CS230

Using Satellite images to determine AQI values

Simon Bumm
Management Science and Engineering
Stanford University
simon.bumm@stanford.edu

Michael Chen
Civil and Environmental Engineering
Stanford University
mike0607@stanford.edu

Sean Hsu
Material Science and Engineering
Stanford University
khhsu@stanford.edu

Abstract

In this project, deep neural networks are used in an attempt to predict Air Quality Index (AQI) differences for pairs of satellite images where images are taken froin the same location at different times

1 Introduction

The goal of our project is to develop a deep neural network to infer AQI (air quality index)[1] based on a satellite image by providing a baseline satellite image and AQI data in the same region. Since today, satellite images are more broadly available than AQI readings, having a model that can determine AQI based on just a satellite image can significantly help people to monitor air quality in their region without the need for a physical measuring station. Such a prediction model could be used e.g. in regions that do not have an air quality reading station set up. With a one-time measurement, residences of the area can receive information about the air quality in the area. This can be especially helpful to the regions that have a long running air quality problems, such as China.

2 Related work

There is plenty of literature on using satellite to predict a plethora of different outcome variables. However, the majority of said work falls under the category of image classification rather than regression. We have picked out a selection of work that most closely relates to our goal and compared them in the following paragraph.

Neal Jean et al. [2], utilized a multi-step transfer learning process to identify and extract nighttime features from satellite images to predict poverty in certain regions. Their setup is similar to our project since extracting night light imagery features depends on very small variations within the picture, which is the goal of our project, too. Additionally, transfer learning is used in this research to help with feature extraction and to compensate for incomplete survey data, similar to the research that detects oil spill. Although we were able to obtain complete AQI data, we believe that transfer learning can help to speed up our training process significantly.

Other work has used satellite images to perform oil spill detection [3], wheat yield prediction [4], habitat mapping [5], biomass estimation [6] and vegetation height estimation [7]. For regression specific problems [6][7], various regression schemes are used for the machine learning model with

,

Figure 1: The left most pictures shows examples of locations which planet images depict. The middle image shows the locations of AQI measurements, and the left most image shows an example image area matched to AQI data

(root) mean squared error ((R)SME) as an evaluation metric. This is similar to our project as we are also trying to obtain a numerical output with different input images.

Other research at the interface of Deep Learning and satellite imagery falls under the category of classification [4][5] or feature detection [3]. In general, while we've come across a substantial literature body on satellite imagery, our team was not able to find research that attempts to predict AQI values using satellite imagery.

3 Dataset and Features

A lot of time throughout the initial phase of the project had to be dedicated to pre-processing both satellite imagery and AQI data. The satellite images were acquired from planet.com [8]. AQI data was acquired from EPA [9]. We set up a three step pre-processing pipeline to retrieve and merge image and AQI data effectively.

Before querying any API for either image or AQI data, we narrowed the scope of our search down by defining a restricted geo-location and a restricted time period. Given the limitation and accessibility of planets image data set, our team have decided to choose California as the main geo-location, while limiting our time period to years after 2016.

Based on said geo- and date filters, we then queried both the AQI- and the image API. However, we only retrieved metadata from the image API in order not to waste bandwidth and local storage capacity, the geo-matching procedure is shown in Figure 1.

The image metadata contained all data points we need to match the available AQI data (see 1 for reference) with the underlying geo-location and date of the images. Since we will be using a model pre-trained on 200x200 images, however, we first need to subset the geo-location and pixel range in the image metadata of our 5000x5000 images, before we can math both data sources.

Date	PM2.5 Concentration	UNITS	DAILY_AQI_ VALUE	Site Name	STATE	COUNTY	SITE_LATITUDE	SITE_LONGITUDE
01/01/2019	5.7	ug/m^3 LC	24	Livermore	California	Alameda	37.687526	-121.784217
01/02/2019	11.9	ug/m^3 LC	50	Livermore	California	Alameda	37.687526	-121.784217

Table 1: Example AQI data acquired from EPA

Mathching subset images and AQI data base on date and location gave us valid sub-image/AQI matches. Based on this exhaustive set of image-metadata, we specified a representative test set of images before we actually went ahead to download the underlying images in bulk.

The image api can simply be queried with a pre-composed list of image IDs that we obtain from the previous step. We import these RGB images (.png) as 5000 x 5000 x 3 numpy matrices and split them up into 25 distinct 200 x 200 x 3 matrices. For every particular geo-location, we then define one anchor image and pair it up with other images from the same location but different dates. Finally, these pairs get stacked together with their AQI labels to compose the dataset which we then divide into a 20% test and 80% training dataset.

4 Method

Our team has decided to take the network developed by Neal Jean et al. [2] as the basic framework of our neural network. The network contains 8 convolutional layers and a pooling layer at the end. Since the goal of our project is to predict AQI values using baseline data that contains a satellite images and AQI values corresponding on both date and location, we figured that the most effective way to train the underlying model is to use a Siamese Network Architecture that discriminates the difference between 2 input images. A Siamese network is a set up as 2 parallel convolutional neural networks that noth use the same weights when 2 different images are passed through the network. This makes sure that the network extracts features the same way for both input images. Our team utilized transfer learning as we used pre-trained weights from the work from Neal Jean et al. [10], in hope that the feature extraction characteristics of this legacy network can help our neural to augment the difference between two input images with fewer training. An illustration of the neural network can be seen in Figure 2.

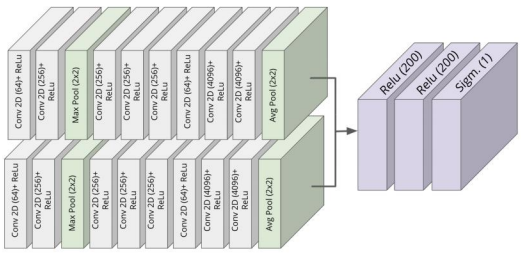

Figure 2: An illustration of the neural network setup

In our neural network setup, the loss function is defined as such: After two images have gone through the Siamese Network, we concatenate their flattened vector representations and feed this one-dimensional vector into an additional, one-layer neural network (i.e. dense layer with linear activation and single output unit) to map the vector towards a real-valued outcome label, which will be the difference between the AQI values of 2 images. We picked mean squared error (MSE) as a loss function to train our model. AQI values are defined on a range between 0 and 500, whereas higher values indicate higher degrees of pollution. Hence, the predictions we're feeding into the loss function are defined on a range between -500 and 500. The only meaningful metric that lends itself to evaluate a model against such a real-valued outcome variable is mean squared error (MSE). We have also determined that a batch size of 60 images (30 image pairs and their opposite) and a learning rate of 0.01 with ADAM optimizer works the best in minimizing loss compare to other hyper parameter settings.

By training the network with this setup, our team hopes that we can accurately predict the difference in AQI values between given images. A reference image can then be used to infer absolute AQI values for any image in the same location without relying on a measurement station.

5 Experiments and Results

5.1 Training Acceleration

Our team decided to implement our code base using tensorflow [11] and keras [12]. In the beginning, our team began to train the entire network without any parallelization (neither in the data pipeline, nor in the training process itself). With a total of more than 300,000 pairs of labeled training images, this approach was hardly feasible in the finite amount of time we had. Especially the batch-wise pre-processing of every 60 image batch took way too much time (about 5 minutes per batch). By leveraging keras' sequential interface [13], we were able to parallelize both pre-processing and training the network, which helped us iterate faster while optimizing other parts of the model architecture and the training process.

5.2 Improvement of Training Predictions

The initial predictions we got after having training the full model using entire data set for around 20 epochs were hardly meaningful in the sense that they were not even close to the actual AQI deltas (MSE > 500). Hence, we discussed our observations with members of the teaching team and came up with four measures to help guide the network closer towards a meaningful range of predicted values.

Given the empirical distribution of AQI deltas (Figure 3), the network should have the tendency of predict small AQI deltas for any given image pair. To leverage this intuition in favor of making faster learning progress, first, we selected highly skewed training data to more forcefully manipulate the legacy knowledge of the pre-trained network. Highly skewed in the sense that we only provide the neural network top 10% subset of our training data that has the highest AQI delta. Second, instead of using a linear activation function in the output layer, we used a scaled sigmoid function (i.e. $f(x) = \sigma(x) * 300 - 150$). A scaled sigmoid function is much more sensible to predictions that are close to zero, which corresponds very naturally to the empirical distribution of our data. In here, we only scaled the sigmoid function by 150 since there are rarely high AQI deltas in our training data set as shown in Figure 3. Third, we freezed the first six pre-trained layers of our network so that we have fewer high-level parameters to train but instead have the network focus on learning lower-level features that are more meaningful to our particular prediction task. Fourth, we added two additional, sigmoid-activated dense layers to the end of our network for it to learn more complex relationships between the images' vector representations and the AQI deltas. This last part of the architecture was inspired by a blog post written by Kevin Mader [14], who used this type of prediction head to gauge image similarity on the MNIST data set. All the above features are added after training results show that the loss are not decreasing, or decreasing relatively slowly.

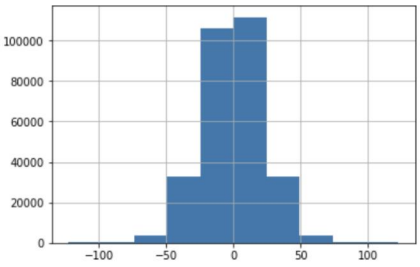

Figure 3: Empirical distribution of AQI delta of the training set

5.3 Improvement of Validation Predictions

With the adjustment made to the neural network mentioned in subsection 5.2, our team was able to minimize loss on the training set. However, when we observed the predictions from the half-freezed network which we trained on the skewed training data using the scaled sigmoid activation and a different prediction head, we immediately observed that we were running into substantial overfitting issues. While our training MSE had come down to 100, our validation MSE was stuck at 800. To counter overfitting, we decided on using two different forms for regularization. First, we added 0.2 dropout layers to the non-pretrained part of our network. Second, we added back random samples to our training data so that we had 50% images particularly with high AQI deltas and 50% images that came from the default training data distribution. With these regularization method implemented, our team observed that the validation set loss has successfully decreased over epochs (Figure 4).

5.4 Model Evaluation

To evaluate our trained neural network, we created a fresh sample of 18,900 labeled image pairs that are fetched from the planet [8] database separately. We ran predictions on these images and benchmark the resulting MSE against both, a meaningful baseline predictor (human performance) and the distribution of the outcome variable.

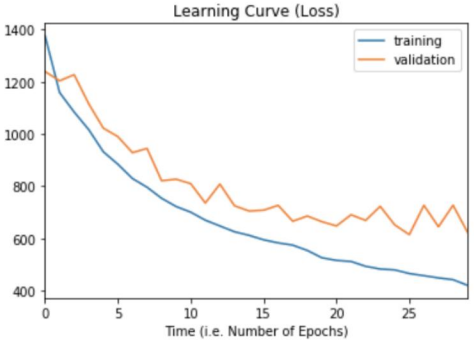

Figure 4: Training- and Validation loss for the Final Model Architecture

The prediction task we are solving is extremely hard even for humans to do. From the image pair in Figure 5, it is easy to see that the human eye can hardly tell which image has better air quality, while it is even harder to determine the difference in AQI values, whereas the AQI values between these images actually differ by 41 (i.e. air quality on in the left image is much worse than on the right).

Therefore, we assume for our benchmark predictor to behave closely to a human who knows about the empirical distribution of AQI deltas and when looking at these image paris, guesses the AQI difference randomly, based on his knowledge about the

underlying distribution. Hence, to come up with a corresponding baseline predictor for our test set (i.e. 18,900 images), we sampled 18,900 AQI deltas from the empirical distribution of our 300,000 training samples (Figure 3). This relatively profane approach yielded a baseline mean squared test error of 1231.53.

We trained the final model architecture (i.e. 3-layer prediction head, dropout, scaled sigmoid) on our entire training set of around 300,000 image pairs for a total of 30 epochs. Over those epochs, we observed a constantly decreasing training loss (MSE) from > 1376 after the first epoch to 421 after the last (i.e. 30^{th}) epoch (Figure 4). At the same time, we saw the gap between training and validation loss to remain far lower compared to when we had not applied any regularization. Quite disappointingly, however, the trained model was not able to get both training and validation loss down to meaningful levels as the validation loss plateaus and oscillates after 20 epochs. Talking from these validation results, we were not surprised to see a test error as high as 1041.38, which indicates that we still have a huge bias problem in our model and training data. The three image pairs below exemplify why it would require much more time and training (data) to use satellite images and make meaningful predictions for AQI values. There are so many different types of landscapes, shades, and outlyer conditions (e.g. cloudy images) that make for a very fat-tailed distribution of the true image and AQI data.

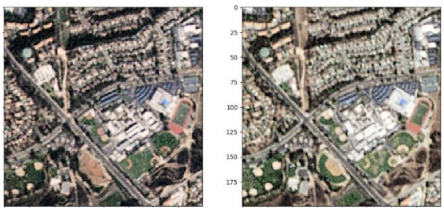

Figure 5: An example of training image pairs. The right image's AQI is 41 higher than the left image's

6 Conclusion/Future Work

As mentioned in section 5, our model only performs slightly better than the baseline predictor. This can be attributed to a number of reasons. Firstly, satellite images varies widely and can be drastically different even with the same AQI values due to clouds, landscape changes and time of the day. A lot of additional traing data might get the network network to a broader range of conditions. On the other hand, we could also narrow the purpose of our model to very specific conditions (i.e. no cloud, mid-day) and adjust training AND test data accordingly. Then, however, the training results won't be as meaningful and limited in their applications by default.

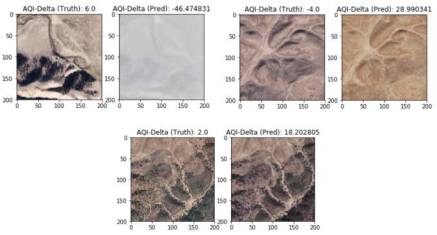

Figure 6: Image pairs illustrating problematic conditions for the NN to grasp.

There are also various hyper parameters that can possibly improve our model. For example, adding more layers with a different activation functions after the Siamese Network might help learn more complex features.

7 Contributions

Each team member made solid contributions to this project. Simon contributed on data processing and -acquisition, including downsizing images, matching AQI data, and building up the sequence training pipeline. Simon also assisted with research on network architecture and performed analysis on our results.

Michael contributed through research on topics, idea generation, including literature research, finding weights and consulting with members from Stanford sustainability and artificial intelligence lab. He also assisted with developing the code base.

Sean contributed on pre-processing AQI values from EPA. He also contributed on creating the training pipeline, fine-tuned the network architecture, and was leading cloud deployment and parameter tuning when training the model.

References

[1] United States Environmental Protect Agency, "Air Quality Index (AQI) Basics." https://www.airnow.gov/index.cfm?action=aqibasics.aqi. [Online; accessed 30-April-2019].

[2] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790–794, 2016.

[3] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine learning*, vol. 30, no. 2-3, pp. 195–215, 1998.

[4] X. E. Pantazi, D. Moshou, T. Alexandridis, R. L. Whetton, and A. M. Mouazen, "Wheat yield prediction using machine learning and advanced sensing techniques," *Computers and Electronics in Agriculture*, vol. 121, pp. 57–65, 2016.

[5] A. Kobler, S. Džeroski, and I. Keramitsoglou, "Habitat mapping using machine learning-extended kernel-based reclassification of an ikonos satellite image," *Ecological modelling*, vol. 191, no. 1, pp. 83–95, 2006.

[6] N. R. Jachowski, M. S. Quak, D. A. Friess, D. Duangnamon, E. L. Webb, and A. D. Ziegler, "Mangrove biomass estimation in southwest thailand using machine learning," *Applied Geography*, vol. 45, pp. 311–321, 2013.

[7] D. Stojanova, P. Panov, V. Gjorgjioski, A. Kobler, and S. Džeroski, "Estimating vegetation height and canopy cover from remotely sensed data with machine learning," *Ecological Informatics*, vol. 5, no. 4, pp. 256–266, 2010.

[8] Planet Team (2017), "Planet Application Program Interface: In Space for Life on Earth." https://api.planet.com. [Online; accessed 8-Jun-2019].

[9] United States Environmental Protect Agency, *Air Quality Index Daily Values Report*. [Online; accessed 8-Jun-2019].

[10] A. Gupta, "Satellite imagery ml." https://github.com/Arshita27/Satellite_Imagery_ML, 2018.

[11] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[12] F. Chollet, "keras." https://github.com/fchollet/keras, 2015.

[13] F. Chollet, "The sequential model api." https://github.com/keras-team/keras/blob/master/keras/engine/sequential.py, 2015.

[14] K. Mader, "Image similarity with siamese networks." https://www.kaggle.com/kmader/image-similarity-with-siamese-networks, 2018.