MARVAir: Meteorology Augmented Residual-based Visual Approach for Crowdsourcing Air Quality Inference

Yao, M, Tao, D, Wang, J, Gao, R & Sun, K

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:
https://dx.doi.org/10.1109/tim.2022.3193197

DOI 10.1109/tim.2022.3193197
ISSN 0018-9456
ESSN 1557-9662

Publisher: IEEE

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.
MARVAir: Meteorology Augmented Residual-based Visual Approach for Crowdsourcing Air Quality Inference

Muyan Yao, Dan Tao*, Jiangtao Wang, Ruipeng Gao, and Kunning Sun

Abstract—Air pollution has become a prominent problem in citizens’ everyday life. Since the weather stations are not densely distributed, it is difficult to get fine-grained PM$_{2.5}$ (Particulate Matter $<2.5$ µm) data. We propose MARVAir to address limitations of traditional PM$_{2.5}$ prediction mechanism. First, we use crowdsourcing and spiderbots to fetch visual and meteorological dataset respectively. Second, a ResNet based visual core is designed to learn the image data, and an 1D-CNN based meteorology core is deployed to tune the inference. Besides, we use decision-level fusion mechanism to unite the sub-models and provide precise yet everywhere available fine-grained PM$_{2.5}$ inference. In addition, cloud-side model training is also proposed to restrict local energy consumption. Evaluation on dataset collected at 8 sites over nearly 2 years suggests that, MARVAir achieves a precision of 98.8%, a recall of 99.0%, and an F1 score of 98.9% under various air conditions, which notably exceed baseline solutions.

Index Terms—Air quality, inference, crowdsourcing, PM$_{2.5}$, ubiquitous, visual, model fusion, smartphones.

I. INTRODUCTION

Nowadays, air pollution has long become one of the biggest public health hazards worldwide [1], causing about 9 million deaths per year [2]. However, observations from sparsely distributed traditional weather sites cannot provide fine-grained PM$_{2.5}$ (particulate matter with an aerodynamic diameter smaller than 2.5 µm) information, urging a new solution for air quality inference.

To provide reference for people’s daily activities, researchers have explored solutions for air quality inference, especially relying on data not sourced from traditional weather stations. In general, existing works can be divided into two categories: using dedicated hardware or using ubiquitous perceptual data. The former one refers to the methods that rely on air status data, which are sourced from a relative portable but still dedicated device. [3] and [4] deploy sensing boards or sensor arrays respectively on large vehicles for data collection. [5] uses stationary but smaller sensor sets, and [6] even integrates necessary components on a wristband. These solutions fail to evolve the conventional sensing framework, involving devices that people do not carry normally, causing inconvenience thus limiting their actual application. The latter one includes the approaches using data collected in a crowdsourcing manner. [7] analyzes tens of million air and weather records to construct a correlation between PM$_{2.5}$, air gradients and urbanization data. [8] also works on the interpolation methods for environment sensors. [9] uses reports from several weather stations along with traffic information to construct a fine-grained inference on PM$_{2.5}$. Under similar frameworks, [10], [11] tune the model to improve PM$_{2.5}$ inference. Besides, there are works for environmental sensing involving camera [12] or crowdsourcing [13]–[15]. These works try to reconstruct a fine-grained model based on sparse data of meteorology, social network, traffic flow, etc. But it is hard to utilize heterogeneous data, and the model barely relies on real-time data, making the inference being possibly less efficient and precise.

To tackle limitations of existing solutions, we propose MARVAir (Fig. 1) and provide a new perspective of portable, real-time and on-site air quality inference. In this framework, smartphones are used to capture outdoor sky images, and widely available coarse weather station data serve as an optional supplementary so that the inference can be tuned. Two sub-models, i.e., the visual core and the meteorology core, are fused at decision level to produce precise inference on air quality. However, despite the promising power of involved techniques, there are still challenges lying ahead.

Challenge 1: Imprecision on outputs of smartphones. To address availability, visual data are sourced from smartphones in this work. However, their outputs are not that precise to be directly involved for later use.

Challenge 2: Utilization on heterogeneous data. Two different heterogeneous data, i.e., image and meteorology data, increase the difficulty in their effective utilization.

Challenge 3: Consumption of resource on local terminals. The data amount makes it improper to handle all the training on terminal devices, however online inference can cause unnecessary energy and communication cost.

These challenges are addressed respectively through:

• To handle Challenge 1, we propose a series of pre-processing solutions, including radiometric calibration,
feature reconstruction, to tune the devices and their outputs.

- Coping with Challenge 2, a decision-level fusion is used to combine the sub-models, so that the heterogeneous data can be used properly. Besides, outputs of the sub-models are organically combined.

- Addressing Challenge 3, we partially offload the training process on cloud, and distribute the pre-trained model to terminals. Local tuning is executed if necessary.

- For validation, we conduct extensive tests on data containing more than 30,000 outdoor images and meteorological recording pairs, which are collected at 8 sites over nearly 2 years. Results show that MARVAir notably outperforms baselines under various air conditions.

The remainder of this article is organized as follows. Section II introduces the data collection and pre-processing. Section III describes details on the implementation of our models. Evaluations are made in Section IV, and Conclusions are drawn in Section V.

II. DATA COLLECTION AND PREPROCESSING

A. The Crowdsourcing Mechanism

Since PM$_{2.5}$ concentration is an environmental property, applying crowdsourcing enables us to supplement desired fine-grained data, enhance their temporal and spatial resolution, and contain the cost [16]. We have designed a crowdsourcing mechanism to drive the data collection task. First, we hire more than two hundred volunteers to take part in this program. They are required to later perform the data collection using their own phones, within one kilometer of designated 8 interest spots. These 8 spots are denoted in noted in Table I.

In the initial stage, every valid data enrollment action was rewarded for a fixed amount of money. As the data collection process progressed, we altered the reward for the data enrollment on the spots dynamically. Volunteers got more rewards if the data enrollment happened on a spot where fewer recordings had been generated.

B. Data Collection

Two categories of data collection happen in this process: the crowdsourcing based image collection and centralized meteorological recording collection.

<table>
<thead>
<tr>
<th>Denotation</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 (MSRA)</td>
<td>Microsoft Research Asia</td>
</tr>
<tr>
<td>L2 (BUPT)</td>
<td>Beijing University of Posts and Telecommunications</td>
</tr>
<tr>
<td>L3 (BJTU)</td>
<td>Beijing Jiaotong University</td>
</tr>
<tr>
<td>L4 (PKU)</td>
<td>Peking University</td>
</tr>
<tr>
<td>L5 (BJUT)</td>
<td>Beijing University of Technology</td>
</tr>
<tr>
<td>L6 (UIBE)</td>
<td>University of International Business and Economics</td>
</tr>
<tr>
<td>L7 (ZGC)</td>
<td>Zhong Guan Cun Science Park</td>
</tr>
<tr>
<td>L8 (CGZ)</td>
<td>Che Gong Zhuang Avenue</td>
</tr>
</tbody>
</table>

1) Crowdsourcing Driven On-site Outdoor Image Collection: To collect image data on regular smartphones, a dedicated application has been developed. This application is distributed in three forms, i.e., Android, iOS, and SaaS (Software-as-a-Service) on WeChat$^1$. When the volunteer captures image, he will be guided to aim the outdoor sky, so that the sky part will be no less than 1/3 in the frame. Additional information, including time, geography coordinates, weather, and device properties, is stored automatically.

We conduct the image collection task with the Internet of Things Laboratory in Beijing University of Posts and Telecommunications. In this process, more than 200 volunteers are hired. A total amount of 31,601 images are taken using their own phones with one kilometer of designated 8 spots (Table I).

2) Meteorological Recordings Collection: Considering that macro weather activities are complex representations of a dynamic equilibrium, we believe the analysis of other gradients can contribute to the PM$_{2.5}$ inference. So, we collect 6 weather parameters, i.e., temperature, atmospheric pressure, weather type, humidity, wind speed and wind direction. Besides, the airborne concentration of NO$_2$, SO$_2$, PM$_{10}$ and CO is also recorded. The properties of relevant measurement system are listed in Table II.

We use spiderbots to implement the meteorological data collection. A total of two data sources are involved in this

$^1$A popular instant messaging application in China.
project, i.e., the National Meteorological Center [17] and the Beijing Municipal Ecological and Environmental Monitoring Center [18]. The former is used to get weather data in Beijing, and we record atmosphere composition information from the latter one. The automated data collection happens at the frequency of every quarter. We use an auto encoder based filter to check on the stored data to further mark possible anomaly recordings. Besides, we also invite a scientist that has experience in meteorology and data science to help assess and improve the quality of this workflow.

<table>
<thead>
<tr>
<th>TABLE II: Properties of the Measurement System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
</tr>
<tr>
<td>Temperature</td>
</tr>
<tr>
<td>Atmospheric Pressure</td>
</tr>
<tr>
<td>Humidity</td>
</tr>
<tr>
<td>Wind Speed</td>
</tr>
<tr>
<td>Wind Direction</td>
</tr>
<tr>
<td>PM$_{2.5}$</td>
</tr>
<tr>
<td>PM$_{10}$</td>
</tr>
<tr>
<td>CO</td>
</tr>
<tr>
<td>NO$_x$</td>
</tr>
<tr>
<td>SO$_2$</td>
</tr>
</tbody>
</table>

C. On-site Image Pre-processing

All image pre-processing happens on terminal device, therefore the visual core does not rely on Internet connection.

1) Basic Image Pre-processing: To avoid involvement of any professional instruments, we use smartphones to get on-site visual data. However, imprecision on outputs of the phone can cause the following problems:

a) Varied brightness response on different devices. Since smartphones are not professional measurement instruments, their irradiance-to-brightness response has not been calibrated in factory and can be inconsistent. As a consequence, they may generate different responses towards the same object in the real world [19].

b) User-caused impacts on image quality. We do not set requirements on when, where, or how to use the application, which may produce unwanted impacts on images. To be more specific, the pitch angle may be improper, causing vertical distortion (Fig. 2a). Besides, the lens may be occluded or stained, influencing the image quality (Fig. 2b - 2c).

To solve these problems, we propose a series of pre-processing techniques to correct the impacted images.

Perspective reconstruction based distortion correction.

To correct possible vertical distortion caused by improper pitch angle, we calculate the perspective transformation matrix $R(\alpha, \beta, \gamma)$, so as to get the projective mapping.

$$R(\alpha, \beta, \gamma) = R_z(\alpha)R_y(\beta)R_x(\gamma)$$  \hspace{1cm} (1)

where,

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ \hspace{1cm} (2)

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$ \hspace{1cm} (3)

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$$ \hspace{1cm} (4)

After applying the transformation matrix to the original image, the perspective of the image can be reconstructed.

HSI based occlusion detection and clipping. Considering the upper area of the image should be sky, the possible occlusion can deviate drastically in specific color gamut. To handle the occlusion, we propose and apply the Algorithm 1.

**Algorithm 1** Process to detect and clip occlusions.

Input: image may contain occlusions, $M$;

Output: image without occlusions, $M_{proc}$;

1: $M_{proc} \leftarrow M$, $Flag \leftarrow True$

2: while $Flag = True$ do

3: $M_{UL}, M_{UR}, M_{BL}, M_{BR} \leftarrow edge-crop(M_{proc})$;

4: for $loc = UL, UR, BL, BR$ do

5: $M_{loc, HSL} \leftarrow HSL-convert(M_{loc})$;

6: $G_{loc}^{HSL} \leftarrow gradient(M_{loc})$;

7: $G_{loc}^{HSL} \leftarrow horizontal-project(G_{loc})$;

8: $G_{loc}^{HSL} \leftarrow vertical-project(G_{loc})$;

9: if $G_{loc}^{HSL} \geq threshold$ or $G_{loc}^{HSL} \leq threshold$ then

10: $M_{proc} \leftarrow central-clip(M_{proc})$;

11: else

12: $Flag \leftarrow False$;

13: end if

14: end for

15: end while

16: return $M_{proc}$

where $edge-crop$ refers to sample the four corners of the target image.

Median filtering based stain removal. To avoid potential impacts on later inference, we use the median filter to process the images. This filter scan the entire image pixel by pixel, replacing each pixel with the median of neighboring pixels. By doing so, unwanted noise or stains can be reduced.

Radiometric calibration based brightness remapping. Considering that the EV (exposure value) setting can also be remapped.

Perspective reconstruction based distortion correction.

To correct possible vertical distortion caused by improper pitch angle, we calculate the perspective transformation matrix $R(\alpha, \beta, \gamma)$, so as to get the projective mapping.

$$R(\alpha, \beta, \gamma) = R_z(\alpha)R_y(\beta)R_x(\gamma)$$  \hspace{1cm} (1)

where,

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ \hspace{1cm} (2)

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$ \hspace{1cm} (3)

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$$ \hspace{1cm} (4)

After applying the transformation matrix to the original image, the perspective of the image can be reconstructed.

HSI based occlusion detection and clipping. Considering the upper area of the image should be sky, the possible occlusion can deviate drastically in specific color gamut. To handle the occlusion, we propose and apply the Algorithm 1.

**Algorithm 1** Process to detect and clip occlusions.

Input: image may contain occlusions, $M$;

Output: image without occlusions, $M_{proc}$;

1: $M_{proc} \leftarrow M$, $Flag \leftarrow True$

2: while $Flag = True$ do

3: $M_{UL}, M_{UR}, M_{BL}, M_{BR} \leftarrow edge-crop(M_{proc})$;

4: for $loc = UL, UR, BL, BR$ do

5: $M_{loc, HSL} \leftarrow HSL-convert(M_{loc})$;

6: $G_{loc}^{HSL} \leftarrow gradient(M_{loc})$;

7: $G_{loc}^{HSL} \leftarrow horizontal-project(G_{loc})$;

8: $G_{loc}^{HSL} \leftarrow vertical-project(G_{loc})$;

9: if $G_{loc}^{HSL} \geq threshold$ or $G_{loc}^{HSL} \leq threshold$ then

10: $M_{proc} \leftarrow central-clip(M_{proc})$;

11: else

12: $Flag \leftarrow False$;

13: end if

14: end for

15: end while

16: return $M_{proc}$

where $edge-crop$ refers to sample the four corners of the target image.

Median filtering based stain removal. To avoid potential impacts on later inference, we use the median filter to process the images. This filter scan the entire image pixel by pixel, replacing each pixel with the median of neighboring pixels. By doing so, unwanted noise or stains can be reduced.

Radiometric calibration based brightness remapping. Considering that the EV (exposure value) setting can also affect the brightness response, we calibrate the devices under various EVs and radiance. By recording the brightness response curve of the devices, the captured images can further be remapped.
2) **Dark Channel Prior Based Feature Extraction:** Airborne particles can contribute to deterioration of visibility when they absorb light arrays and cause atmospheric scattering. Considering He et al. [20] point out that dark channel can extract this feature, we deploy dark channel to complete our feature map:

\[
J_{\text{dark}}(x) = \min_{y \in \Omega(x)} \left( \min_{c \in \{r, g, b\}} J^c(y) \right) \tag{5}
\]

where \(c \in \{r, g, b\}\) is the mark of channels in RGB mode, and \(J^c\) stands for a color channel of the image. Besides, \(\Omega(x)\) is used to represent the local patch around the \(x\).

Moreover, we find that dark channel extraction can ease unwanted noise, thereby enhancing the robustness of the consecutive operations.

3) **Pseudo Three-channel Feature Map Implementation:** The backbone of the visual core requires three-channel input. To avoid computational redundancy and enhance the inference performance, we propose a pseudo three-channel feature map implementation technique to enrich the input. This process is depicted in Fig. 3, and described as follows.

a) We create an empty three-channel array with the shape of the original image. Then the first channel is replaced with the dark channel feature map.

b) We convert the original image to HSL color gamut. [10], [21] mention that, it is necessary to separate sky from other part before further operations. So, we utilize the H (hue) gamut to roughly estimate the sky part. Besides, we use the spatial information and morphological operations to refine the result. After the estimation, we use a preset threshold to binarize the image and create a mask.

c) Since haze, fog and airborne particles cause a variation in light density [11], [22]. We use L (lightness) gamut to reflect this feature. We filter the L gamut with the sky area mask, and form the second sub-channel of the feature map.

d) By doing decolorization, we then obtain a gray-scale feature map to fill the third channel. The decolorization process can be expressed as follows:

\[
m = r \times 0.11 + g \times 0.59 + b \times 0.3 \tag{6}
\]

where \(m\) refers to the obtained monochrome image; \(r\) represents the R gamut of the input picture, and similar for \(g\) and \(b\).

D. **Meteorological Recording Pre-processing**

1) **Sequentialization and data persistence:** We use a custom spider-bot to fetch the meteorological data and other airborne gradient recordings from official agencies. When the recordings are collected, we first export and combine the data to the storage. Then an anomaly detection filter is used to screen the data. Data that deviate too far from the normal range are discarded.

2) **Missing value restoration:** Due to reasons including network failure, service maintenance, anomaly filtering, etc., there are missing values in the collected data. These values are notated as NaN (Not a Number) in corresponding data field.

   We understand that those missing values should be treated with care. To address this point, we utilize a GRU-based approach to deal with those blanks in the data sequence. In our case, we take advantage of the time-domain connection in this structure, so that the complement of missing values becomes possible. The multi-in multi-out structure of the GRU network used for missing value complement is depicted in Fig. 4.

III. **SUB-MODELS AND THE FUSION CORE**

A. **Terms Explanation**

Definitions and brief concepts on terms in deep learning are listed here to facilitate readers that may have less experience in relevant field.

**CNN:** Convolution Neural Network is used to extract features from the raw input. Besides, the fully connected layers can map intermediate feature to desired output. The network can self-optimize parameters through automated learning, thus requiring less hand-engineered pre-processing.

**ResNet:** Residual Neural Network uses skip connections to maintain its trainability over a deep structure. It uses skip connections to provide skip weights, so as to address the problem of high training error or vanishing gradient.

**GRU:** Gated Recurrent Unit is a gating mechanism to maintain information from historical data and help adjust the output on current timestamp.

B. **Visual Core**

1) **ResNet50 backbone:** To automatically learn information from the customized three-channel feature map, we need a CNN-based structure for feature extraction in the visual core.

   To finish this goal, we use ResNet50 [23] as the backbone of the visual core. The structure of it is depicted in Fig. 5. We
modified the input size to $448 \times 448$, twice larger compared to the original settings.

2) **Visual core implementation**: Since the application of ResNet was different from our work, we have to modify the original backbone to fit this powerful network in our task. To finish this goal, we remove the top few layers in the original ResNet in order to adjust its mapping close to the output. Then, to utilize the generalizable feature extraction ability in the remaining structure, we add a few fully connected (FC) layers to transform the network into an airborne PM$_{2.5}$ inference model. Those layers are listed in Table III.

The newly added fully connected layers can provide necessary mapping structures for air quality detection, although the parameters of them are randomly generated and have not been trained. In detail, stacked on the remaining ResNet backbone, these cascaded FC layers can narrow down the feature map gradually. By selecting a narrow FC layer, we can reduce the amount of trainable parameters. Besides, we use a one-node FC layer with linear activation function to output the regression result about airborne PM$_{2.5}$ concentration.

During the training process, we froze most layers in the ResNet backbone to preserve the learned patterns for feature extraction. We also use mean squared error (MSE) as the loss function for model iteration.

### Table III: Top Layers of the Visual Core

<table>
<thead>
<tr>
<th>Layer Seq.</th>
<th>Layer Setting</th>
<th>Output Shape</th>
<th>Para. Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dense + ELU</td>
<td>(None, 256)</td>
<td>524 544</td>
</tr>
<tr>
<td>2</td>
<td>Dense + ELU</td>
<td>(None, 16)</td>
<td>4112</td>
</tr>
<tr>
<td>3</td>
<td>Dense + Linear</td>
<td>(None, 1)</td>
<td>17</td>
</tr>
</tbody>
</table>

C. Meteorology Core

1) **Meteorology core implementation**: The distribution of atmospheric components, including the airborne particles, are highly relevant to all kinds of atmospheric phenomena. To implement on air quality inference, we need to create a proper model, so as to build the mapping between the weather station reports to area-specific fine-grained PM$_{2.5}$ concentration. We propose a dedicated 1D-CNN model (Fig. 6) to implement the meteorology core. The deep learning structure allows us to construct a mapping relationship between the airborne PM$_{2.5}$ concentration, meteorological phenomena and other atmospheric components.

Considering the collected weather information forms a natural time-series sequence, the deployed 1D-CNN model
can help to extract features, perform the inference task, while containing the computation overhead in a low profile. We stack the information in a fixed length of timesteps to form the input data, and then use a channel attention mechanism to capture the most important part of the inputs. Sequentially, we use a series of cascaded convolutional blocks to further narrow down dimension of the intermediate layers. Each of these blocks is equipped with an FC layer, two convolutional layers, a max pooling layer, a batch normalization layer, and a dropout layer. Finally, on top of this structure, two FC layers are used to output the regression result about airborne PM$_{2.5}$ concentration.

2) Data digestion and tuning: The tuning process of this model is quite straightforward. When the sequential data are loaded into the loader and split into different batches, an overlap is deployed so that the regression on a specific moment also draw reference of the atmospheric status from previous timesteps. For the tuning part, we also choose MSE as the loss function to guide how the parameters are updated. The tuning process is depicted in Fig. 7.

**D. Decision-level Fusion and Inference**

**TABLE IV: Preliminary Test on the Sub-models**

<table>
<thead>
<tr>
<th>Sub-module</th>
<th>Precision Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual core</td>
<td>+ 5.27 %</td>
</tr>
<tr>
<td>Meteorology core</td>
<td>+ 0.00 %</td>
</tr>
</tbody>
</table>

1) Empirical studies on the performance of the sub-models: We first conduct a preliminary benchmark to learn about the performance of the two sub-models. Benchmark results of the visual core and the meteorology core on the validation dataset are presented in Table IV. From the results, the visual core outperforms the meteorology core on precision, on which a higher value implies better performance. In this framework, the data digested by the meteorology core are fetched from the weather stations. Since the atmospheric status is a complex result of dynamic equilibrium, the reports of these stations can hardly represent a fine-grained air quality status of the user's location. On the contrary, the visual core relies on the smartphone camera's output, thus provides a more accurate result. The result suggests that, a mechanism to utilize the output from both the two models, is necessary.

2) Decision-level fusion: To process the heterogeneous data in this work, we use the aforementioned two sub-models to construct an organically combined fusion core. To accomplish this idea, we use decision-level fusion to integrate these two sub-models into a unified framework. The structure to combine these two sub-models is illustrated in Fig. 8, and the fine-tuning process can be denoted as follows.

![Fig. 6: The structure of the meteorology core.](image)

![Fig. 7: The tuning process of the meteorology core.](image)

![Fig. 8: Fusion of the two sub-models.](image)

Given a dataset containing $m$ samples, we then use $V(m)$ and $M(m)$ to represent the outcome from the visual core and the meteorology core. To fuse these two outcomes, we first use a series of 1D convolutional kernel to upscale the dimension and concatenate them. The output here can be denoted as $[\omega_1 V(m) \mid \omega_2 V(m) \mid \cdots \mid \omega_n V(m)] \sim V_n$ and $[\omega_1 M(m) \mid \omega_2 M(m) \mid \cdots \mid \omega_n M(m)] \sim M_n$, where $n$ would be the number of the kernels used to finish the upscale. We also denote the distance the location of the record and its nearest weather stations as $D$, with $D$ being defined as:

$$D = \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2} \quad (7)$$

where $x_s$ stands for the longitude coordinate of the nearest weather station, and $y_r$ means the latitude coordinate of the record.

Similarly, we feed the data into two convolutional layers and a series of fully connected layers for fuse $V_n$, $M_n$ and $D$. This process can be described as adding weights and bias to convoluted input values, and we used MSE to reduce the difference between this output and the ground truth of airborne PM$_{2.5}$ concentration. The final goal can be described as:

$$\min MSE(\sigma(\omega_1 V_n + \omega_2 M_n + \omega_3 D + \beta_{sum}), \text{GT}) \quad (8)$$

where $\text{GT}$ represents the ground truth of the airborne PM$_{2.5}$ concentration.

**E. Cloud-side Accelerated Model Training**

In this work, we use smartphones to implement the inference workflow. It is obvious that, operations that can cause huge
computation burden should be restrained on these resource-constrained devices. To further optimize local resource consumption on terminals, we use a cloud-side manner to assist the model training process, as shown in Fig. 9.

![Cloud-side model training diagram]

Fig. 9: Cloud-side model training.

First, considering cloud servers are equipped with powerful computing resources, we use the cloud to accelerate the workflow. Based on the full-volume data, GPU (Graphics Processing Unit) clusters on the cloud are used to train the visual core and the meteorology core. Once these two sub-models are ready, decision-level fusion is applied to form the fusion core. Then, the fusion core need to be sent to terminals. There are several solutions for this, including over-the-air distribution via Wi-Fi or cellular network, preloaded while the device is charging, or the model can be pre-installed at the factory.

IV. Evaluation

A. Settings

1) Data and AQI Standard: We conduct data collection during May 2015 to March 2017 on 8 spots (Table I). The following preparation has been done during this period.

a) Outdoor sky images. A total amount of 31 601 outdoor sky images are taken.

b) On-site information. Time, geography coordinates, weather, and device properties are recorded along with the images.

c) Meteorological data. The meteorological data, including the atmospheric phenomena and the airborne concentration of NO$_2$, SO$_2$, PM$_{10}$ and CO, are recorded from two resources [17], [18] by spider-bot at the frequency of every quarter.

d) Device calibration. Up to 253 different off-the-shelf phones are used in this work. All involved devices are calibrated so that the mapping between irradiance to brightness can be adjusted, and relevant data are stored.

In this work, we use the current local regulation on AQI [24], and the details are presented in Table V.

2) Evaluation Indices: Considering that the likelihood of different weather conditions may differ drastically, the distribution of the data can be uneven. Consequently, the index of accuracy cannot function as a general performance measurement. To address this problem, we use a more general system to suggest the classification performance.

<table>
<thead>
<tr>
<th>AQI Level</th>
<th>AQI Categories</th>
<th>Concentration of PM$_{2.5}$ (µg/m$^3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 50</td>
<td>Excellent</td>
<td>[0 - 35]</td>
</tr>
<tr>
<td>51 - 100</td>
<td>Good</td>
<td>[35 - 75]</td>
</tr>
<tr>
<td>101 - 150</td>
<td>Lightly Polluted</td>
<td>[75 - 115]</td>
</tr>
<tr>
<td>151 - 200</td>
<td>Moderately Polluted</td>
<td>[115 - 150]</td>
</tr>
<tr>
<td>201 - 300</td>
<td>Heavily Polluted</td>
<td>[150 - 250]</td>
</tr>
<tr>
<td>&gt; 300</td>
<td>Severely Polluted</td>
<td>[250 - 500]</td>
</tr>
</tbody>
</table>

1) **Precision.** The fraction of relevant instances among the retrieved instances.

2) **Recall.** The fraction of relevant instances that were retrieved.

3) **F1 Score.** The harmonic mean of the precision and recall.

B. Overall Comparison with Baselines

We first use full-volume data to conduct the overall comparison. During this part, the dataset is split in a ratio of 7:3, so as to form the training set and the test set. In this process, the distribution of the recordings, no matter concerning the temporal or spatial one, has not been manually adjusted. The two existing works that serve as the baselines are:

1) **U-Air.** A work proposed by Zheng et al. in [9], which leverages the use of temporal and spatial classifier.

2) **MSH-WSN.** A work [11] providing fine-grained air quality monitoring service in urban areas, which utilizes wireless sensor networks and other information sources.

We demonstrate the corresponding performance indices using a category-specific manner in Table VI and Fig. 10. The results suggest **MARV Air** outperforms all the baselines under all the AQI categories, i.e., a 7.62 % and a 25.19 % higher F1 Score than [11] and [9] respectively. We can also notice that, **MARV Air** can provide a much more precise inference for lightly and moderately polluted air status, which can provide useful information to the user.

<table>
<thead>
<tr>
<th>Solutions</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Air</td>
<td>0.810</td>
<td>0.782</td>
<td>0.790</td>
</tr>
<tr>
<td>MSH-WSN</td>
<td>0.923</td>
<td>0.920</td>
<td>0.919</td>
</tr>
<tr>
<td>MARV Air</td>
<td>0.988</td>
<td>0.990</td>
<td>0.989</td>
</tr>
</tbody>
</table>

C. Sub-model Benchmark

In the **MARV Air**, the fusion core consists of two heterogeneous, but organically incorporated sub-models, i.e., the visual core and the meteorology core. We conduct tests to compare the standalone inference performance of these two sub-models respectively. The data used in this experiment are also randomly split in a ratio of 7:3, and the results are demonstrated in Table VII.

All the three indices in the results suggest that, the visual core outperforms the meteorology core. Considering that the meteorological recordings are fetched from weather stations,
those data can only represent the actual atmospheric status of the place where the station is located. Therefore, there are deviations in the inference result of the meteorology core. On the other hand, the visual core directly digests on-site images, thus being able to provide more fine-grained inference on air quality.

D. Category-specific Performance of Fusion Core

Considering that the likelihood of different weather conditions may differ drastically, e.g., the chance of AQI level 2 (Good) is considerably higher than the one of AQI level 6 (Severely Polluted), the distribution of the data can be uneven. Consequently, an excellent overall performance on the full-volum dataset does not guarantee the performance on specific air conditions. The result for this experiment is demonstrated in Fig. 10c.

It can be noticed, that *MARVAir* outperforms all the baselines under all the AQI categories. In detail, for the 3 air conditions that are easily overlooked, i.e., Good, Lightly Polluted and Moderately Polluted, the proposed solution has an increase of around 12 % compared to Feng et al. [11]. Besides, the proposed solution also achieves an average F1 Score of 0.99 % on the rest air conditions.

E. Scalability of *MARVAir*

In this part, we conduct two experiments to test the scalability of *MARVAir* under different conditions.

1) Availability on unobserved locations: During the data collection, the image data are gathered within one kilometer of designated 8 interest spots in Table I. To simulate the situation that the user being on an unobserved location, so the model need to use data from the nearest available spots, we design a special experiment. In this experiment, the dataset is split in a spatial manner. We use the data from 7 sites, and perform PM$_{2.5}$ inference on images that are taken on the left one location. The results are depicted in Fig. 11.

It is obvious that, *MARVAir* is able to deliver precise PM$_{2.5}$ inference, even when the target location has not been adopted in the training set.

2) Availability when Internet is not reachable: We would like to push the experiments to a more extreme condition, i.e., on which the user has constantly been unreachable to Internet. This test can validate the availability of *MARVAir* when the current meteorological data have been long unattainable, so that the model can only rely on cached history data. To implement this test, we split the dataset in a temporal approach, i.e., use the data from the first three weeks in each month to form the training set, and use the data from the later week(s) to form the validation set. The results are demonstrated in Table VIII.

E. Scalability of *MARVAir*

In this part, we conduct two experiments to test the scalability of *MARVAir* under different conditions.

1) Availability on unobserved locations: During the data collection, the image data are gathered within one kilometer of designated 8 interest spots in Table I. To simulate the situation that the user being on an unobserved location, so the model need to use data from the nearest available spots, we design a special experiment. In this experiment, the dataset is split in a spatial manner. We use the data from 7 sites, and perform PM$_{2.5}$ inference on images that are taken on the left one location. The results are depicted in Fig. 11.

It is obvious that, *MARVAir* is able to deliver precise PM$_{2.5}$ inference, even when the target location has not been adopted in the training set.

2) Availability when Internet is not reachable: We would like to push the experiments to a more extreme condition, i.e., on which the user has constantly been unreachable to Internet. This test can validate the availability of *MARVAir* when the current meteorological data have been long unattainable, so that the model can only rely on cached history data. To implement this test, we split the dataset in a temporal approach, i.e., use the data from the first three weeks in each month to form the training set, and use the data from the later week(s) to form the validation set. The results are demonstrated in Table VIII.

E. Scalability of *MARVAir*

In this part, we conduct two experiments to test the scalability of *MARVAir* under different conditions.

1) Availability on unobserved locations: During the data collection, the image data are gathered within one kilometer of designated 8 interest spots in Table I. To simulate the situation that the user being on an unobserved location, so the model need to use data from the nearest available spots, we design a special experiment. In this experiment, the dataset is split in a spatial manner. We use the data from 7 sites, and perform PM$_{2.5}$ inference on images that are taken on the left one location. The results are depicted in Fig. 11.

It is obvious that, *MARVAir* is able to deliver precise PM$_{2.5}$ inference, even when the target location has not been adopted in the training set.

2) Availability when Internet is not reachable: We would like to push the experiments to a more extreme condition, i.e., on which the user has constantly been unreachable to Internet. This test can validate the availability of *MARVAir* when the current meteorological data have been long unattainable, so that the model can only rely on cached history data. To implement this test, we split the dataset in a temporal approach, i.e., use the data from the first three weeks in each month to form the training set, and use the data from the later week(s) to form the validation set. The results are demonstrated in Table VIII.

E. Scalability of *MARVAir*

In this part, we conduct two experiments to test the scalability of *MARVAir* under different conditions.

1) Availability on unobserved locations: During the data collection, the image data are gathered within one kilometer of designated 8 interest spots in Table I. To simulate the situation that the user being on an unobserved location, so the model need to use data from the nearest available spots, we design a special experiment. In this experiment, the dataset is split in a spatial manner. We use the data from 7 sites, and perform PM$_{2.5}$ inference on images that are taken on the left one location. The results are depicted in Fig. 11.

It is obvious that, *MARVAir* is able to deliver precise PM$_{2.5}$ inference, even when the target location has not been adopted in the training set.

2) Availability when Internet is not reachable: We would like to push the experiments to a more extreme condition, i.e., on which the user has constantly been unreachable to Internet. This test can validate the availability of *MARVAir* when the current meteorological data have been long unattainable, so that the model can only rely on cached history data. To implement this test, we split the dataset in a temporal approach, i.e., use the data from the first three weeks in each month to form the training set, and use the data from the later week(s) to form the validation set. The results are demonstrated in Table VIII.

### Table VII: Performance Comparison on Sub-models

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>F1 Score Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Core</td>
<td>0.978</td>
<td>0.975</td>
<td>0.977</td>
<td>+ 5.39 %</td>
</tr>
<tr>
<td>Meteorology Core</td>
<td>0.929</td>
<td>0.928</td>
<td>0.927</td>
<td>+ 0.00 %</td>
</tr>
</tbody>
</table>

### Table VIII: Evaluation on Offline Mode

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion Core</td>
<td>0.897</td>
<td>0.897</td>
<td>0.883</td>
</tr>
</tbody>
</table>

It can be seen that, even there is a slight decline in performance, *MARVAir* can still deliver reasonable inference quality when the system is running on an offline mode.

V. Conclusion

We propose *MARVAir*, a crowdsourcing based hybrid air quality sensing system. It utilizes the outdoor sky images taken on regular smartphones along with meteorological recordings from the nearest weather station to deliver on-site real-time air quality inference. We implement this framework by designing two sub-models so that heterogeneous data are used effectively. To deal with imprecision on outputs of smartphones, we propose a customized workflow to rectify the captured images. Based on the two sub-models, a decision-level fusion is deployed to generate the fusion core. Experimental results
suggest that MARVAir can deliver a portable and precise solution for real-time PM$_{2.5}$ on-site detection.

However, we also notice spaces for further improvements. First, the size of the sub-model can be compressed to accelerate model downloading during the first initiation. Besides, when the user is on an unobserved location, how to effectively utilize information from the nearest spot can be further explored.

REFERENCES


[24] (2021) Technical regulation on ambient air quality index (on trail) - ministry of ecology and environment of the people’s republic of china (former ministry of environmental protection of the people’s republic of china).