



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AUTO REGRESSION BASED RESNET METHODOLOGY FOR AIR POLLUTION DETECTION

R.Udaya Shanmuga
Department of Computer Science Engineering
Government college of Engineering, Tirunelveli-7

Dr.G.TamilPavai, (HOD)
Department of Computer Science Engineering
Government college of Engineering, Tirunelveli-7

Abstract:

The health effects of air pollution endanger human lives, especially among the at-risk population and those who suffer from respiratory illnesses. Hence the detection and monitoring of air pollution becomes necessarily important. Given the exorbitant cost of putting highly precise air pollution monitors throughout a city is quite expensive. In contrast to previous machine learning studies that primarily focus on pollutant estimation based on single day-time images, our proposed deep learning model integrates ResNet with Long Short-Term Memory, extracting spatial-temporal features of sequential images taken from smartphones instead for estimating air pollution using high equipped sensors and other sophisticated equipments. In this study, logistic regression is used to determine whether or not a data sample is polluted. Based on prior PM2.5 data, auto regression and other extracted features, ResNet is used to predict PM2.5 levels. Knowing the level of PM2.5 in the coming years, months, or weeks allows us to reduce it to a safe level. This proposed work aims to determine air quality level based on data set of camera images taken from a specific city in various time intervals.

Keywords— Air pollution, Autoregression, Deep Learning, ResNet (Residual Network)

I. INTRODUCTION

Particulate matter can be created by humans or exist naturally. Dust, ash, and sea-spray are just a few examples. Particulate matter (including soot) is produced when solid and liquid fuels are burned for power generating, home heating, and automobile engines. Particulate matter comes in a variety of sizes (i.e. the diameter or width of the particle). PM 2.5 refers to the mass of particles having a diameter of less than 2.5 micrometres (m) per cubic metre of air [13]. Fine particulate matter (PM2.5) is another name for PM2.5 (2.5 micrometres is one 400th of a millimeter). Fine particulate matter (PM2.5) is noteworthy in the pollutant index because it poses a considerable health risk to individuals when levels in the air are elevated

visibility and make the air appear hazy when concentrations are high. On the basis of a dataset containing daily atmospheric conditions, various machine learning models have been used to detect air pollution and predict PM2.5 levels. Dan Wei [2] used Bayes classification and support vector machine methods to achieve the lowest possible error in air quality prediction in Beijing. José Juan Carbajal et al.[4] proposed a fuzzy inference method for performing parameter classification utilizing a reasoning process and combining them into an air quality index. The problem is simplified to a binary classification due to the ambiguity of the particular number PM2.5 level, which divides PM2.5 levels into two categories: "high" ($> 115 \mu\text{g}/\text{m}^3$) and "low" (less than $115 \mu\text{g}/\text{m}^3$). The number is based on the Air Quality Level standard, which defines $115 \mu\text{g}/\text{m}^3$ as mild pollution [2]. Based on AQI, existing systems [2, 4, 14, 15] identify the air quality of a specific city specified by the user and categorize it as acceptable, satisfactory, moderate, poor, extremely bad, or severe (Air Quality Index). The information is presented monthly, weekly, or daily. Furthermore, once the numbers have been anticipated, they do not change in response to a sudden change in the weather or an unexpected rise in traffic. The values are detected for the entire city and cannot be verified for the accuracy of predetermined values. There are apps that show the PM2.5 levels in real time, and others that present the forecast for a specific day. PM2.5 readings for dates following a week, on the other hand, are not predicted. Based on a data collection of atmospheric conditions in a certain city, this system uses machine learning models to detect and predict PM2.5 levels. The proposed system accomplishes two goals, based on supplied atmospheric variables, detects PM2.5 levels and predicts PM2.5 levels for a specific date. To determine if a data sample is polluted or not polluted, logistic regression is used. Based on prior PM2.5 data, auto regression is used to predict future PM2.5 levels. The major purpose is to use ground data to anticipate air pollution levels in the city.

[1]. PM2.5 is a term that refers to very small particles. PM2.5 is used to describe microscopic particles in the air that limit

II. RELATED WORKS

When taking a photograph, PM_{2.5} can impact the light scattering coefficient [3] because it obscures the landscape and distorts the sky, reducing visibility [5]. Estimating pollution levels using smartphone photos is useful since it enables for quick capturing of any changes in air pollution levels. The traditional machine learning method correlates ambient light scattering coefficients with PM_{2.5} levels. To estimate scattering coefficients from single photos, the haze-image model [11] was commonly used. From pure sceneries with light-scattering effects, this model learns how to construct observed images (haze-images). When taking a photograph, PM_{2.5} can impact the light scattering coefficient [19] because it obscures the landscape and distorts the sky, reducing visibility [22]. From pure sceneries with light-scattering effects, this model learns how to construct observed images (haze-images). Some investigations used the dark channel model [12] in combination with the haze-image model to determine light coefficients directly from single photos [23], and Yang and Chen [14] used relative humidity to improve pollution estimation.

Liu et al. [7] and Zhang et al. [8] extracted picture attributes such as image entropy, contrast, and saturation for further improvement, as opposed to [33], [14]. Liu et al. [7] used Support Vector Regression (SVR) to predict PM_{2.5} concentrations using the haze-image model and extracted picture attributes. To assess air quality, Zhang et al. [8] made good use of multi-kernel learning. Similar features were employed by Liu et al. [9], who used a linear least square regression to estimate PM_{2.5} values from smartphone photos. Liu et al. [9] utilized a linear least square regression to predict PM_{2.5} values and used comparable features. Gu et al. [10] created a picture-based predictor instead of employing fundamental image attributes. Non-linear mapping was utilized to estimate PM_{2.5} values based on the overall likelihood of naturalness using the entropy information from the picture saturation map. Similar elements were adopted by Liu et al. [9].

III. METHODOLOGY

Regression analysis is a statistical technique for determining the relationship between two or more variables. Typically, researchers want to know if independent factors Y have a causal effect on dependent variables. Xi. When applying the model to forecast y for a certain set of xi values, we get., we'd like to have a look at how large the forecast's mistake might be All of these components, both dependent and independent, A regression analysis includes variables and errors, and the resulting forecast equation is typically referred to as a regression model In air pollution forecasting, regression analysis is a fundamental technique.

In the field of statistical methods, linear regression serves a solely utilitarian purpose. Its meaning is as follows:

$$Y = b_0 + b_1x + e \quad (2)$$

A multiple-linear regression (MLR) model is given as:

$$Y = b_0 + b_1x_1 + b_2x_2 + \dots + b_ix_i + e \quad (3) \text{ or:}$$

$$Y = b_0 + \sum_{i=1}^n b_ix_i + e_i \quad (4)$$

$Y = b_0$ concentration expected at time $t + 1$, in air pollution forecasting represents x_i pollutant concentrations and climatic

factors at time t , and b_i represents the regression coefficients. e is an estimated error factor derived from independent random sample coefficients. A least squares error technique can be used to calculate b_i . The dependent variable is Y , the independent variables are x and x_i , the regression coefficients are b and b_i , and the error is e . It has a normal distribution with a mean of 0 and a standard deviation of 0. Y denotes the pollutant. b_i Nonlinear regression analysis is an extension of the linear regression analysis, as well as the structural model of a traditional econometric analysis. In the social reality of economic life, many relationships between the analysis and forecast are generally used in nonlinear regression methods instead of a linear relationship. In the classical regression analysis, solving the nonlinear regression problem requires the conversion of variables to a linear relationship and the use of linear regression theory to determine the regression coefficients. This method has been widely used for many years in practice. General nonlinear regression models can be written in the following form:

$$Y = \phi(x_1, x_2 \dots x_m, \beta_1, \beta_2, \dots \beta_r) + \varepsilon \quad (5)$$

The cluster algorithm is used to find relationships between PM₁₀ and meteorological variables and then used multilayer regression to forecast the concentration of PM₁₀. The results show that meteorological variables are important in air pollution forecasting.

A. ARIMA Methods

The autoregressive integrated moving average (ARIMA) model is a linear model that can show steady state in both stationary and non-stationary time series. When constructing the ARIMA model, there are three main steps given below,

Step 1. Tentative identification

Step 2. Parameter estimation

Step 3. Diagnostic checking

ARIMA with a seasonal difference is called SARIMA. SARIMA processes the data with a seasonal period length S ; and if d and D are non-negative integers, the difference series, $W_t = (1 - B)^d(1 - B^S)^D x_t$, is a stationary autoregressive moving average process. The SARIMA model can be written as:

$$\phi_p(B)\phi_p(W^S)W_t = \theta_q(B)\theta_q(B^S)\varepsilon_t$$

$$t = 1, 2, \dots, N \quad (6)$$

where N is the number of observations up to time t ; B is the backshift operator defined by $B \alpha W_t = W_{t-\alpha}$; $\phi_p(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ is called a regular (non-seasonal) autoregressive operator of order p ;

$\phi_p(B^S) = 1 - \phi_p B^S - \dots - \phi_p B^{pS}$ is a seasonal autoregressive operator of order p ;

$\theta_q(B) = 1 - \theta_1 B - \dots - \theta_q B^q$ is a regular moving average operator of order q ;

$\Theta_q(B^S) = 1 - \Theta B^S - \dots - \Theta_q B^{QS}$ is a seasonal moving average operator of order Q ; In the definition above, p represents the autoregressive term; q is moving average order; P represents the seasonal period length of the model, S , of the autoregressive term; Q represents the seasonal period length

of the model, S , of moving average order; D represents the order of seasonal differencing; and d represents the order of ordinary differencing. When fitting a SARIMA model to data, the estimation of the values of d and D is primary, with the orders of differencing needed to make the series stationary and to remove most of the seasonality. The values of p , q and Q need to be estimated by the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the differenced series and other parameters can be estimated by suitable iterative procedures

B. Deep Learning

Deep Learning is an area of machine learning that is concerned with algorithms inspired by the structure and function of data. Artificial neural networks are a type of brain function [10]. Machine learning became more powerful as the amount of data increased. In terms of performance, techniques are insufficient, and Deep learning improves performance in areas such as accuracy. There are several modeling techniques appropriate for air pollution prediction in deep learning. LSTM (Long short term memory) method is the mostly used one for this purpose

IV. RESNET ARCHITECTURE

The primary advantage of a highly deep network is that it can represent extremely complicated functions. It can also learn features at a variety of

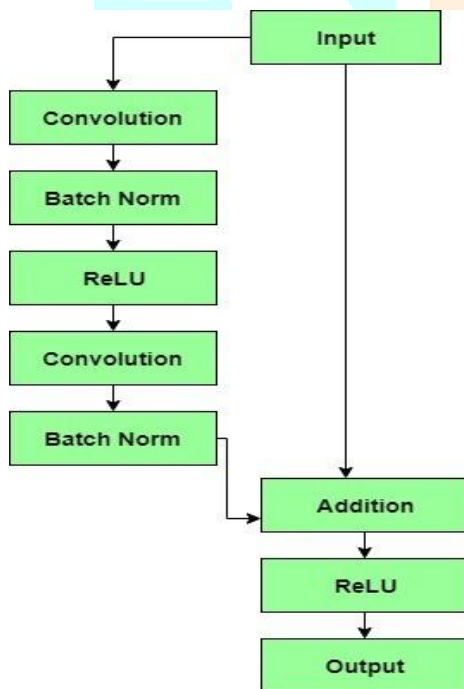


Fig. 1. ResNET Architecture

abstraction levels, ranging from edges (at shallower layers closer to the input) through more sophisticated features (at the deeper layers, closer to the output). However, due of the vanishing gradient problem, very deep neural networks are difficult to train. When the gradient is back-propagated to older layers, the repeated multiplication operations cause the gradient to become endlessly tiny. When a result, as the network depth increases, its performance saturates or even decreases rapidly. This topic inspired the residual network (ResNet), which enables the construction of robust deep convolutional neural networks. The main idea of the ResNet model is the residual blocks, whose architecture is shown in

Figure 4. The residual block provides two paths for the input: the main path and the shortcut (or more commonly known as skip-path). The main path provides the normal flow as with any convolutional neural network. Still, the shortcut skips N convolution layers (in our approach $N = 2$) and provides its input to the following convolution layer. The authors in argue that stacking layers should not degrade the network performance because one could simply stack identity mappings (layers that learn the identity mapping which ultimately does not make any change) upon the current network, and the resulting architecture would perform equally. This indicates that the deeper model should not produce a training error higher than its shallower counterparts. They hypothesize that letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlying mapping.

A. Residual Block:

The residual block explicitly allows the model to perform this task. The basic residual block is also known as the identity block. This block corresponds to the case where the input activation has the same dimension as the output activation. Apart from this block, there is another block called the convolutional block. The architecture of this block is shown in Figure 1. In this block, the input and output dimensions do not match. The difference between this block and the identity block is that there is a convolutional layer in the shortcut path to match the dimensions. The ResNet model's overall architecture (Figure 1) consists of numerous convolutional and identity blocks. It also includes convolution and max pooling layers in the model's beginning, as well as flatten and dense layers at the conclusion to generate the final prediction. Better results can be produced by utilizing this model, which can be explained by the fact that much deeper networks that can learn complicated features and are not influenced by the problem of vanishing gradient can be employed. The emergence of ResNet or residual networks, which are made up of Residual Blocks, has relieved the challenge of training very deep networks. The first difference we notice is that there is a direct link that bypasses several levels (which may vary depending on the model) in between. This connection is known as the 'skip connection,' and it lies at the heart of residual blocks. Because of this skip link, the layer's output is no longer the same. Without this skip link, the input ' x ' is multiplied by the weights of the layer that comes after it. Next, this term goes through the activation function, $f()$ and we get our output as $H(x)$.

$$H(x) = f(wx + b)$$

$$\text{Or } H(x) = f(x)$$

Now with the introduction of skip connection, the output is changed to

$$H(x) = f(x) + x$$

There appears to be a slight problem with this approach when the dimensions of the input vary from that of the output which can happen with convolutional and pooling layers. In this case, when dimensions of $f(x)$ are different from x , we can take two approaches:

1. The skip connection is padded with extra zero entries to increase its dimensions.
2. The projection method is used to match the dimension which is done by adding 1×1 convolutional layers to input. In such a case, the output is:

$$H(x) = f(x) + w_1 \cdot x$$

Here add an additional parameter w_1 whereas no additional parameter is added when using the first approach. The skip connections in ResNet alleviate the problem of disappearing gradient in deep neural networks by enabling the gradient to flow through an additional shortcut channel. Another way these connections aid is by allowing the model to learn the identity functions, ensuring that the higher layer performs at least as well as the lower layer, if not better. A shallow network and a deep network that use the function H to convert an input 'x' to an output 'y' (x). The deep network must perform at least as well as the shallow network, without degrading performance as we with simple neural networks (without residual blocks). One method is for additional layers in a deep network to learn the inputs, preventing them from degrading identity function, such that their output equals performance even with extra layers. It has been demonstrated that residual blocks make it extremely simple for layers to learn identity functions. The formulas above demonstrate this. The output of plain networks is a shallow network and a deep network. $H(x) = f(x)$, So an identity function, $f(x)$ must be equal to x which is grader to attain whereas in case of ResNet, which has output:

$$H(x) = f(x) + x$$

$$f(x) = 0$$

$$H(x) = x$$

Initialize $f(x) = 0$ which is easier and get x as output which is also input. In the best-case scenario, extra layers of the deep neural network can better approximate the mapping of 'x' to output 'y' than its shallower equivalent, reducing error by a large margin. As a result, we anticipate ResNet to outperform traditional deep neural networks. Using ResNet greatly improved the performance of neural networks with more layers, as seen by the plot of error percent when compared to neural networks with simple layers. Clearly, the difference is significant in networks with 34 layers, where ResNet-34 has a substantially lower error percentage than plain-34. Also, we can observe that the error percentage for plain-18 and ResNet-18 is nearly identical. Using ResNet has considerably improved the performance of neural networks with more layers.

B.. Vanishing Gradient Problem:

This isn't a big deal for a shallow network with only a few layers that use these activations. However, if more layers are utilized, the gradient may become too small for training to be successful. Back propagation is used to find neural network gradients. Simply put, back

propagation finds the network's derivatives by moving layer by layer from the final to the start layer. The derivatives of each layer are multiplied down the network (from the final to the start) by the chain rule to compute the derivatives of the original layers. When n hidden layers utilize an activation such as the sigmoid function, the n tiny derivatives are multiplied together. As a result, the gradient diminishes exponentially as we descend to the original layers. When n hidden layers utilize an activation such as the sigmoid function, the n tiny derivatives are multiplied together. As a result, the gradient diminishes exponentially as we descend to the original layers with a small gradient, the weights and biases of the initial layers will not be efficiently updated with each training session. Because these first layers are frequently critical to recognizing the main aspects of the incoming data, it can lead to overall network inaccuracies. The most straightforward answer is to utilize other activation functions, such as ReLU, which do not produce a small derivative. Another option is to use residual networks, which provide direct links to preceding layers. As shown in Image 2, the residual connection adds the value at the start of the block, x , to the end of the block ($F(x) + x$). This residual connection avoids activation functions, which "squash" the derivatives, resulting in a greater total derivative of the block. Finally, batch normalization layers may be able to fix the problem. As previously indicated, the issue emerges when a vast input space is transferred to a tiny one, leading the derivatives to vanish. This is particularly visible when $|x|$ is large. Batch normalization solves this problem by simply normalizing the input so that $|x|$ does not reach the sigmoid functions outside edges. It normalizes the input so that the majority of it falls in the green region, where the derivative isn't too small, as seen in Figure 2.

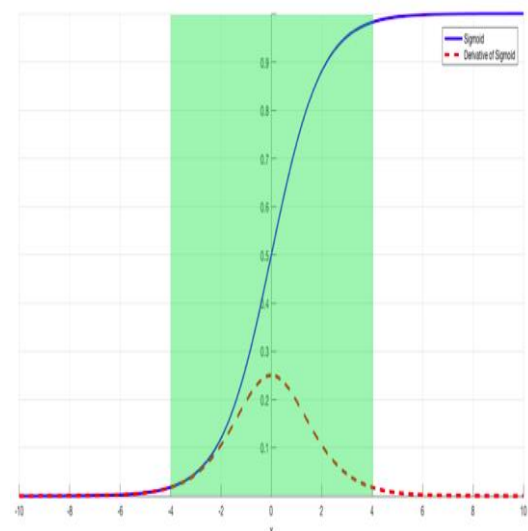


Fig. 2. Problem of vanishing gradient

V. IMPLEMENTATION

The complete dataset consists of 178,992 images, where 80% of the images are taken for training the models, 15% of the images are for testing, and 5% for validating. We decided to use a smaller percentage for validation and testing datasets because we have a bigger dataset, and using this approach, we can leverage the large amount of data for training purposes. The validation dataset was used for hyper-parameter tuning, where a grid search approach was used. The hyper-parameters tuned are: learning rate, batch size, and optimizer used. Our proposed models have incorporated pollution data and weather information for the last two years. The air pollution data is collected from the API endpoints of pulse.eco (<https://pulse.eco>). The API provides information about different sensors and different pollutants at different timesteps. The timesteps are irregular, so the data is aggregated hourly for each sensor. Every row consists of weather information, timestamp (date and time), the pollutant type, and the amount measured. Our models have used only the PM2.5 pollution to labeling the images during the training process. Therefore, they are not inputs in the machine learning models. This method also uses images taken from a stationary camera. The camera takes periodical pictures of the center of the city, and at the same time, air pollution sensors measure the exact air quality. Based on the air quality measurements in terms of PM2.5 concentration, the images were labeled with six classes depending on the Air Quality Index (AQI) of the European Union, as shown in Table 1. The implications of the six AQI indexes are the following. AQI-1 means that the air quality is satisfactory, and air pollution poses little or no risk. AQI-2 means that the air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution. AQI-3 means that members of sensitive groups may experience health effects. The general public is less likely to be affected. With AQI-4, some members of the general public may experience health effects, and members of sensitive groups may experience more serious health effects. AQI-5 requires a health alert because the risk of health effects is increased for everyone. AQI-6 entails issuing a health warning of emergency conditions because everyone is more likely to be affected. Ideally, a system would be able to estimate air pollution precisely. However, the fact that the proposed models are attempting to do this based on the camera images and not actual air pollution sensors makes it unreasonable to define the task as a regression problem. Therefore, our initial approach was to use six classes, one for each AQI category. We additionally redefined the problem as a binary classification problem, collapsing AQI-1 and AQI-2 into the “not polluted” class, and the other AQI indexes into the “polluted” class. In some models, we have also incorporated the weather information to distinguish between weather conditions and pollution. The weather information was collected from the API endpoints of World Weather Online (<https://www.worldweatheronline.com>). The data consists of temperature, wind speed, wind direction, weather description, precipitation, humidity, visibility, pressure, cloud coverage, heat index, and the UV index. This is used a simple strategy that considers the last measured value to handle the missing pollution measurements, although more sophisticated approaches based on generative models can be incorporated. Even though we have merged the six classes into two general

classes, the dataset is still highly imbalanced, and our models could become very biased to non-polluted images. For that purpose, we applied different techniques for balancing the dataset and evaluated their impact on the classification performance.

TABLE I. AIR QUALITY INDEX CATEGORIES

Table 1. Air Quality Index (AQI) categories based on ranges of PM2.5 values and mapping to labels in.

AQI Category	PM2.5 Range	6-Class Labels	Binary Labels
Good	0–50	AQI-1	Not polluted
Moderate	51–100	AQI-2	Not polluted
Unhealthy for Sensitive Groups	101–150	AQI-3	Polluted
Unhealthy	151–200	AQI-4	Polluted
Very Unhealthy	201–300	AQI-5	Polluted
Hazardous	301 and above	AQI-6	Polluted

TABLE II. DISTRIBUTION OF DATASET

Table 2 shows the distribution of the dataset in when using six classes. The distribution of the dataset after collapsing the six classes into two for illustrative purposes Table 2. Distribution of the dataset in 6 classes.

Dataset	AQI-1	AQI-2	AQI-3	AQI-4	AQI-5	AQI-6
Train	80,331	21,623	13,954	11,087	9862	6337
%	56.1%	15.1%	9.7%	7.7%	6.9%	4.4%
Test	13,342	4219	3022	987	1564	1135
%	52.8%	16.7%	12.0%	7.9%	6.2%	4.5%

This shows the different architectures’ training and testing accuracy, depending on the number of epochs. It is clearly visible that the models are stable, and after a small number of epochs, the performance does not vary significantly. However, another clear result is that the accuracy is about 56%, the same as the majority class ratio. This means that the models learned to classify all images as “AQI-1,” i.e., the “good” AQI category. As such, the predictions are not useful, which was one of the main reasons to evaluate the binary classification approach that collapsed multiple categories into only two. The training and testing accuracy of the different architectures, depending on the number of epochs and whether class balancing was performed or not is predicted accurately. These results confirm that the models are stable, and after about 40 epochs, the performance does not vary. Note that the test set remains unbalanced in all experiments because the balancing is performed only on the training set. The reason for that is that in a production setting, the camera images would not be going that class over the total number of images.

VI. CONCLUSION

Our study consists of four stages. First, analyzed and collected meteorological data based on input image. Second, calculated relevant climate factors and other datas by auto regression analysis. Third, based on our features and RESnet, a comprehensive dataset containing 3024 images have been constructed. Modal training is performed to derive empirical features and finally display the predicted class to determine the level of air pollution.

REFERENCES

- [1] Environmental Protection Department. Accessed: Oct. 29, 2020. [Online]. Available: <https://www.aqhi.gov.hk/en/what-is-aqhi/about-aqhi.html>
- [2] H. Akimoto, *Atmospheric Reaction Chemistry* (Springer Atmospheric Sci-ences). Tokyo, Japan: Springer, 2016.
- [3] M. Kampa and E. Castanas, "Human health effects of air pollution," *Environ. Pollut.*, vol. 151, no. 2, pp. 362–367, Jan. 2008.
- [4] X. Meng, Y. Zhang, K.-Q. Yang, Y.-K. Yang, and X.-L. Zhou, "Potential harmful effects of PM_{2.5} on occurrence and progression of acute coronary syndrome: Epidemiology, mechanisms, and prevention measures," *Int. J. Environ. Res. Public Health*, vol. 13, no. 8, p. 748, Jul. 2016.
- [5] J. C. Chow, J. G. Watson, J. L. Mauderly, D. L. Costa, R. E. Wyzga, S. Vedal, G. M. Hidy, S. L. Altshuler, D. Marrack, J. M. Heuss, G. T. Wolff, C. A. Pope, III, and D. W. Dockery, "Health effects of fine particulate air pollution: Lines that connect," *J. Air Waste Manage. Assoc.*, vol. 56, no. 10, pp. 1368–1380, Oct. 2006.
- [6] Y. Xing, Y. Xu, M. Shi, and Y. Lian, "The impact of PM_{2.5} on the human respiratory system," *J. Thoracic Disease*, vol. 8, no. 1, p. E69, 2016.
- [7] C. Liu, F. Tsow, Y. Zou, and N. Tao, "Particle pollution estimation based on image analysis," *PLoS ONE*, vol. 11, no. 2, Feb. 2016, Art. no. e0145955.
- [8] Z. Zhang, H. Ma, H. Fu, L. Liu, and C. Zhang, "Outdoor air quality level inference via surveillance cameras," *Mobile Inf. Syst.*, vol. 2016, pp. 1–10, Jan. 2016.
- [9] X. Liu, Z. Song, E. Ngai, J. Ma, and W. Wang, "PM_{2.5} monitoring using images from smartphone
- [10] *Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 630–635.
- [11] K. Gu, J. Qiao, and X. Li, "Highly efficient picture-based prediction of PM_{2.5} concentration," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3176–3184, Apr. 2019.
- [12] G. Yue, K. Gu, and J. Qiao, "Effective and efficient photo-based PM_{2.5} concentration estimation," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 3962–3971, Oct. 2019.
- [13] C. Zhang, J. Yan, C. Li, H. Wu, and R. Bie, "End-to-end learning for image-based air quality level estimation," *Mach. Vis. Appl.*, vol. 29, no. 4, pp. 601–615, May 2018.
- [14] A. Chakma, B. Vizena, T. Cao, J. Lin, and J. Zhang, "Image-based air quality analysis using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3949–3952.
- [15] C. Zhang, J. Yan, C. Li, X. Rui, L. Liu, and R. Bie, "On estimating air pollution from photos using convolutional neural network," in *Proc. ACM Multimedia Conf.*, 2016, pp. 297–301.
- [16] Liu, W. Liu, Y. Zheng, H. Ma, and C. Zhang, "Third-eye: A mobilephone-enabled crowdsensing system for air quality monitoring," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–26, 2999018.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [20] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf., BMVA Press*, 2015, p. 41, doi: [10.5244/C.29.41](https://doi.org/10.5244/C.29.41).
- [21] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.
- [22] T. Zheng, M. H. Bergin, S. Hu, J. Miller, and D. E. Carlson, "Estimating ground-level PM_{2.5} using micro-satellite images by a convolutional neural network and random forest approach," *Atmos. Environ.*, vol. 230, Jun. 2020, Art. no. 117451.