How Hazy it is Outside: Using Images to Predict Air Pollution with CNN

Adele Kuzmiakova
Stanford University
adele.kuzmiakova@stanford.edu

ABSTRACT

1. Introduction

While the air quality trends have been consistently improving in the US and Europe, the situation in rapidly developing countries is the opposite. Due to rapid urbanization, industrialization, and level of automobile usage, 98% of cities in low- and middle-income countries fail to meet World Health Organization (WHO) air quality guidelines [17]. Regions in East Asia are often hit the worst with annual air quality levels exceeding the WHO limits by a factor of 5-10. Air pollution is a significant cause of death and has been associated with an increased likelihood of stroke, heart disease, lung cancer, and asthma. Yet, the spatial extent and mapping of air pollution is often poorly understood due to a lack of density of existing on-the-ground monitoring stations. This lack of information leads to sub-optimal use public funds, time, and human resources since the decision-makers do not know which areas might benefit the most from immediate actions.

One way to approach the problem and provide the necessary information is to ask whether public webcam images taken at regular time intervals can be used to estimate the degree of outdoor haze, which is correlated with air quality. Therefore, the question is: do the time series of public images provide enough information for the CNN models to learn and predict the outdoor air quality?

To explore the feasibility of this premise, in Section 4.1 we test multiple baselines to infer which weather and image features are the most important predictors for the air quality index. In Section 4.2 we design 3 different CNN architectures: DehazeNet, VGG, and ResNet, and evaluate them in terms of their training and validation loss and quality of their predictions. Currently, the R^2 coefficient values for the VGG, ResNet, and DehazeNet vary from 0.1 to 0.32 for the validation set. To achieve better results, we close off with a summary of next steps in Section 5.

2. Background

2.1. Deep learning approaches

Deep learning approaches consist of multiple processing layers, which enable them to learn representations of raw input data (images) with multiple levels of abstraction. As such, deep learning models can be considered as representation-learning models, which use non-linear mapping functions to transform a representation at one level to a representation at a more abstract level. Despite their widespread applications in image recognition and classification [6, 14], deep learning models haven't been applied to haze detection very extensively – which is a problem that this work is exploring. This is presumably because it might be more difficult to learn global representations, such as haze, which affects all pixels globally, than it is to learn local representations, such as the presence or arrangement of edges, corners, and shapes at particular locations in the image.

In the past, deep learning approaches have been used to denoise images [5], predict the depth map [7, 9], and recently to estimate the transmission matrix to dehaze a single image [1, 8, 19]. Specifically, by learning the mapping between haze images and their corresponding transmission matrices, the researchers created a multi-scale CNN architecture for single-image dehazing. In [1], the network architecture takes a small 16×16 pixel patch and estimates a mean transmission value for that patch. By learning on multiple patches, the model is able to infer the final transmission map, which is then used to recover a haze-free image. Additionally, Ren et al. [11] developed a coarse-to-fine deep learning model for transmission estimation. The architecture consists of two sub-networks: one for coarse transmission prediction and one for fine transmission prediction. The coarse-scale net infers a holistic transmission matrix (using the whole image), which is then used as an input to the fine-scale net to refine the transmission map locally. The architecture utilizes large convolutional filters, for instance 11×11, 9×9, and 7×7.

Similarly, other researchers used CNN learning for dehazing the existing foggy images [3, 13, 12, 4, 8, 19].

Specifically, Song et al. [13] proposed a ranking layer, which changes the ordering of elements in each feature map so that statistical and structural attributes of the input images can be captured simultaneously. Additionally, Tang et al [16] investigated different haze-relevant features in a learning framework to identify the best combination for a single-image dehazing. A CNN-estimated transmission map can also be used to calculate the distance between an observer and an object to facilitate vision-based obstacle perception [2]. Other remaining approaches for haze visibility enhancement are summarized in [8, 15].

While the above works focused on inferring the transmission map to recover a haze-free image, we actually want to use a hazy image as an input to recover a scalar value, which corresponds to the outdoor air quality index. The closest work to ours appears to be that of Zhang, C. et al. [18], who used a CNN architecture to estimate air pollution. The architecture includes a modified activation function to dampen the effects of vanishing gradient during training. Additionally, a negative log-log ordinal classifier is adopted since it tends to perform better with labels that can be ordered – in this case, by an increasing particulate matter (pm) concentration. In terms of other deep learning approaches, Ong et al. [10] predicted the pm concentration in Japan using environmental monitoring data using a deep recurrent neural network pre-trained with auto-encoders. Finally, Li et al. [9] proposed a method to use a depth map of an image and its corresponding transmission matrix to predict the haze level.

3. Images and air pollution labels

3.1. Images

In the early stages of the project, we worked with public webcam images collected from 16 different locations across the US from 2008 to 2017. Figure 1 a shows their location on a map. We started with fitting individual models to each site individually although the ultimate goal was to pool all available sites into one model in order to make the learning algorithm more generalizable and scalable. While that still remains the goal, for the purpose of this paper we only pool data from 4 sites together. These include: 1) Anchorage, Alaska, 2) St Louis, Missouri, 3) Newburgh Heights, Ohio, and 4) Hamilton, Montana. Sample images from each site are included in Figure 1 b–e.

The images typically scan public highways, city skylines, or a common gathering point. Originally, we used a random 60%, 20%, and 20% split for the training, validation, and test split. However, since some of the images were taken within an hour (or even less) apart, they ended up in both train and validation sets, which led to an overfitting of the training set. Another option, which we ultimately used, is to split the data into training and validation

Figure 1. a) The location of 16 webcam sites that were chosen originally, b) sample image from Alaska, AK, c) St Louis, Mississippi, d) Newburgh Heights, OH, e) Hamilton, MT.

sets based on the year they were collected. Specifically, we partitioned images into the training set if there were collected between 2008 and 2014 and into the validation set if the collection date was between 2015 and 2017. We also checked a sample of images across all years to verify that camera's viewpoint did not change from year to year and compared the yearly air quality label distributions for consistency across the years. The time-split leads to an unequal size of the training and validation contributions with respect to each webcam location but avoids the problem of information leakage that arose from the randomized split. Table 1 summarizes the number of images from each webcam location in the training and validation sets.

Webcam ID	Location	# of images in the training set	# of images in the val. set
1066	Anchorage	22,945	11,348
17603	St Louis	31,861	6,904
21587	Newburgh H.	35,490	13,797
18879	Hamilton	20,969	10,586

Table 1. The number of samples from each webcam location in the training and validation sets.

We apply several pre-processing techniques. First, we filtered out all night-time images, corrupted empty images, single-color images due to erroneous background, and images associated with negative air quality label values since they might have introduced biases into our learning algo-

rithms. Second, a majority of public images carried a meta-data timestamp in the top or bottom corner. These sub-blocks were removed by cropping all images with respect to the center such that their resulting size was the same. Third, we subtract the mean image from each webcam location by calculating mean arrays for each (RGB) channel.

3.2. Air pollution labels

Their corresponding air quality indices supplied by the Environmental Protection Agency, also referred to as particulate matter (pm) labels, vary from 0 to around 70. (Figure 1 c) shows that the distribution of pm labels is asymmetric (right skewed). Therefore, a majority of the mass distribution lies on the left and is bounded by relatively small pm values [0, 10], which from the practical standpoint are not the most practical to optimize over and are sources of relatively low R^2 coefficients our baseline and deep learning models. We briefly experimented with various transformations for the pm label distribution, including centering by the mean and dividing by standard deviation. However, we also agreed that the label distribution is nearly lognormally distributed and both log base 2 and log base 10 were feasible transformations. For a quick experiment, we chose to work with log base 2 given the range of the original labels (max. value is below 100) and also the fact that the data might come from multiplicative processes governed by a factor of 2. The existence of sharp edges on the left is due to the fact that the majority of pm labels are reported as integers (e.g. 1, 2, 3, etc).

3.3. Framing the problem

There are two variants for examining and presenting the output labels: (1) regression and (2) classification. Although somewhat harder, regression is more informative because it allows us to compare predictions on the original scale as opposed to collapsing the spectrum of labels into several bins used in classification. Therefore, as our error metric, we use the mean absolute error for regression, MAE_r, which represents the 1-norm of absolute deviations from actual label values:

$$MAE_r = \sum_{i=1}^{n} \frac{|\, actual\, label - predicted\, label\,|}{n}$$

4. Methods

4.1. ElasticNet

We start with a simple baselines to get an estimate of how well they predict the outdoor haze from a series of haze-related features described in Tang et al [16]. Specifically, we use ElasticNet, which is a variable selection and regularized regression method. The main highlights of ElasticNet are a linear combination of L1 and L2 penalties arising from the lasso and ridge components. Additionally, ElasticNet

removes the limitation on the number of selected variables and encourages grouping effect, where it selects either an entire group of variables or assigns zero coefficients to all variables in that group. In terms of predictors for the ElasticNet, we include the following list of features:

- transmission features
- dark channel features
- saturation
- contrast
- power spectrum
- weather information, including temperature and relative humidity
- local meta-data, including local hour, day, and month

The hyper-parameters and λ were chosen using 10-CV on the validation set.

4.2. ResNet-50

Here we implemented ResNet-50 with pre-trained weights. The ResNet-50 model was trained on the last layer for 50 iterations and then on the entire architecture for another 50 iterations. We used the validation set to find optimal batch size from choices of 16, 32, 64, and 128. We used Adam optimizer and a random search to find the optimal learning parameter. As for comparison, we also implemented ResNet-101 with pre-trained weights. ResNet-101 converged faster to the optimal training and validation loss and R^2 but the final results were very similar to those obtained by ResNet-50. We also implemented batch normalization, which uses statistics from each mini-batch to normalize the activations.

5. Results

5.1. ElasticNet

We ran several sets of experiments using ElasticNet on the entire dataset consisting of 4 webcam locations listed in Table 2. For each experiment, hyper-parameters α and λ were chosen from scratch (separately) using the above-mentioned CV. We compare R^2 as a measure of the fit between actual and predicted labels.

In terms of feature selection, we extracted values for ElasticNet coefficients that were non-zero and normalized

Type of pm label distribution	Mean image subtraction	Training R^2	Validation R^2
Original	N	0.35	0.30
Log base 2 transform	N	0.36	0.30
Original	Y	0.29	0.21
Log base 2 transform	Y	0.31	0.21

Table 2. Summary of R^2 statistics from ElasticNet experiments.

the coefficients with respect to the maximum value to allow for relative comparison. Figure 2 summarizes the coefficients based on their groups: 1) transmission features, 2) dark channel features, 3) weather features, and 4) local meta-data features. Remaining features, such as power spectrum or contrast, were zero and thus are not included in the results below.

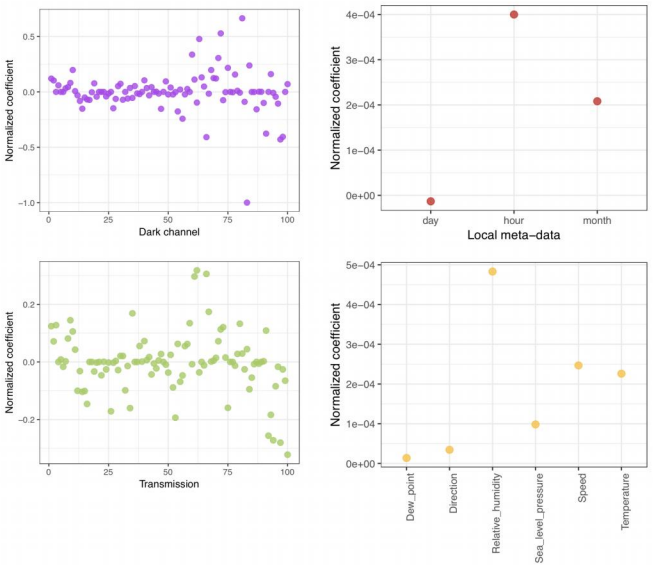

Figure 2. Normalized coefficients for each of the variable groups from ElasticNet

Surprisingly, we can see that neighboring transmission values and dark channel values are not correlated in terms of their weights. This is surprising because they are calculated from the neighboring patches and hence we would expect them to be at least somewhat correlated. At the same time, we can see that none of the weather features, such as temperature or relative humidity, play a significant role in the linear regression prediction. As a result, that may indicate that deep learning models might not need any extra features to be concatenated in the last layer and might be able to arrive at optimal predictions just by learning from images alone.

5.2. ResNet-50

Table 3 summarizes the results of R^2 from a variety of ResNet-50 experiments with pre-trained weights. Here we used Adam optimizer with values β_1 of 0.9 and β_1 of 0.999. We used different types of label distributions: either original or log-transformed, different pre-processing operation: either subtracted mean image for each site or not, and different learning rates by increments of 0.1. The best results were obtained for a configuration with learning parameter of 0.0001, without mean image subtraction, and using log-transformed labels for training.

Type of pm label distribution	Mean image subtraction	Learning rate	Train. R^2	Val. R^2
Original	N	0.001	0.33	0.31
Log base 2 transform	N	0.001	0.36	0.33
Original	Y	0.001	0.30	0.28
Log base 2 transform	Y	0.001	0.30	0.29
Original	N	0.0001	0.39	0.34
Log base 2 transform	N	0.0001	0.41	0.35
Original	Y	0.0001	0.36	0.32
Log base 2 transform	Y	0.0001	0.35	0.32
Original	N	0.00001	0.36	0.33
Log base 2 transform	N	0.00001	0.37	0.33
Original	Y	0.00001	0.33	0.30
Log base 2 transform	Y	0.00001	0.34	0.29

Table 3. Summary of R^2 statistics from ResNet-50 experiments.

The effect of mean subtraction is a little surprising because normally, we would expect that Gaussian scaling would help with the learning process. There were some options and uncertainties in terms of calculations of mean array for each RGB channel, and thus perhaps this is something worth re-visiting. It is also possible that R^2 for both training and validation sets may increase as a result of increasing sample size. This means that the bigger dataset we have and the more samples we include in the training, the more robust and generalizable algorithm will be. It is possible that right now the algorithm learns mostly on shapes of cars since these are dominant features after mean subtraction but ideally, the local statistics of the pixel should be invariant to their location or changing background (for instance, as caused by moving cars).

6. Conclusions and for further study

This project shows promise for using deep learning models to represent the degree of outdoor haze to predict the level of air quality for each image from the time-series records. Although the current best R^2 value for the validation set is around 0.4, the CNN model could be refined further to achieve even better accuracy. In that, the model could provide a means to monitor changes in the outdoor air quality to the extent they appear on pictures from public webcams or social media channels. At the same time, we identified several steps for further study, listed in the order of importance:

Logarithmic transformation of labels: Since most of the variations in R^2 coefficients comes from the low pm values (0 to 10), logarithmic transformation will shift the center of the mass towards higher end values with mean pm of around 8, thus leading to an increase in R^2 for both training and validation sets.

Median or median image subtraction: This is an interesting and potentially useful pre-processing technique because it could remove common patterns, which are not associated with haze itself. Background subtraction should work quite well for this static environment since it subtracts the silhouette of static objects, for instances building or house shapes but preserves dynamic elements, such as clouds, light, shadows, and other weather phenomena. Currently, we are using the mean RGB channel subtraction which does not lead to the best results. Since the RGB intensities are very similar to the mean, the resulting images are mostly black or contain dark intensities and lead to worse R^2 statistics.

Weather and time-of-day data: Concatenating the weather and time-of-day data with the image tensor in the last layer of the deep learning network could potentially improve R^2 statistics as well.

References

[1] B. Cai, X. Xu, K. Jia, C. Quing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *Proceedings of the 2016 ACM on Multimedia Conference - MM*, 2016.

[2] J. O. Gaya, L. T. Goncalves, A. C. Duarte, B. Zanchetta, P. Drews, and S. S. Botelho. Vision-based obstacle avoidance using deep learning. *Robotics Symposium and IV Brazilian Robotics Symposium*, 2016.

[3] F. Hussain and J. Jeong. Visibility enhancement of scene images degraded by foggy weather conditions with deep neural networks. *Journal of Sensors*, 2016.

[4] A. Ignatov, N. Kobyshev, K. Vanhoey, R. Timofte, and L. Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. 2017.

[5] V. Jain and S. S. Natural image denoising with convolutional networks. pages 769–776, 2009.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 2012.

[7] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016.

[8] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. Aod-net: All-in-one dehazing network. 2016.

[9] Y. Li, J. Huang, and J. Luo. Using user generated online photos to estimate and monitor air pollution in major cities. 2015.

[10] B. T. Ong, K. Sugiura, and K. Zettsu. Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting pm2.5. neural computing and applications. 2015.

[11] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Singlen image dehazing via multi-scale convolutional neural networks. *European Conference on Computer Vision*, 2016.

[12] Y.-S. Shin, Y. Cho, G. Pandey, and A. Kim. Estimation of ambient light and transmission map with common convolutional architecture. 2016.

[13] Y. Song, J. Li, X. Wang, and X. Chen. Single image dehazing using ranking convolutional neural network. *IEEE Transactions on Multimedia*, 2017.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[15] R. Tan, N. Pettersson, and L. Petersson. Visibility enhancement for roads with foggy or hazy scene. 2007.

[16] K. Tang, J. Yang, and J. Wang. Investigating haze-relevant features in a learning framework for image dehazing. 2014.

[17] WHO. Ambient (outdoor) air pollution database, by country and city. Technical report, World Health Organization.

[18] C. Zhang, J. Yan, C. Li, X. Rui, L. Liu, and R. Bie. On estimating air pollution from photos using convolutional neural network. *Proceedings of the 2016 ACM on Multimedia Conference - MM*, 2016.

[19] H. Zhang, V. Sindagi, and V. Patel. Joint transmission map estimation and dehazing using deep networks. 2017.