Project 5 Pollution Vision

This manuscript (permalink) was automatically generated from Saran-Wang/dsproject@d395f39 on December 6, 2020.

Authors

- **Shiyuan Wang**
  · Saran-Wang
  Department of Civil and Environmental, University of Illinois

- **Weiqi Ni**
  · weiqini
  Department of Civil and Environmental, University of Illinois; Department of Environmental and Resources, Zhejiang University

- **Gemma Clark**
  · 441gclark
  Department of Civil and Environmental, University of Illinois

- **Xueao Li**
  · XueaoLi
  Department of Civil and Environmental, University of Illinois

Abstract

Introduction

Literature Review

There are many studies using digital camera and advanced algorithm to estimate the concentrations of Particulate Matters. Hong et al. [1] developed a novel method of predicting the concentrations and diameters of outdoor ultrafine particles using street-level images and audio data in Montreal, Canada. Convolutional neural networks, multivariable linear regression and genralized additive models were used to make the predictions.

Exploratory Data Anlysis

1. Variables Explanation

Table 1: Variables Explanation

Data Fields	Explanation
Temp(C)	ambient temperature
Pressure(kPa)	air pressure
Rel. Humidity	relative humidity
Errors	if the air measurement equipment has error during sampling (0=no)
Alarm Triggered	if any instrumental warning shows during sampling (0=no)
Dilution Factor	an instrumental parameter (should close to 1)
Dead Time	another instrumental parameter (ideally close to 0)
Median, Mean, Geo. Mean, Mode, and Geo. St. Dev.	parameters describe particle sizes, which can be ignored
Total Conc.	an output variable from the instrument that should not be used
image_file	the visual information of the traffic condition, corresponding to an image in the “frames” directory
Wind_Speed	the wind velocity during sampling
Distance_to_Road	the distance between camera and road
Camera_Angle	the angle of incidence between the camera and the road
Elevation	the elevation between the camera and the breathing zone
Total	the total measured particle number concentration (# / cm3) This is the dependent variable

2. Data Cleaning

- Delete the useless columns in the dataset
- Delete the rows with equipment error during sampling

3. Visualization of the distributions of varibales

Figure 1 shows that “Wind_Speed”, “Camera_Angle”, “Distance_to_Road” and “Elevation” are all in discrete distributions, while “Temp(C)” are in continuous distribution. “Pressure(kPa)” has four clusters. It should also be noted that the “Dead Time” almost shares the same distribution as “Total”.

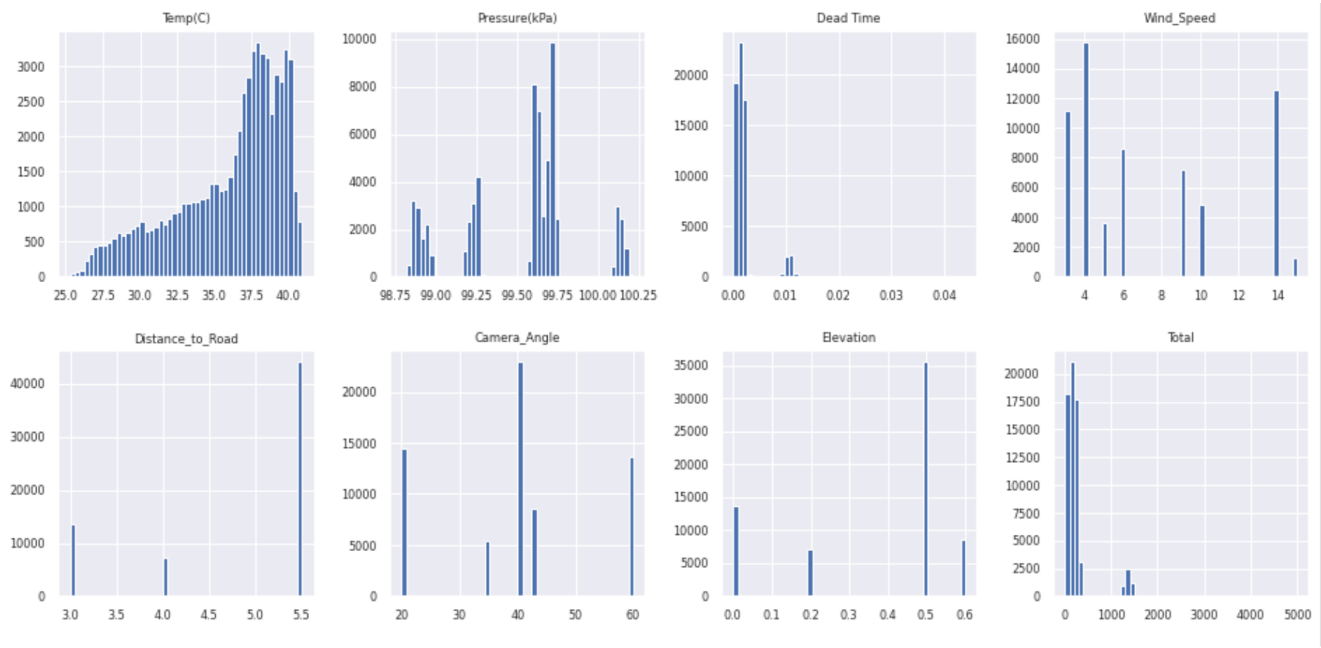

Figure 1: Variables Distribution

4. Correlations among variables

From the correlation map 2 we could see that "Dead Time" are extremely correlated with "Total", with a coefficient of 1, followed by "Camera_Angle", "Pressure(kPa)" and "Distance_to_Road", with coefficient of 0.52, 0.49, 0.44 respectively. Here you may be curious why "Dead Time" could be so closely related to "Total", and there is one possible explanation: Actually, "Dead Time" is an instrument parameter, and if there are more PM concentrations in the air, the instrument need more time to process, and vice versa.

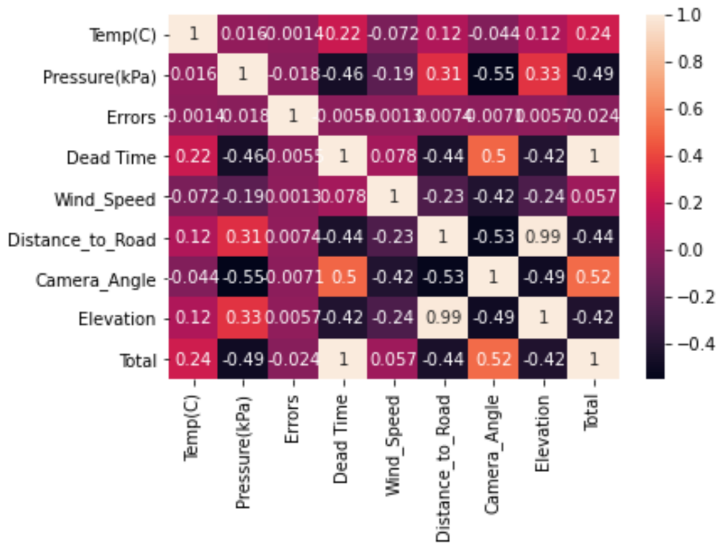

Figure 2: Variables Correlations

Model

Shiyuan's Model

My model setup splits into two part, the first is image data extraction, the second is the selection of appropriate model to fit this dataset.

Image Extraction

First I want to digitize images by extracting image features, there are mainly 6 features I want to extract: RGB, image luminance, image contrast, image entropy, transmission and amount of haze removed and number of cars on streets.

1. RGB

The RGB color model is one of the most straightforward parameters describing an image. Intuitively, in this case, we may expect more blueness and greenness if the PM concentrations are low since the color of tree and sky would be brighter when the air conditions are good. For each image, after deriving the RGB of each pixel, we take the average of them, and then divide each value by 255 to normalize it. The figure below 3 shows the distributions of RGB in this dataset. We can see that they are nearly normally distributed with mean 0.45, 0.55 and 0.35 respectively. For blueness, we could see a second peak at around 0.42.

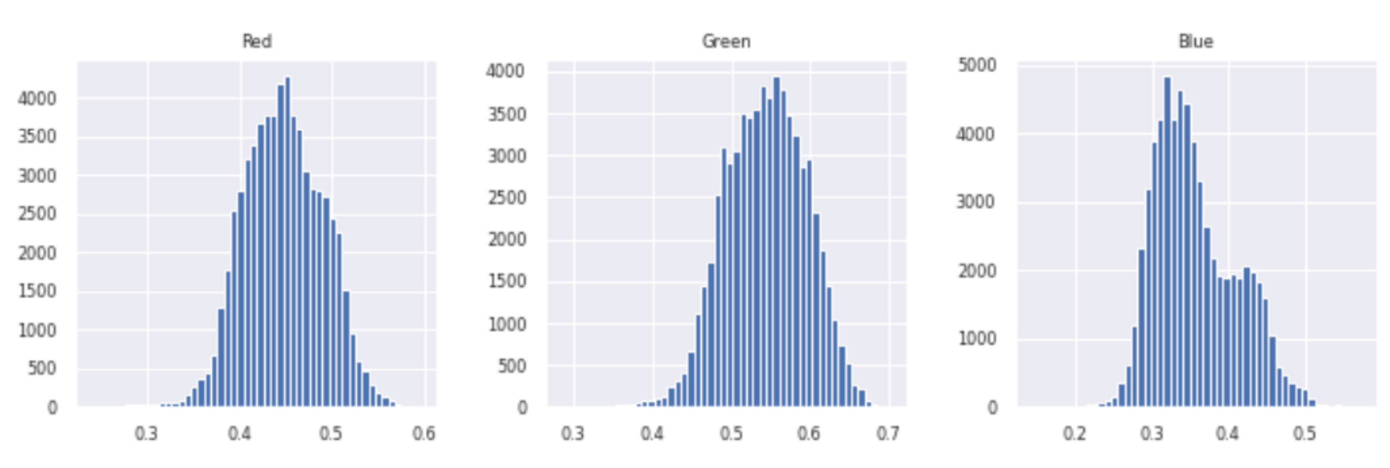

Figure 3: RGB Distribution

2. Luminance

Like RGB, luminance is also a very basic parameter describing an image, which could be an indicator of how bright the image will appear. The luminance of each image is calculated by taking the average of the luminance intensity of each pixel. From figure 4 we could also see that it's also normally distributed with a mean of around 130.

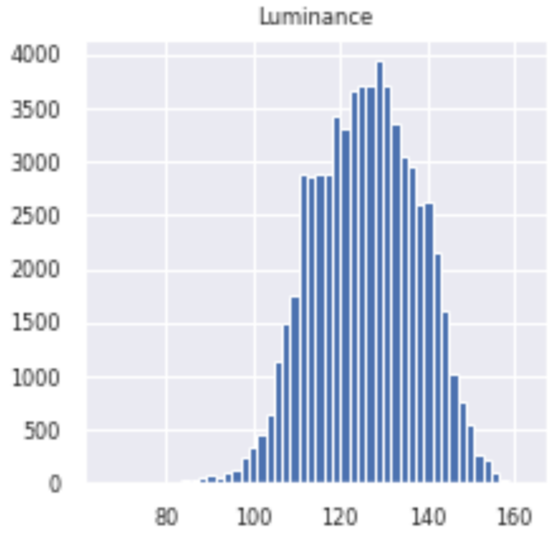

Figure 4: Luminance Distribution

3. Contrast

The image contrast is defined as the difference between the max and min luminance intensity of an image. Study [2] shows that the higher the PM concentrations, the lower contrast would be. It makes sense since the image would become vague and lighter when there are more particulate matters in the air. And often, one image would have pixels with the highest intensity of 255, as well as the lowest intensity of 0. Therefore, we can't see much difference if we want to derive the absolute contrast, since it would be 1 for most of those images. Therefore, we use root mean square of image intensity to describe image contrast.

$$Absolute_Contrast = \frac{I(i_{max}, j_{max}) - I(i_{min}, j_{min})}{I(i_{max}, j_{max}) + I(i_{min}, j_{min})} \tag{1}$$

$$RMS_Contrast = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(I(i,j) - avg(I))^2} \tag{2}$$

where I(i,j) is luminance intensity at (i,j) pixel.

From figure 5 we could see that the distribution is a little bit right-skewed with a small peak at around 35, and a larger one at around 50.

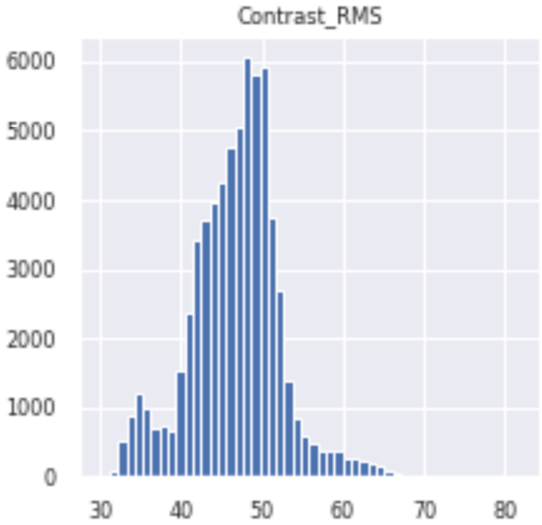

Figure 5: Contrast Distribution

4. Entropy

Image entropy is a statistical measure of randomness that quantifies information contained in an image. Usually, an image would lose its details with the increasing PM concentrations, and the image entropy will decrease as a result [2] . To do the calculation, I first converted the original RGB image to grayscale image and then used a module within python: *skimage* to calculate image entropy directly. The example code is shown as below:

```
colorIm =
        Image.open('../input/pollutionvision/frames/frames/video06082020_0.jpg

greyIm = colorIm.convert('L')
ImContrast = skimage.measure.shannon_entropy(greyIm)
```

Figure 6 is the original figure and figure 7 shows its entropy.

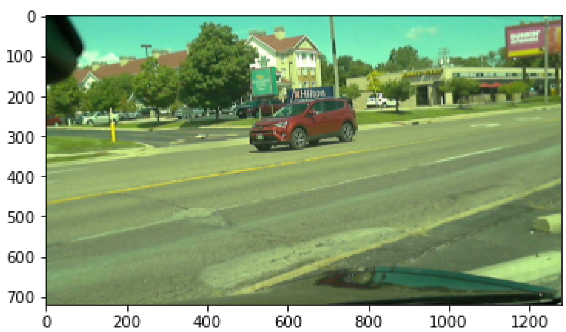

Figure 6: Original Image

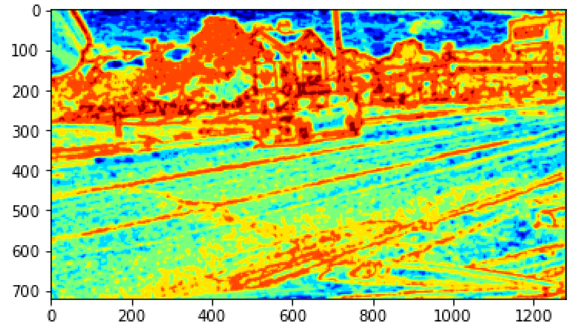

Figure 7: Entropy

5. Transmission and amount of haze removed

6. Number of cars on streets

Also, I take the number of cars on streets into account. Intuitively, the more cars on the street, the higher PM concentrations would be. There is a very efficient library in python called *cvlib*, within which a function called *object_detection* could detect the number of different objects appearing on an image. The example code is shown as below:

```
im =
        cv2.imread('../input/pollutionvision/frames/frames/video06082020_0.jpg
bbox, label, conf = cv.detect_common_objects(im)
output_image = draw_bbox(im, bbox, label, conf)
number_of_car = label.count('car')
```

As we can see in figure 8, the left image has 2 cars on street, and we can detect exactly two cars; while figure 9 has no car on street but 5 cars parking in the parking lot, and we could detect 5 cars.

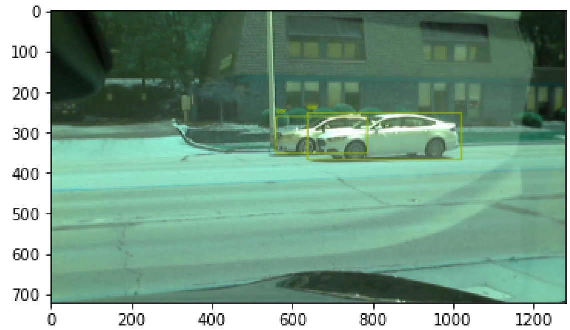

Figure 8: Two cars detected

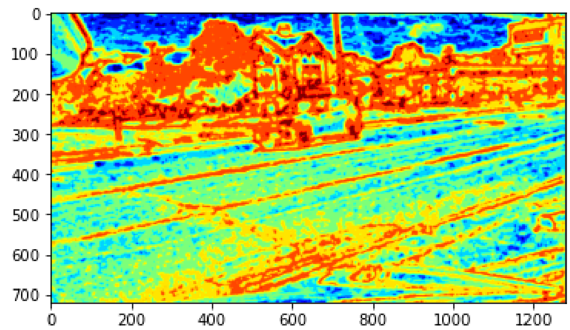

Figure 9: Five cars detected

Here comes the problem, this function could only detect the number of cars appearing on an image but can't identify which is in motion. But the moving cars are actually the ones which contribute to PM concentrations at the very moment. However, in this case, I just keep the original detection results, since if there are more cars in the parking lot, I just assume it's a traffic busy day, on which the PM concentrations would be higher than normal days.

7. Correlations among variables

Here we plot out the spearman correlations among those features in figure 10, the last column shows the spearman correlations between each feature and the Total PM concentrations. As we can see, the dead time, which is an instrument parameter, is closely correlated with PM concentrations, followed by Pressure, RGB, luminance and temperature. However, since the dataset is rather complicated, the correlations may mean nothing. Actually, the different combination of different features may have various impacts on the results of our model. And the correlations just provide us with a straightforward perception.

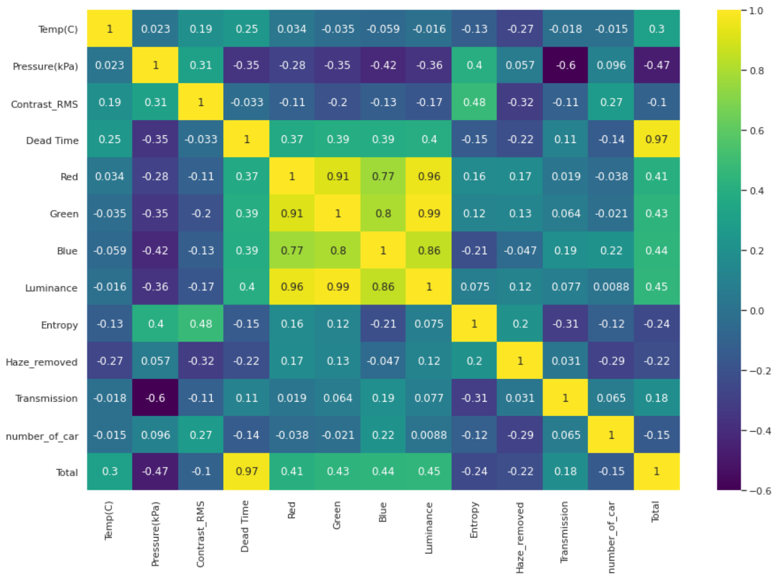

Figure 10: Variables Correlations (include digitized image data)

Model Selection

As mentioned before, we can't select the features barely based on their correlations with PM concentrations, since I have both numerical data and digitized image data, which could be very complicated. Therefore, I selected different combinations of features and run the model several times to select the one with best performance. At first, I tried the Neural network, but it doesn't do well in this dataset, with an MSE of around 800. Then I calculated the model accuracy using *cross_val_score*, and then I switch to random forest, which gives me an accuracy of 0.997. With the help of * GridSearchCV*, I could decide on the parameters for random forest: `RandomForestRegressor(max_depth=20, n_estimators=1000, random_state=3)`. With those parameters, I could get an RSME around 11.

Gemma’s Model

Weiqi’s Model

Xueao's Model

Conclusion

References

1. **Predicting outdoor ultrafine particle number concentrations, particle size, and noise using street-level images and audio data**
   Kris Y. Hong, Pedro O. Pinheiro, Scott Weichenthal
   Environment International (2020-11) https://doi.org/ghnh6n
   DOI: 10.1016/j.envint.2020.106044 · PMID: 32805577

2. **Particle Pollution Estimation Based on Image Analysis**
   Chenbin Liu, Francis Tsow, Yi Zou, Nongjian Tao
   PLOS ONE (2016-02-01) https://doi.org/ghnjkc
   DOI: 10.1371/journal.pone.0145955 · PMID: 26828757 · PMCID: PMC4734658