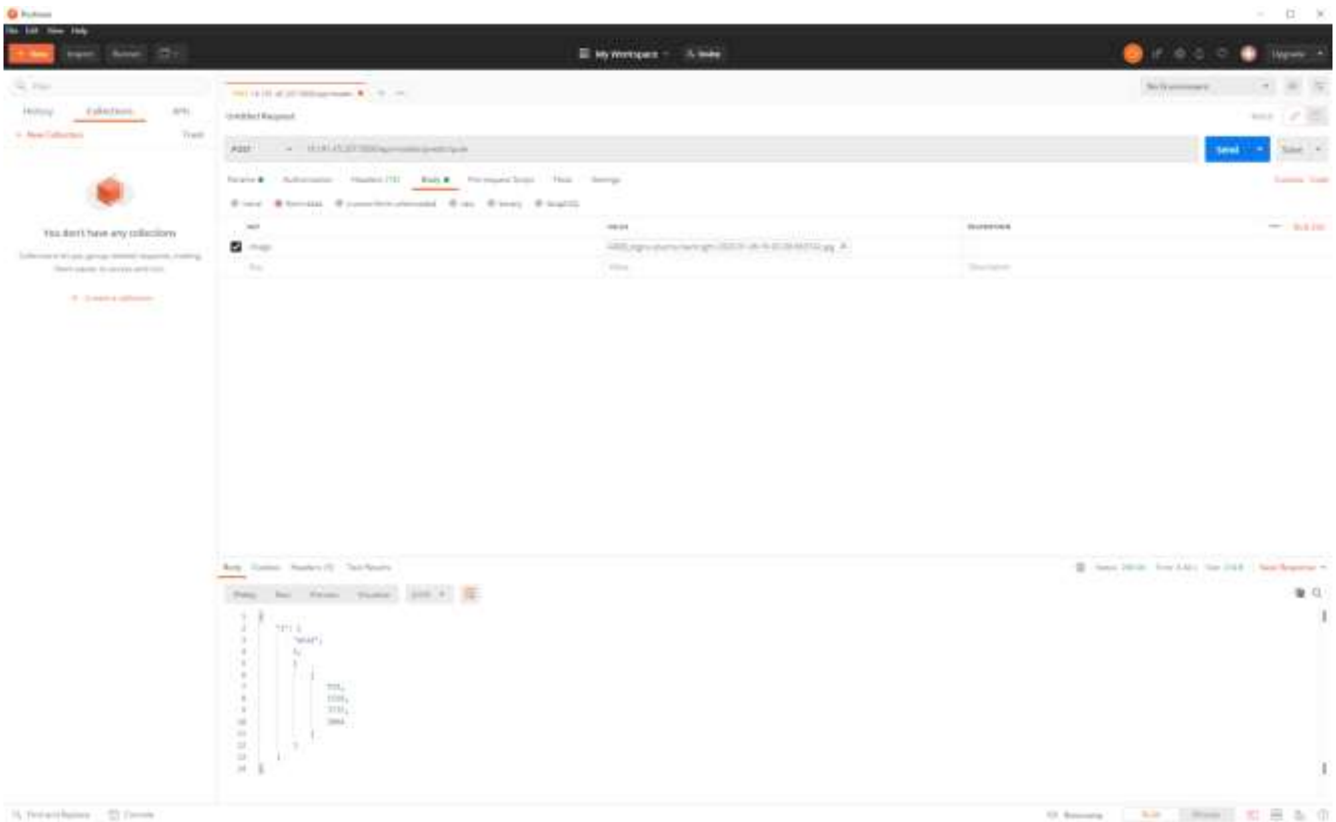
a) **Originality** - Does it do something entirely novel, or at least take a fresh approach to an old problem?
b) **Execution** - Is the application usable in its current state? Is the user experience smooth? Does everything appear to work? Is it well designed?
c) **Usefulness** - Is the application practical? Is it something people would actually use? Does it fulfill a real need people have? Did it produce new and useful data?
d) **Presentation** - How well was the project presented? Did it make the application more compelling? Did it give a good idea of its purpose?
e) **Learning** - Did the team stretch themselves? Did they try to learn something new? What kind of projects have they worked on before?

Model API

The model API is a one route api that will allow you to leverage our object detection model to detect power poles in images. To use the api you will send a copy of an image to the below route and receive a list of all detected poles and their bounding boxes within the image

Request:

Send a POST request to 18.191.45.207:5000/api/models/predict/pole with a form-data section in the body key “image” value the read in image data



Response

You will receive json back, it will be empty if no poles are found. If one or more poles are detected you will receive a list containing the bounding box information for each pole. The bounding box arrays will contain four pixel values [y1, x1, y2, x2] of the top right corner and the bottom left corner pixel coordinates of the bounding box. For images it is important to note that the top left corner of the image is (0,0), 1000 pixels in the x direction is 1000 pixels from the left hand side 1000 pixels in the y direction is 1000 pixels from the top of the image.

```
{
    "1":
 [
        "pole",
        1,
        [
            [
                555,
                1518,
                3735,
                2064
            ]
        ]
    ]
}
```

Metadata

S3 Bucket - https://osuhackathondata.s3.us-east-2.amazonaws.com/

Here is an example of the metadata you will have available for each image.

S3 bucket will contain two file per image. First the .jpg file containing the image and a second .json file containing the metadata. The file will be named the same, for example file1.jpg and file1.json.

JSON Metadata


```
{
   "pole":
 {
      "coordinates":
 [
         -83.0454061204676,
         39.97812820187546
      ]
   },
   "image":
 {
      "width":
 3000,
      "height":
 4000,
      "heading":
 3.4146,
      "latitude":
 39.9778982,
      "longitude":
 -83.0452891,
      "imageUrl":
 "path to the image",
      "azimuth":
 338.5502666666667,
      "FOV":
 58,
      "yaw":
 55,
      “Type”:
 “pole”
   },
   "Esri
 Data":{
      "assets":
 {
         "transformers":
 0,
         "crossarm":
 7
      }
   }
}
```


Metadata description

Pole[coordinates] is a list of the longitude and latitude of the detected pole with in the image

Image[width] is the width of the image in pixels

Image[height] is the height of the image in pixels

Image[gps][heading] is the heading of the device when the image was taken. This is in degrees from true north

Image[latitude] is the latitude of the camera when the image was taken

Image[longitude] is the longitude of the camera when the image was taken

image[imageUrl] is the path to the image with in an s3 bucket that you can use to download the image

Image[azimuth] The degrees from true north of a line starting at the camera that intersects with the gps of the pole.

Image[fov] the horizontal field of view of the camera

Image[yaw] the vertical offset of the camera

Esri[assets] a dictionary with keys as different types of assets on the pole and the value being the total number on the pole