Pollution Level Estimation Through Image Analysis

Shashank Saurav Bhavishaya

M.Tech CS

Supervisor: Dr. Sarbani Palit

A thesis submitted in fulfilment of
the requirements for the degree of
M.Tech

CVPR

Indian Statistical Institute

Kolkata

Dedication

To my parents and close ones, without your help and encouragement it wouldn’t have been possible.

Acknowledgements

I would like to thank my dissertation supervisor Dr. Sarbani Palit for agreeing to guide me and for helping me to undertake work in the topic. Without her continuous guide and support this wouldn’t have been possible.I am also very much thankful to Harsh Sir,and my lab mates and seniors of the CVPR unit of Indian Statistical Institute , Kolkata for helping me through out the project with their valuable time and suggestions.

Abstract

Climate change is one of the hardest problems humanity will have to face in the next century. Data analysis and computer vision are two powerful tools that can help us perform tasks that would usually take more time and resources to finish. Therefore, monitoring air quality, especially in developing countries should be the first step to save the environment. Measurement of air quality is a task that, currently needs the help of specialized equipment and infrastructure. These equipments are either very costly or require skills to operate or both making it difficult to provide air quality information at remote locations or at desired spots even in cities. In this study, we have tried to measure the air quality through images which can be taken using a normal camera. For this purpose, we used deep learning techniques, where we trained ResNet18 using a public image database. Performance is evaluated by plotting confusion matrix. We also measure precision, recall, F1-score and accuracy. Results are analyzed by plotting ROC curve and precision-recall curve.

Contents

Chapter 1

Introduction

The term air pollution can be defined as “It refers to the release of pollutants into the air that are detrimental to human health and the planet as a whole.” In the last decade we have witnessed exponential growth in factories and industries, especially in developing countries like India, China, Afghanistan, Argentina etc. Due to industrialization, economic condition of these countries has been stabilized, but it comes at the cost of the air quality of these countries which has worsened continuously. Among these countries, China is at the top of the list while India comes at second position. The term air pollution can be defined as “It refers to the release of pollutants into the air that are detrimental to human health and the planet as a whole.” There are many causes of air pollution, some of which are listed below:

1. Most air pollution comes from energy use and production;
2. Burning fossil fuels releases gases and chemicals into the air;
3. Air pollution in the form of carbon dioxide and methane raises the earth’s temperature;
4. Another type of air pollution is then worsened by that increased heat: Smog forms when the weather is warmer and there’s more ultraviolet radiation.

The air quality badly affects the human health and environment changes, few of the effects are mentioned below.

- Global warming: Global warming is a direct consequence of the greenhouse effect, which is produced by the high emission of CO_2 and methane into the atmosphere.
- Climate changes: When the temperature of the planet increases, there is a disturbance in the usual climatic cycles**.**
- Smog effect: This is a kind of fog which is a load of pollutants and can be of 2 types: sulphurous smog and photochemical smog, both dangerous and harmful to health**.**
- Deterioration of fields: Acid rain, climate change and smog all damage the Earth's surface. Contaminated water and gases seep into the earth, changing the composition of soils. This directly affects agriculture, changing crop cycles and the composition of the food we all eat**.**
- Last but not least, there are other harmful effects such as extinction of animal species, skin damage and negative impact on human health caused by air pollution.
- The common approach to analyze the quality of air and pollution level present in the environment is to collect samples and analyze it by measuring presence of carbon, nitrogen, sulphur and other chemicals. With advances in image processing and computer vision techniques, it is now possible to measure air quality index using machine learning and image processing techniques. For this purpose, we need some images of polluted area, which may contain smog and dust. A sample image of how a city looks from above in some polluted areas is shown in Figure 1.

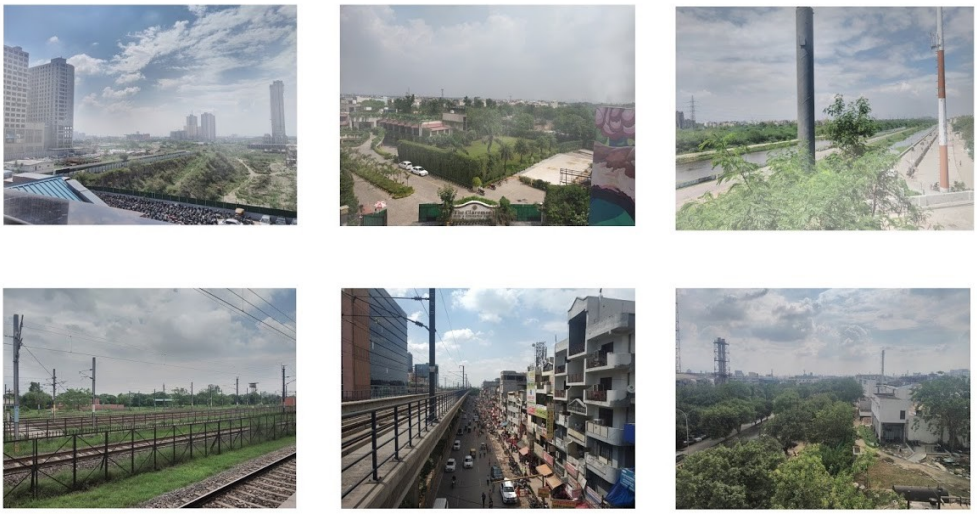


Figure 1: Images of polluted cities showing foggy environment.

Therefore, estimation of air quality index based on image/picture of a city gained attention of researchers. Some methods and algorithms can be found in existing literature. In [1], Chung et al., used satellite images to detect and monitor air quality. Filter based gravimetric method for air quality estimation is proposed by Hauck et al [2]. A similar study conducted by Chen et al., where they collected data from social media and captured through cameras to analysis smog disaster [3]. In the past decade, many studies highlighted the impact of ambient airborne particulate matter (PM), especially PM2.5, as a critical pollutant leading to different cardiopulmonary diseases and lung cancer. Providing real-time PM2.5 measurements can help people to schedule their daily routines to avoid exposure to airborne carcinogens. In [7], authors have tried to infer air pollution by sniffing social media.

A similar effort is made in [4], where researchers used mobile phones to capture image and provide air quality index (AQI) in real-time. The architecture of ML model followed in this study is presented in Figure 2.

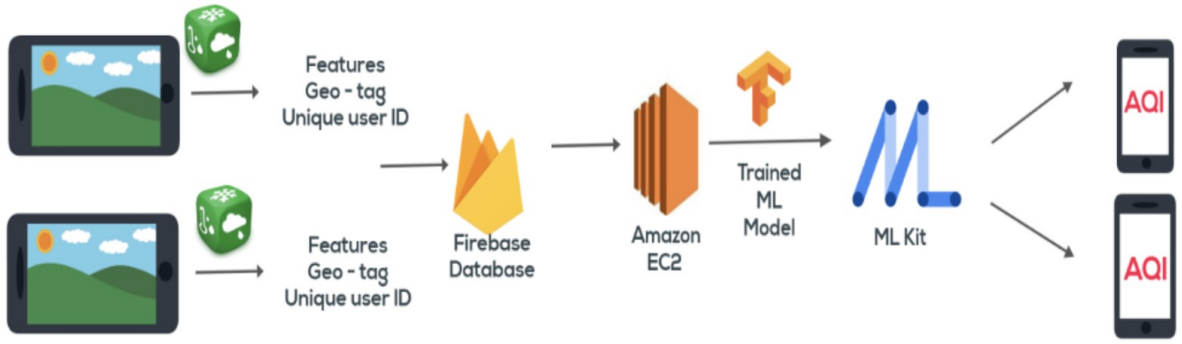


Figure 2: Architecture of machine learning model to estimate air quality index.

The model is trained with images samples having >50% skylines i.e. mostly outdoor images and < 50% skyline i.e., indoor images. Gradient of Sky, entropy and RMS contrast features are extracted from images. The limitation of this method is that, the features are hand crafted. The performance of the model can be improved if automatic means of features extraction algorithm is used to extract feature pattern from images. Due to excellent performance of deep learning and its characteristics to extract feature automatically, in this study we adopt deep model to estimate air quality index.

CHAPTER 2

RELATED WORK

Many work has been done in past in the area of predicting air pollution level or air quality(prediction of AQI value) and suspended particle estimation like pm2.5,SO_2,NO_2 ,etc through image analysis. Athanasiadis[9] proposed a *σ-fuzzy-lattice* neurocomputing classifier model to categorise O_3 level with help of meteorological data.Kalapanidas[10] proposed a classifier model using lazy learning approach to classify air pollution into four different level (low, med, high, and alarm) with meteorological data.Other researchers have worked on predicting concentrations of pollutants.[11]Corani proposed a neural network to predict O_3 and PM_{10} concentraion by data from previous day,Feed forward neural network and pruned neural network are the two neural network models used to for comparison.Further developments on Feed Forward Neural Networks have been done: Gray model and Rolling mechanism is advised by Fu [12] to improve traditional Feed Forward Neural Networks models. Jiang [13] applied several models (chemical and physical model, regression model, and multiple layer perceptron) on the air pollutant prediction task, and their results show that statistical models are competitive with the classical physical and chemical models.Ni, X. Y. [14] compared multiple statistical models on the basis of PM 2.5 data around Beijing, and their results implied that linear regression models can in some cases be better than the other models.However, the process of converting regression tasks to classification tasks is problematic, as it ignores the magnitude of the numeric data and consequently is inaccurate.

Chapter 3
Architecture of ResNet

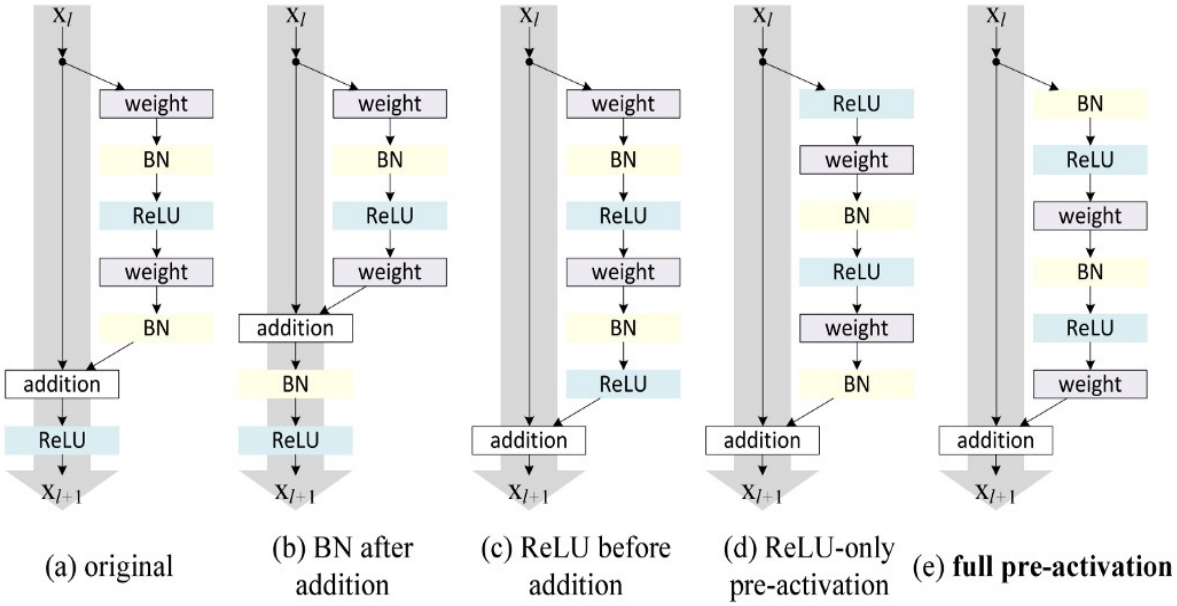
Among all deep learning architectures, VGG16 [5] and its variants got more attention of researchers. Because of its depth, it can extract most dominants features and produces excellent result. Due to extensive number of layers in its network architecture, VGG get popularity but it also increased the training time and cost. To overcome this issue, residual network architecture, also known as ResNet 18 has been proposed. It has 18 layers but having skip connection. Before going to discuss about skip connection, here, first we will see the design of each network. Both architectures used 3x3 filters and down sampling with CNN layers with stride 2. Both designs have global average pooling layer and a 1000-way fully-connected layer with Softmax in the end. Figure 3 presents the architecture of plain network, i.e., VGG and VGG with residual blocks.

Plain Network: The plain baselines (Figure 3, middle) are mainly inspired by the philosophy of VGG nets [41] (Figure 3, left). The convolutional layers mostly have 3x3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer.

Residual Network: Based on the above plain network, a shortcut connection (Figure 3, right) which turn the network into its counterpart residual version. The identity shortcuts can be directly used when the input and output is of the same dimensions (solid line shortcuts in Fig. 3). When the dimensions increase (dotted line shortcuts in Figure 3).

Again, there are many variants of ResNet like ResNet18, ResNet 34, ResNet50, ResNet 101 and ResNet152. The details of the architectures are shown in Figure 4. However, in our study we used ResNet18. Like variants

of ResNet architectures, the residual blocks used in ResNet has many variants and shown in Figure below.



In our experiment, we used the original version of residual block as shown in (a). The basic difference between these variants are performing addition operations. In the original version of residual connection, addition operation is done after batch normalization (BN) and before applying ReLu while in other versions, addition is done after BN (b), before ReLu (c), ReLu only pre-activation (d) and full pre-activation (e).

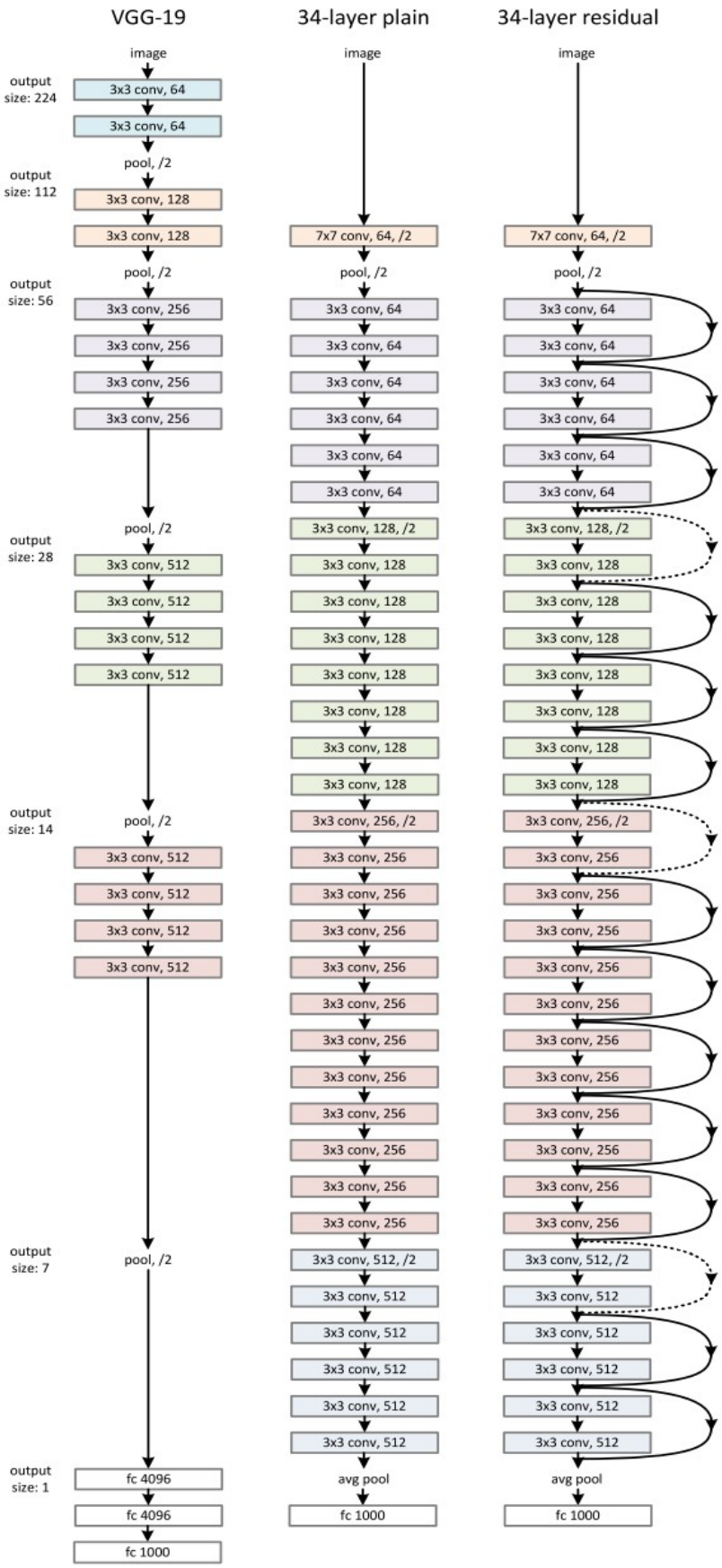


Figure 3: Plain VGG and VGG with Residual Blocks [6]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 4: Variants of ResNets [6]

The ResNet18 accept input image of fixed size, i.e. 224x224 pixel resolution and after execution of each block, reduces it dimensions by half. Finally, it downgrades image till image resolution 7x7 which are passed to 4096 neurons after flattening operation. After each block operation, input image size downgrades from 224x224, 112x112, 56x56, 14x14 to 7x7 and number of filters used increases from 64, 128, 256 up-to 512. In other words, as depth of network increases, input image size decreases and number of filters increases. The original ResNet18 architecture is trained on ImageNet dataset which consist of 1000 objects. Therefore, it has 1000 neurons in fully connected layers. Finally, softmax classify input image to one of the 1000 categories.

Chapter 4

Proposed Approach: Transfer Learning and Fine Tuning of ResNet 18

The core idea of ResNet is introducing a so-called “identity shortcut connection” that skips one or more layers. It is claimed that stacking layers shouldn’t degrade the network performance, because we could simply stack identity mappings (layer that doesn’t do anything) upon the current network, and the resulting architecture would perform the same. 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 underlaying mapping. And the residual block above explicitly allows it to do precisely that.

4.1 Theory

Let us focus on a local neural network, as depicted below. Denote the input by $\mathbf{x}$. We assume that the ideal mapping we want to obtain by learning is $f(\mathbf{x})$, to be used as the input to the activation function. The portion within the dotted-line box in the left image must directly fit the mapping $f(\mathbf{x})$. This can be tricky if we do not need that particular layer and we would much rather retain the input $\mathbf{x}$. The portion within the dotted-line box in the right image now only needs to parametrize the *deviation* from the identity, since we return $\mathbf{x}+f(\mathbf{x})$. In practice, the residual mapping is often easier to optimize. We only need to set $f(\mathbf{x})=0$. The right image in Figure 5 illustrates the basic Residual Block of ResNet. ResNet follows VGG’s full 3×3 convolutional layer design. The residual block has two 3×3 convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a

ReLU activation function. Then, we skip these two convolution operations and add the input directly before the final ReLU activation function. This kind of design requires that the output of the two convolutional layers be of the same shape as the input, so that they can be added together.

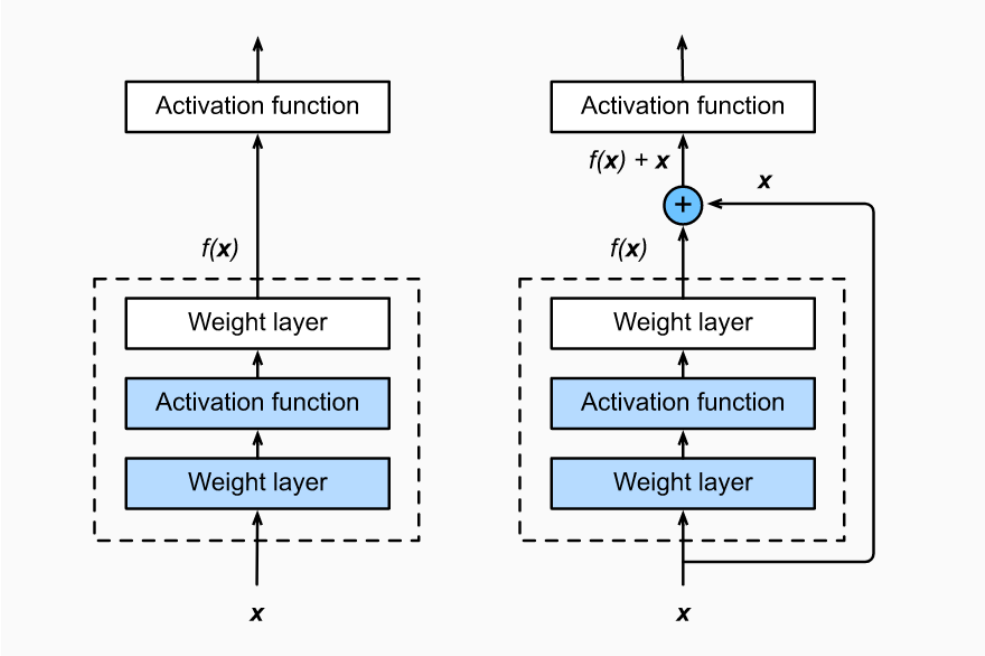


Figure 5: Understanding or residual connection [6]

4.2 Training technique

There are two ways to train a deep model: Transfer learning with fine tuning and training from scratch.

1. *Transfer learning with fine tuning*: In this approach of training we transfer knowledge of model which has already trained with another dataset. In other words, in this technique, we use pre-trained model and fine tune network architecture by modifying last layer (softmax) to the number of class we want to classify. Except last layer, we usually freeze all layer and don't train it.

   This technique is mostly used when we have a smaller number of images in the dataset. This approach reduces the training time and cost as well as also proved to improve accuracy of the system.
2. *Training from scratch*: In this approach of training, model is trained from scratch. Input image is passes through each layer and back

propagate feedback/error/loss. After each epoch, model learn some features and repeat it again till loss is converges to minimum.

This approach of training is suitable for large data set. The greater number of images, the more accuracy of the model we can expect. The drawback with this approach is that, it requires comparatively large amount of time and cost to train the model from scratch.

In our study, keeping in mind the number of images in the database, we preferred 1st approach. In other words, we used Transfer learning with fine tuning approach to train ResNet18 deep model architecture.

4.3 Network Architecture Fine Tuning

The basic architecture of ResNet 18 is discussed in Chapter 2. However, in our experiment, we are training model using transfer learning by freezing all layers except last layer. And modifying the last layer to the number of class we have. Therefore, the architecture of network used to train with database under study is shown in Figure 6.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	6	

Figure 6: ResNet18 architecture with fine tuning

If we analyze the architecture of basic ResNet18 and fine-tuned ResNet18, we can notice that, in the basic model there are 1000 objects to classify, while in our case, we need to classify images into six levels. Rest all the internal layers are frozen.

4.4 Database used

To predict the air quality level, we are using Chinese database. The database images are collected at different index of air quality and divided into six level. Level 1 having 29 images, Level 2 and Level 3 has 14 and 29 respectively. Similarly, there are 22, 34 and 13 images that are belongs of Level 4, Level 5 and Level 6 categories. All images are collected in outdoor and saved in .jpg format. The samples of database image with corresponding levels are shown below:

Figure 7: Sample database images with corresponding image quality/levels.

4.5 Model training

Our contribution: Throughout this study, we used ResNet18 deep learning architecture to train the model with Chinese image database in order to predict air quality through image processing technique. We freeze all layers except the last one. The number of classes/labels are changed to the required number of classes. For this purpose, we fine-tuned the network architecture to the number of classes we want to classify. One of the advantages of transfer learning and fine tuning is that it reduces the training time and resource cost. The use of transfer learning is proved to be more robust to over-fitting/under-fitting.The tuned

ResNet18 architecture is shown in Figure 6. We trained model multiple times with objective to improve in the performance. We set a low learning rate i.e. , 0.001 with momentum 0.9 and trained with 50 epoch which was experimentally found suitable to yield better result with over-fitting/under-fitting. We also employed data augmentation by performing multiple types of transformations. The details of the experimental study are discussed below.

The database images are divided into two sets: Training and testing. Training set consist of 80% images while model is tested with 20% of images. Due to a smaller number of images in the database, we performed data augmentation.

Data Augmentation: Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples. Image data augmentation is the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image. Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more.

In our experiment, we are generating data by performing data augmentation using following transformation.

1. *Rotation*: We rotate images randomly by 30 degrees in clockwise and counter clockwise direction.

2. *Zoom*: We perform zooming operation by the scale of 20%.

3. *Width shift range*: We shift width by a factor of 20%,.

4. *Height shift range*: Similar to width shift, we shift height by 20%

5. *Shear:* We perform shear transformation by a factor of 15%

6. *Shear:* We are performing shear transformation by a factor of 15%

7. *Flip:* Flipping of images in horizontal and vertical direction are performed

8. *Fill mode*: Whenever due to transformation, filling of pixels is required, we used nearest neighbour interpolation.

4.6 Hyper parameters:

Learning rate: The amount that the weights are updated during training is referred to as the step size or the "*learning rate*." Specifically, the learning rate is a configurable hyper parameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0.

In our experiment, we trained model with learning rate = 1e-3.

Decay: When training neural networks, it is common to use "weight **decay**," where after each update, the weights are multiplied by a factor slightly less than 1. This prevents the weights from growing too large, and can be seen as gradient descent on a quadratic regularization term. In our experiment we set decay=1e-5 and found it suitable for convergence of accuracy/loss.

Momentum: The momentum by which learning rate will fluctuate is set to 0.9 .

Optimization algorithm: Gradient Descent is a popular optimization technique in Machine Learning and Deep Learning, and it can be used with most, if not all, of the learning algorithms. A gradient is the slope of a function. It measures the degree of change of a variable in response to the changes of another variable.

Types of Gradient Descent:
Typically, there are three types of Gradient Descent:

1. Batch Gradient Descent
2. Stochastic Gradient Descent
3. Mini-batch Gradient Descent

Stochastic Gradient Descent (SGD): The standard gradient descent algorithm updates the parameters θ of the objective $J(\theta)$ as,

$$\theta = \theta - \alpha \nabla_\theta E[J(\theta)]$$

where the expectation in the above equation is approximated by evaluating the cost and gradient over the full training set. Stochastic Gradient Descent (SGD) simply does away with the expectation in the update and computes the gradient of the parameters using only a single or a few training examples. The new update is given by,

$$\theta = \theta - \alpha \nabla_\theta J(\theta; x(i), y(i))$$

with a pair ($x(i)$, $y(i)$) from the training set.

Following figure shows how SGD works with initial weight decreases and convergence at global minimum.

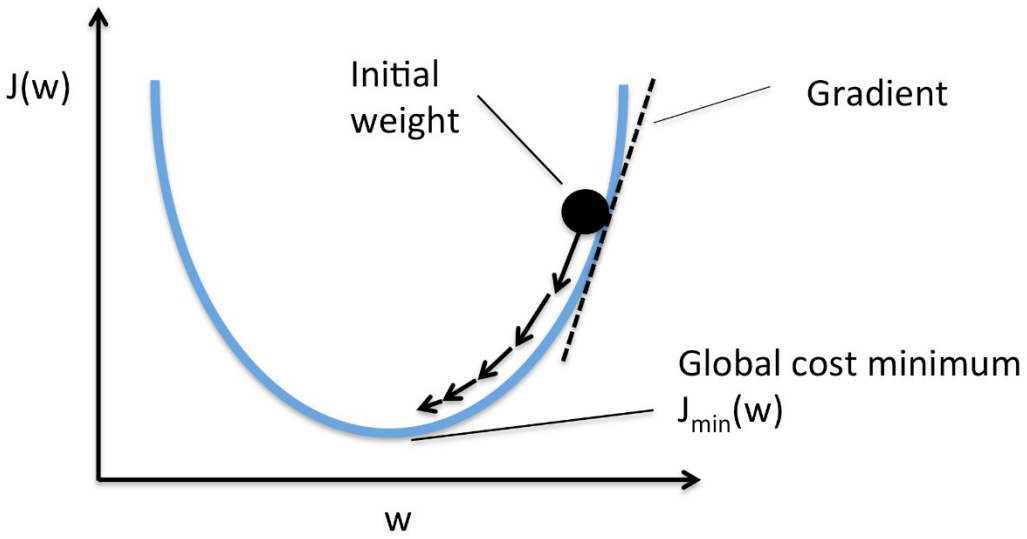


Figure 8: Minimization of weight using SGD [8].

Epoch: We trained model with 50 epoch which were found to be sufficient for convergence of validation accuracy and loss.

Software and Tools:

Operating System: Windows 10

Memory: 8GB, 512 SSD

Language: Python 3.6.6

Framework: Keras and Tensorflow

Resources: GPU NVIDIA, CUDA 10.2

4.7 Experimental Results and Analysis:

We plot learning curve by varying number of epochs with respect to accuracy/loss. The curve is shown in Figure 9. From this figure, we observe that as number of epochs increases training accuracy and validation accuracy also increases while training and validation loss decreases. After 50 epochs we found that there is fall in the validation accuracy so we stop the training process there to avoid overfitting/underfitting. However, fluctuations in the curve can be noticed which may have risen due to a low number of images in the database.

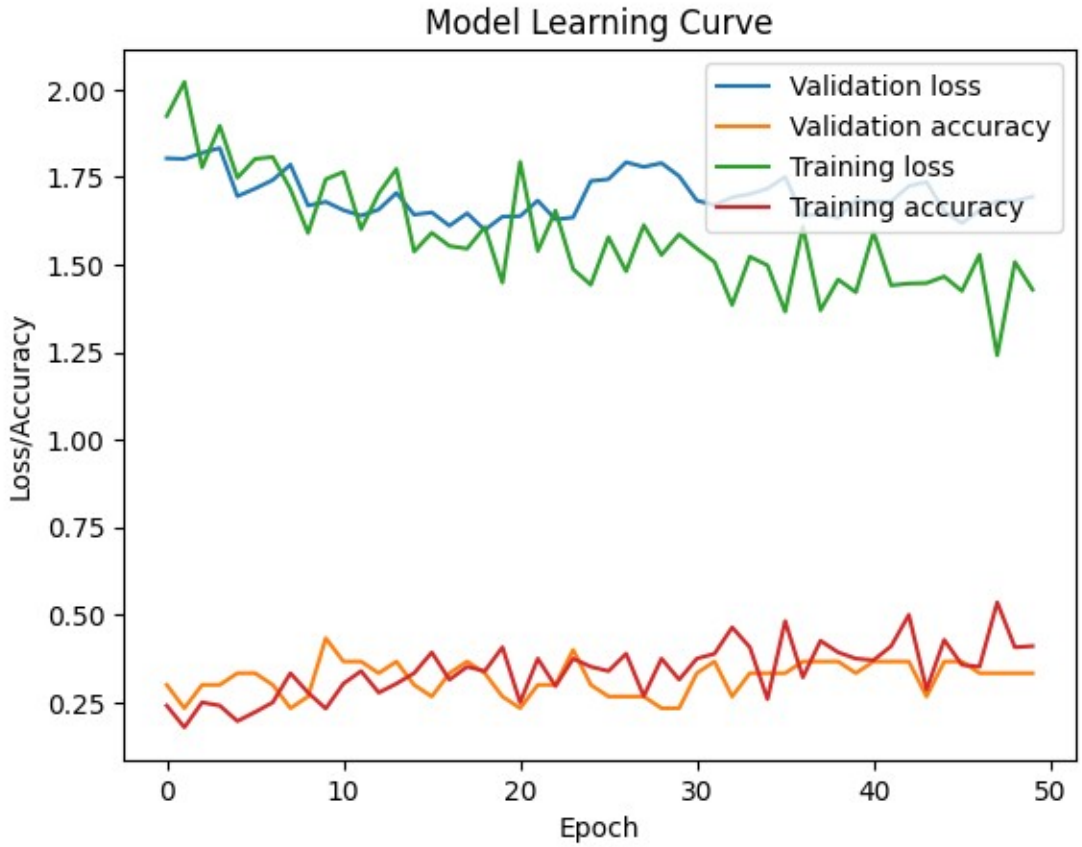


Figure 9: Learning curve showing plot of accuracy/loss with respect to epochs.

With 20% of test set we evaluate the performance of model by calculating confusion matrix. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which a classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly, the types of errors that are being made. We normalized confusion matrix to show how many % of image have been correctly classified and how many are misclassified. Normalized confusion matrix is

shown in Figure 9

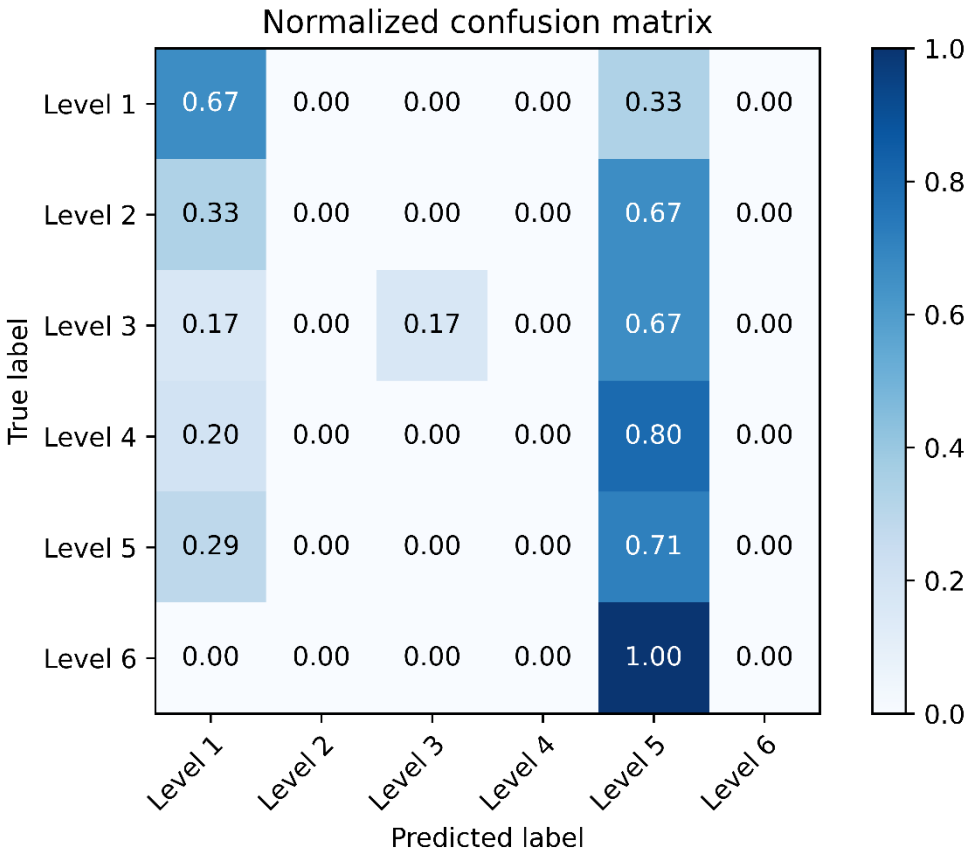


Figure 9: Confusion matrix showing classification and misclassification rate.

Discussion: From the figure 9 we can notice that, 76% of images that belongs to Level 1 have been classified correctly while 33% images which actually belong to Level 1 have been classified as Level 5. Similarly, 17% images which belong to Level 3 have been classified as Level 3 and rest of the images have been predicted as Level 5. 71% images are correctly

classified for images that belong to Level 5. However, Level 2, 4 and 6, we didn’t get any classification results (i.e., 0% images are correctly classified) and all of them have been misclassified. The main reason for low classification accuracy is the amount of images in the database. The misclassification rate can be decreased if we train with large dataset.

From the confusion matrix, we calculated precision, recall, F1 score and accuracy of model. We also calculated precision, recall and F1 score of each class as well as average (macro and weighted) value.

Level	**Precision**	**Recall**	**F1 score**
Level 1	0.44	0.67	0.53
Level 2	0.00	0.00	0.00
Level 3	1.0	0.17	0.29
Level 4	0.00	0.00	0.00
Level 5	0.25	0.71	0.37
Macro avg	0.28	0.26	0.20
Weighted avg	**0.35**	**0.33**	**0.25**

Table 1: Performance results of experimental study

Precision tells how model can precisely predict the level of a given image. In other words, it tries to answer the question “When the model predicts positive, how often is it correct?” Precision talks about how precise/accurate your model is out of those predicted positive, how many of them are actual positive. Mathematically, it can be calculated as:

Precision = True positive / (true positive + false positive)

The higher the value means better the classification. In our experiment, we got weighted average precision 0.35.

Recall measure the ration of true positive and sum of the true positive and false negative. Recall actually calculates how many of the Actual Positives our model capture through labelling it as Positive (True Positive). It can be written as:

recall = True positive / (true positive + false negative)

From the table 1, we can notice the calculated recall value is 0.33.

F1 is an overall measure of a model's accuracy that combines precision and recall, in that weird way that addition and multiplication just mix two ingredients to make a separate dish altogether. That is, a good F1 score means that one has low false positives and low false negatives, so real threats are being correctly identified and withought being disturbed by false alarms. An F1 score is considered perfect when it's 1, while the model is a total failure when it's 0. The mathematical representation of F1 score is:

F1 = 2 x (precision x recall) / (precision + recall)

Finally, accuracy of the model is calculated by dividing sum to total positive and true negative by all number of examples. The equation of accuracy can be written as:

Accuracy = (true positive + true negative) / total example

From the experiment we found trained model is 33% accurate.

We also plot **receiver operating characteristic curve**, or **ROC curve.** It is a graphical plot that illustrates the diagnostic ability of a classifier system as its discrimination threshold is varied. The obtained ROC curve is shown in Figure 10.

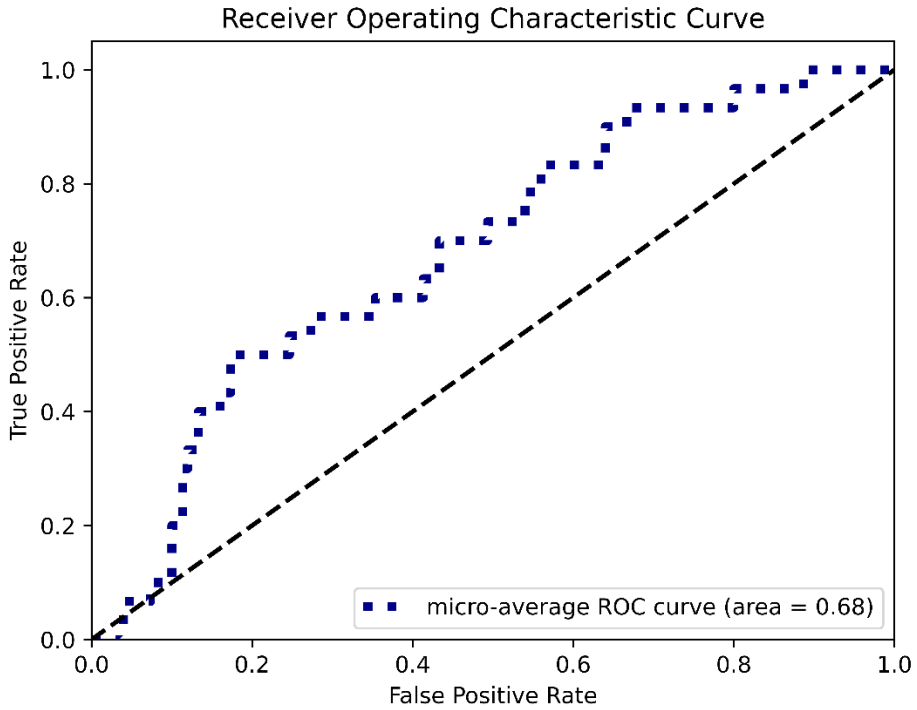


Figure 10: ROC curve

Discussion: From the curve we can see that the area under curve (AUC) is = 0.68. The classification is said to be perfect if we have more area under the curve. A good classifier has objective to yield AUC value ~ 1. The low value of AUC might be because of smaller amount of data. The area under curve can be improved by adding more images for each class in the database. The ROC curve shows the change in true positive rate with respect to false positive rate at different value of thresholds. The selection of threshold depends on the type of application. Usually, the threshold at which false positive rate becomes equal to true positive rate, is considered for the development of application.

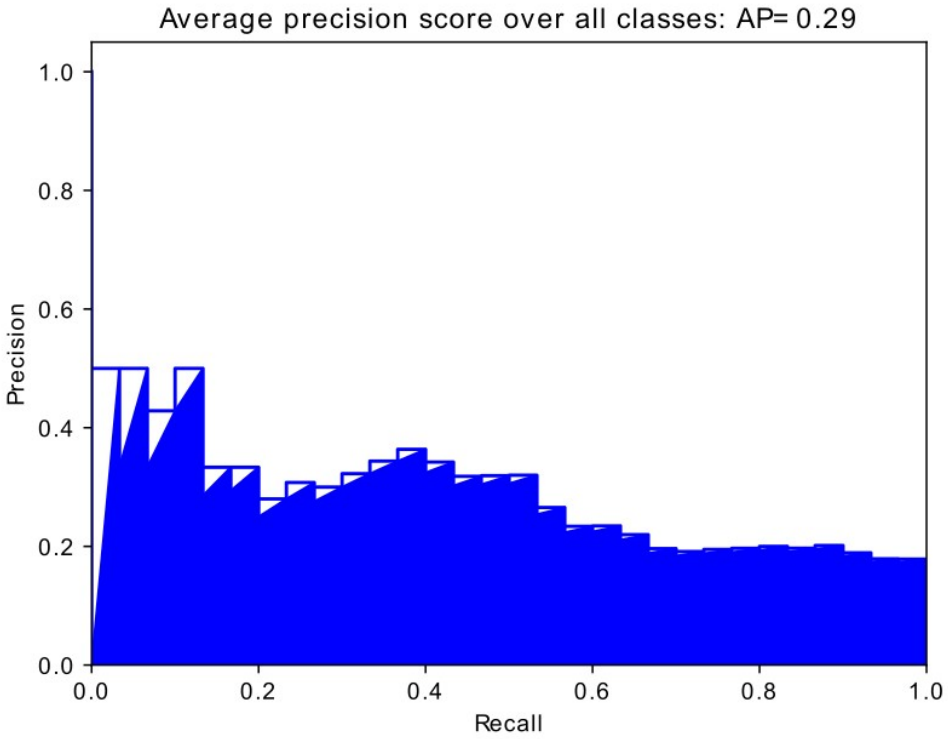


Figure 11: Precision-Recall curve

Discussion: In Figure 11, we plot precision-recall curve. It shows validity of database. If images in the database is unbalanced, i.e., number of images in each class is not equal. In that case, precision-recall curve may produce low score. Another reason that may yield low average precision score is number of images in the database. Since in our database, number of images are very small, we have been able to achieve average precision equal to 0.29 only. The precision score can be improved by adding more images in the database. Precision-recall curve also reflects this drawback of the database.

4.8 Conclusion

Throughout this experiment, we tried to predict air quality index divided into six labels using images captured by a camera. For this purpose, we trained deep model such as ResNet18 with the Chinese image dataset. Transfer learning with fine tuning is used to train model. Since, we have

six levels, we modified architecture by freezing all layers except last one. Last layer is modified to the number of classes we have, i.e. six. Model is trained and tested with database image dividing it in the ration 80:20. From test set, we evaluated the performance of trained model by calculating confusion matrix. To analyze the overfitting and underfitting, we plot learning curve. Performance of model is also shown and analyzed by receiver operating characteristic (ROC) curve and precision recall curve (PR curve). We calculated average area under curve while plotting ROC and average precision score using PR curve. From the test set, we also calculated precision, recall and F1 score of each class as well as average and weighted average values. These results are presented in Table 1. Finally, we calculated accuracy. From the results we found that model is 33% accurate with precision 35%. This is the best result we can have with limited database. From learning curve, we also shown that model is not underfit/overfit. The performance of the model can be improved by adding more images in the database.

CHAPTER 5

FUTURE WORK

In this experiment all the images are taken from a single camera. Images from multiple cameras may be utilized with color corrections applied to the images in order to have all the images with similar color characteristics. Two possible color corrections are midway histogram equalization for multiple images and automatic white balance. Many countries do not have uniform weather conditions throughout the year, and so developing an approach which would work in all weather (including fog) would be the next challenge. Developing methods to estimate pollution levels from pictures taken during night will also be explored.

Finally, developing a mobile application incorporating the above ideas would provide easy reach and accessibility to common people.

References

[1] Y. Chung. Air pollution detection by satellites: The transport and deposition of air pollutants over oceans. Atmospheric Environment (1967), 20(4), pp.617-630, 1986.

[2] H. Hauck, A. Berner, B. Gomiscek, S. Stopper, H. Puxbaum, M. Kundi, and O. Preining. On the equivalence of gravimetric pm data with teom and beta-attenuation measurements. Journal of Aerosol Science, 35(9), pp.1135- 1149, 2004.

[3] J. Chen, H. Chen, G. Zheng, J. Z. Pan, H. Wu, and N. Zhang. Big smog meets web science: smog disaster analysis based on social media and device data on the web. In Proceedings of the companion publication of the 23rd international conference on World wide web companion, pages 505{510. International World Wide Web Conferences Steering Committee, 2014.

[4] https://blog.tensorflow.org/2019/02/air-cognizer-predicting-air-quality.html

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In dvances in neural information processing systems, pp. 1097-1105, 2012.

[6] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. In Computer vision and Pattern Recognition, 2015.

[7] S. Mei, H. Li, J. Fan, X. Zhu, and C. R. Dyer. Inferring air pollution by sniffing social media. In Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, pp. 534-539. 2014.

[8] http://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/

[9] Athanasiadis, I.N.; Kaburlasos, V.G.; Mitkas, P.A.; Petridis, V. Applying machine learning techniques on air quality data for real-time decision support. In Proceedings of the First international NAISO Symposium on Information Technologies in Environmental Engineering (ITEE'2003), Gdansk, Poland, 24–27 June 2003.
Information Technologies in Environmental Engineering (ITEE'2003), Gdansk, Poland, 24–27 June 2003.

[10] Kalapanidas, E.; Avouris, N. Short-term air quality prediction using a case-based classifier. Environ. Model. Softw. 2001, 16,pp. 263–272.

[11] Corani, G. Air quality prediction in Milan: Feed-forward neural networks, pruned neural networks and lazy_learning. Ecol. Model, 185,pp. 513–529.2005

[12] Fu, M.; Wang, W.; Le, Z.; Khorram, M.S. Prediction of particular matter concentrations by developed feed-forward neural network with rolling mechanism and gray model. Neural Comput. Appl., 26, pp.1789–1797, 2015.

[13] Jiang, D.; Zhang, Y.; Hu, X.; Zeng, Y.; Tan, J.; Shao, D. Progress in developing an ANN model for air pollution_index forecast. Atmos. Environ. 2004, 38, 7055–7064.

[14] Ni, X.Y.; Huang, H.; Du, W.P. Relevance analysis and short-term prediction of PM 2.5 concentrations in Beijing based on multi-source data. Atmos. Environ., 150, pp.146–161, 2017.