Decision Analytics Journal 8 (2023) 100283

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

An end-to-end pollution analysis and detection system using artificial intelligence and object detection algorithms

Md. Yearat Hossain [a], Ifran Rahman Nijhum [a], Md. Tazin Morshed Shad [a], Abu Adnan Sadi [a], Md. Mahmudul Kabir Peyal [b], Rashedur M. Rahman [a,*]

[a] Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
[b] Department of Electrical and Electronic Engineering, Brac University, Dhaka, Bangladesh

ARTICLE INFO

Keywords:
Artificial intelligence
Pollution detection system
Visual pollution
Environmental management
Machine learning
Data mining

ABSTRACT

Environmental pollution is generally a by-product of various human activities. Researchers have studied the dangers and harmful effects of pollutants and environmental pollution for centuries, and many necessary steps have been taken. Modern solutions are being constantly developed to tackle these issues efficiently. Visual pollution analysis and detection is a relatively less studied subject, even though it significantly impacts our daily lives. Building automatic pollution or pollutants detection systems has become increasingly popular due to the modern development of advanced artificial intelligence systems. Although some advances have been made, automated pollution detection is not well-researched or fully understood. This study demonstrates how various object detection models could identify such environmental pollutants and how end-to-end applications can analyze the findings. We trained our dataset on three popular object detection models, YOLOv5, Faster R-CNN (Region-based Convolutional Neural Network), and EfficientDet, and compared their performances. The best Mean Average Precision (mAP) score of 0.85 was achieved by the You Only Look Once (YOLOv5) model using its inbuilt augmentation techniques. Then we built a minimal Android application, using which volunteers or authorities could capture and send images along with their Global Positioning System (GPS) coordinates that might contain visual pollutants. These images and coordinates are stored in the cloud and later used by our local server. The local server utilizes the best-trained visual pollution detection model. It generates heat maps of particular locations, visualizing the condition of visual pollution based on the data stored in the cloud. Along with the heat map, our analysis system provides visual analytics like bar charts and pie charts to summarize a region's condition. In addition, we used Active Learning and Incremental Learning methods to utilize the newly collected dataset by building a semi-autonomous annotation and model upgrading system. This also addresses the data scarcity problem associated with further research on visual pollution.

1. Introduction

Introducing toxins into a natural environment that reduces its suitability for human habitation is called pollution. Fundamentally, we all need to be aware of the concept of the environment because all living creatures depend on it to survive, making it challenging to ignore without taking care of it. Although the environment can persist in its natural state, human interference has severely damaged many ecosystems. These activities have caused a significant pollution issue, which has disturbed the planet's atmosphere [1–3]. The indoor and outdoor environments are affected by pollution, which has long been a public concern. Several different kinds of pollution have been identified. Other than the most well-known types of pollution, such as air, land, and water pollution, the science of environmental pollution lists a number of others that have a marginal but important impact on us. Such pollution is visual pollution, which affects how we see our surroundings. Our first impression of a society is formed by its appearance, frequently a patchwork of man-made buildings and natural structures. All irregular structures that are unappealing and out of context prevent people from enjoying their surroundings. Visual pollution is defined as all the unsightly objects that are out of context to their surroundings and ruin the aesthetic quality of the surrounding landscape, which causes harm to human vision and health [4]. Visual pollution can be caused by anything that obscures gorgeous sights. Garbage thrown in various locations, cables or wires hanging over streets, dumped construction materials in urban areas, ill-arranged and bright billboards, old decaying objects, utility poles, and so on [5,6]. This is a problem that is associated especially in the urban areas, which lower the quality of life and may cause health issues ranging from health disorders to emotional distress.

* Correspondence to: Department of Electrical & Computer Engineering, North South University, Bashundhara, Dhaka 1229, Bangladesh.
E-mail address: rashedur.rahman@northsouth.edu (R.M. Rahman).

https://doi.org/10.1016/j.dajour.2023.100283
Received 21 January 2023; Received in revised form 9 April 2023; Accepted 4 July 2023
Available online 7 July 2023

Visual blight and visual congestion are two phrases that come up frequently in this context. Billboards, power wires, and unsightly structures are examples of visual blight. Visible congestion can be found in everyday life, such as a cluttered workstation or a congested roadway. It may hinder a person's ability to locate particular objects in such situations and someone we are attempting to meet on the street. One of the causes of visual congestion can be attributed to administrative neglect. Without proper rules and regulations set by the governments, advertising companies showcase their advertisements in almost every corner of public places without considering the effects it may have on the people [7]. These commercials overcrowd the areas with unkempt, crooked, and uneven objects like billboards and pamphlets, destroying the surrounding environment to the point when it is difficult to identify the place. Although commercials are meant to enlighten consumers about a variety of things that may benefit them in their daily lives, the moment that they start to proliferate in our everyday environments is when they start to pose a problem [6]. Visual pollution caused by disorderly, damaged, uneven, and massive promotions can have far-reaching and widespread implications. Distraction, loss of identity, traffic congestion, various forms of health concerns, annoyance and psychological illnesses, eye strain, loss of feeling of sanitation and aesthetics, loss of politeness, and general loss of the resident community's quality of life are among them [8,9]. Visual pollution degrades the natural view, and as a result, urban areas around the city lose their uniqueness as a location. Visually calming environments, such as vast fields, stunning landscapes, forests, hills, greenery, and so on, help re-energize us, relieve our pains, and restore our productivity. According to [10], our stress levels may be influenced by our surroundings, which can affect our bodies. Our neurological, endocrine, and immune systems' functioning are all affected by what we are seeing, hearing, and experiencing at any moment. Unpleasing environments can give rise to an anxious feeling, which may increase our blood pressure and heart rate. A pleasing environment reverses that. Hence it is crucial that we preserve the natural balance within our environment.

Environmental pollution is not a new phenomenon, yet it continues to be the biggest threat to mankind and one of the key factors contributing to human life. Urbanization, industrialization, mining, and exploration are some human activities contributing to global environmental contamination. Together, developed and developing countries bear this responsibility. Although developed countries have made a greater contribution to environmental protection due to increased knowledge and tighter legislation, there are still issues yet to be addressed regarding environmental pollution in third-world countries like Bangladesh, India, and Pakistan. Despite the increased awareness of pollution worldwide, its negative long-term effects are still being noticed. The areas of air pollution, water pollution, and soil contamination have received a lot of attention and have been the center of study for numerous kinds of research [11–14]. Moreover, new techniques involving the usage of Artificial Intelligence to conduct research have emerged in the 21st century [15–18]. Deep Learning is one of the disciplines of Artificial Intelligence or AI that receives the most attention. Deep Learning essentially extracts high-level information from data and makes decisions based on the data, which can be easily interpreted by humans.

Although visual pollution is a serious issue, very few efforts have been done on the subject that makes use of cutting-edge AI methods. AI-based systems can offer solutions to effectively address the problems caused by visual pollutants because the idea of visual pollution is not fully understood by the general public. To find the things that produce visual pollution, deep learning-based approaches like object detection and image classification can be used. These models can then be used to create automated systems that assist the authorities in monitoring and taking the appropriate action to safeguard the environment from such contaminants. Even though the definition of visual pollution remains the same, the objects that cause visual pollution can vary from place to place. Furthermore, they frequently go unnoticed despite being in plain view. Therefore, in order to address these kinds of environmental concerns, automated systems based on AI are required. Additionally, little research exists regarding the proper design and upkeep of these automated systems for the management of waste or pollution. As the types of pollutants might change over time, developing systems that can automatically identify and provide insights on pollution control can be quite difficult. Although there have been some works performed on visual pollution classification and detection, it has not yet been suggested how to use the models to create a system to analyze the pollution [19,20]. It is also little understood how these kinds of data might be gathered, handled, and ultimately used for developing the systems. Additionally, it is crucial to comprehend how the recently trained models might be used for pollution analysis. Therefore, research into creating automated systems using machine learning and deep learning approaches that can quickly adapt to the increasing need for pollution management is required.

In our work, we utilized modern computational algorithms like deep learning to solve an important concern of environmental pollution which is visual pollution. As this sub-field of environmental pollution lacks works that utilize modern computational techniques, we directly jump into how object detection algorithms can be used to automatically detect visual pollutants and how such kind of automated systems can be utilized to enhance the quality of the environment that we all live in. We did a comprehensive study of visual pollution detection using three popular object detection algorithms and built an end-to-end application to demonstrate the usability of such a system. The system also includes a crowd source-based data collection system through an Android application using the cloud. The system can run analyses on the collected dataset with the help of the trained model and display a heat map of visual pollutants on the map along with other information that summarizes the overall condition of a given region. We have also employed Active and Incremental learning to upgrade the model efficiently with the help of the newly collected dataset over the time. These two machine learning methods plays important roles for developing modern deep learning-based systems as such systems require continuous development and deployment and human supervision is not always possible.

Our main contributions are summarized as follows:

- We experimented with the three most popular object detection models, namely, YOLOv5, Faster R-CNN, and EfficientDet, and performed a comparative analysis of their performance in visual pollution detection.
- We developed an end-to-end system, where we deployed our best-trained model to experiment with the usability of such a system in a real-world use case.
- We developed an Android application using which users could capture and upload images containing visual pollutants along with their locations that generate new datasets.
- We utilized the collected dataset to visualize and analyze the impact of visual pollution in a geospatial manner with the help of our model and heat map.
- We also used Active and Incremental Learning methods to manage and upgrade both the dataset and the model in the system efficiently.

2. Related works

With the rapid advancement of deep learning, researchers are focusing their efforts on finding ways for applying deep learning to real-world situations. Solving problems related to environmental management has also been the primary focus of many studies. The authors of [21] highlighted the use of artificial intelligence in environmental sustainability, artificial intelligence in the reduction of air and water pollution, and potential artificial intelligence-based techniques in various industrial sectors. A similar research paper [22] evaluated different techniques for handling environmental challenges in pollution. The authors emphasized the characteristics, advantages, and limitations of

single AI and hybrid AI techniques in areas of environmental pollution such as water pollution.

In their work [23], presented a deep learning-based approach to smart waste management. They used the YOLOv3 algorithm, which was trained to detect 6 different classes of trash objects. Waste segregation can be aided by models like this, which can assist in efficiently recycling and disposing of waste materials. To forecast groundwater arsenic contamination, [17] used a machine learning-based technique. This study was carried out in India, and the researchers used machine learning techniques to map out the areas with the highest levels of groundwater contamination due to arsenic. Their prediction map reveals which areas should get priority attention from policymakers for future testing campaigns and preventive initiatives. In [16], the author used a deep learning network to predict river pollution under the impact of rainfall-runoff. In their study, they identified the dry period as the most significant factor affecting river pollution, followed by average rainfall intensity, maximum rainfall in 10 min, the total amount of rainfall, and initial runoff intensity. Based on the relationships between rainfall characteristics and event mean concentration, they used an artificial neural network to predict the event mean concentration of Chemical oxygen demand in the river. In a paper, [24], proposed an object detection model named AquaVision for detecting and categorizing various pollutants and hazardous waste floating in the waters. The proposed method could locate waste objects, which assists in the cleaning of surface waters and contributes to environmental protection by preserving the aquatic habitat. A. Nazerdeylami et al. used a Deep Neural Network (DNN) model to identify objects in seaside areas in their study [25]. This model was used to detect man-made pollutants and hazardous objects. The seaside scenes were semantically labeled using a pre-trained VGG architecture and for object detection, the Single Shot Detector (SSD) approach was used. An aggregated LSTM (ALSTM) based on a well-liked deep learning technique LSTM was proposed by the authors in [18]. They have incorporated neighborhood air quality monitoring stations, stations in close proximity to industrial regions, and stations for outside pollution sources in this new proposed methodology. Their findings demonstrated that the suggested approach might be utilized successfully for forecasting air pollution. A similar research paper [26], introduced a deep-learning solution for predicting the air quality index of the city of Chennai. The AQI values were classified using a deep learning model based on a combination of Support Vector Regression (SVR) and Long Short-Term Memory (LSTM). This study demonstrated how deep learning might raise public awareness about air pollution and assist officials in taking the required steps to improve air quality. Another study [27], utilized deep learning to detect pollution caused by vehicles. The image recognition model Inception-v3 was used in this study.

The idea of the significance of visual quality in the built environment was covered in depth by Portella in her book [6]. The researchers' representation of the detrimental effects that commercial signage may have on urban areas' aesthetic appeal and, moreover, on people's quality of life, was done from the perspectives of architecture, urban planning, and psychology. Visual pollution is a subjective issue, making its identification and assessment more difficult than other types of pollution recognized. The manual assessment and quantification of visual pollution utilizing color images, public surveys, and geospatial technology have all been studied. To solve the problem of measuring visual pollution, the authors of [28] have used a traditional cumulative area approach. The authors used a picture booklet survey to collect responses from people in an architectural urban zone of Kuala Lumpur, Malaysia. The results are based on the respondents' greater tolerance levels, which helped identify the visual contaminants when combined with demographic factors including gender, education level, and home location. Another research, [29] attempted to look at the environmental issues, particularly the visual pollution brought on by billboards and advertisements, as well as its potential remedies in the Serbian historical city of Ni. The goal of the study was to identify the complexity, challenges, and significant ramifications of aesthetically offensive things. The article used an inductive method to assess this kind of pollution, based on a survey of urban residents' perceptions of visual pollution.

Researchers in the field of deep learning have just lately been interested in visual pollution. The majority of studies used publicly available information to identify and classify visual pollution. The authors of the research [19] introduced a deep learning model for categorizing visual contaminants. This study looked at four different types of visual pollution, and they gathered their data using the Google image search engine. The total number of photos in their final dataset was 800. As a deep learning model, the study suggested a Convolutional Neural Network (CNN) architecture. Finally, they were able to attain a 95 percent training accuracy and an 85 percent testing accuracy for visual pollution categorization. Similar work was done in the study [30] where deep learning networks were used to classify textile visual pollutants. The data was originally gathered by the authors through search engine crawling, as well as local clothing factories, roadside sellers, and shopping centers. After annotation, the authors experimented with different deep learning networks, YOLOv5, EfficientDet, and Faster R-CNN. Through their experiments, they have concluded EfficientDet as being the best model achieving the highest accuracy for their case of study. The authors of the research [20] introduced the usage of deep-learning techniques for detecting visual contaminants from natural scenarios. They gathered their data using the image from Google Street View. They were able to accumulate a total of 1400 images through their method of data collection. Using a deep learning approach based on YOLOv5, the authors identified six distinct visual pollutants in Dhaka, Bangladesh. A computer vision annotation tool called CVAT was used to manually annotate 1400 photos that the author had manually collected. The final mAP score achieved by the authors was 0.80. Compared to this work, our work proposes a visual pollution analysis and detection system through a comprehensive study of multiple object detection models. Instead of trying out a single model, our work performs a comparative analysis of visual pollution detection on multiple models and then utilizes the best performed model to develop a system that can provide analyses on visual pollution. Also, our proposed work demonstrates how these type of data can be collected and utilized efficiently using modern deep learning based methods and proper software design.

So far it has also been noticed that there is a lack of datasets that can help researchers to research and develop more sophisticated systems regarding visual pollution. In all the works related to visual pollution with deep learning, the authors had to manually collect and label or annotate their dataset which makes the whole process much more difficult for the researchers. In our work, we also proposed a system that uses a crowdsource-based approach to collect and continuously utilize the collected dataset for real-world applications of visual pollution detection. For this, an Android application was developed through which a small to a large group of people can submit images that might contain visual pollutants. These data are efficiently stored in the cloud and then utilized by the trained detection models. The models are deployed in the system in a way that they are used for multiple purposes at the same time. First, they are used to run inferences on the incoming datasets and provide analyses on given geo-locations. Second, their confidence scores are utilized to generate a new larger dataset for further development of the system. The model utilizes the concept of Active Learning to label and store new data that they are confident about and leaves the other data that might require some human intervention. Later these newly generated datasets are used to retrain the existing model using another technique called Incremental Learning. This method aids the model's ability to learn new data while maintaining its memory of previously learned data. The proposed system is not only limited to visual pollution management but also any kind of pollution and waste management that can be detected from images.

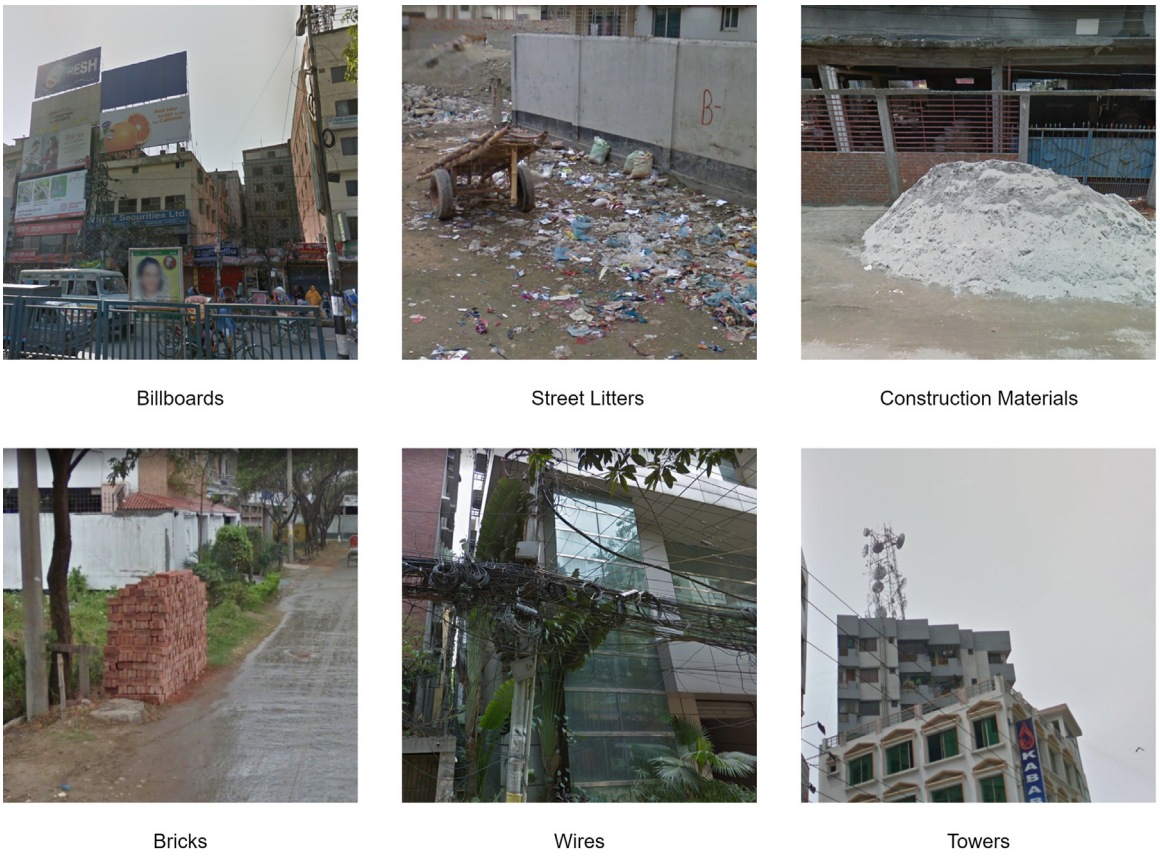

Fig. 1. Samples from each class presented in our dataset. The dataset represents 6 different kinds of visual pollutants that can be seen around the roads in Dhaka city.

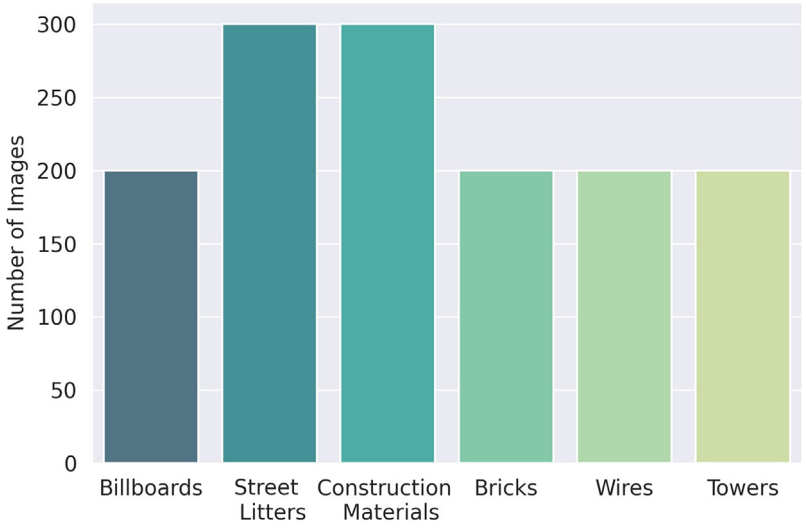

Fig. 2. Per class distribution of the dataset we used for our experiments. The class 'Street Litters' and 'Construction Materials' contain 300 images per class while the others contain 200 images each.

3. Methodology

3.1. Dataset

We did not have access to a dataset suitable for detecting visual pollution. As a result, we extended the utilization of the same dataset used in the study [20]. The dataset was created by collecting screenshots from Google Street View around various locations in Dhaka city. Fig. 1 illustrates some of the sample images from the dataset. The authors manually pre-processed and annotated this dataset. The dataset is divided into six categories of common visual pollutants: "Billboards", "Street Litters", "Construction Materials", "Bricks", "Wires", and "Towers". In total, there are 1400 photos in the dataset. The per-class image distribution of the images in the dataset is shown in Fig. 2. All these images were first resized into 500 × 500 pixels and later annotated using the CVAT annotation tool. To annotate the images, the authors used the rectangular bounding box technique. Because this dataset was designed primarily for object detection, several of the images featured

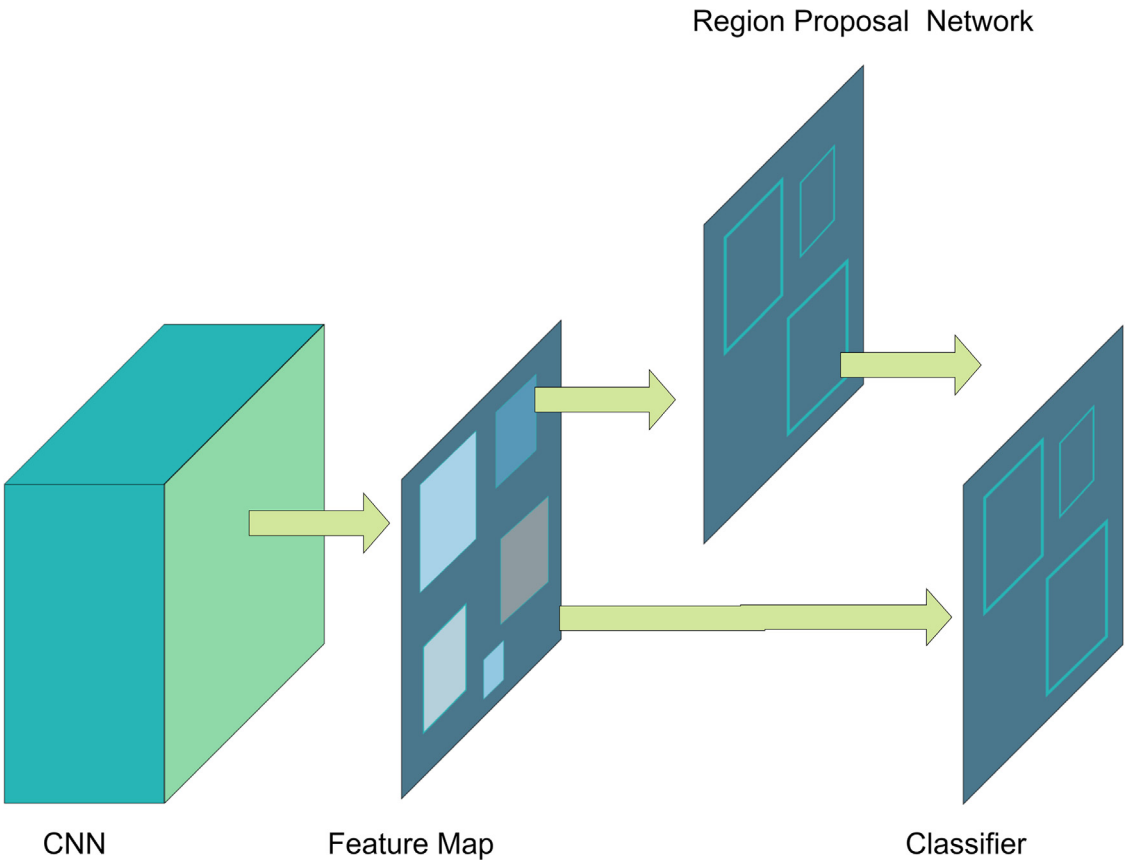

Fig. 3. Architecture of Faster R-CNN.

multiple visual pollutant types in a single image. In such cases, different bounding boxes were created to annotate each visual pollutant object present in the image. These annotations were then converted to a format required by the respective models.

3.2. Models

3.2.1. Faster R-CNN

Faster R-CNN is a widely used region-based neural network for object detection. It has derived from the Fast R-CNN, which is derived from another model called R-CNN. R-CNN or Region-based Convolutional Neural Networks are a family of neural networks which mainly work by proposing regions of objects in a given image using an algorithm called selective search. Selective search helps the model extract the region of interest where the objects can be in a picture. Region of interest can be thought of as drawing boxes around the regions, also known as bounding boxes. The first R-CNN model was introduced by [31], and it used selective search as a combination of exhaustive search on color-segmented parts of the image. Initially, the algorithm generates small regions of interest. Eventually, a greedy algorithm combining sections of a similar color increases the size of the regions.

The similarity between the regions can be calculated by: $\boldsymbol{S}(\boldsymbol{a},\boldsymbol{b}) = \boldsymbol{Stexture}(\boldsymbol{a},\boldsymbol{b}) + \boldsymbol{Ssize}(\boldsymbol{a},\boldsymbol{b})$, where the $\boldsymbol{Stexture}(\boldsymbol{a},\boldsymbol{b})$ is visual similarity and $\boldsymbol{Ssize}(\boldsymbol{a},\boldsymbol{b})$ is similarity between the regions. Later the extracted regions are passed into a CNN to extract the features from the image patches. As a CNN model expects a fixed size of the input, the region patches are resized into a fixed size first, also known as warping. CNN feature extractor extracts the features from an image by identifying shapes, structures, textures, etc from the given input, and these are also known as feature maps. Later these feature maps are sent to a Support Vector Machine (SVM) classifier which finally classifies the object. But this entire process was slow, and the model was not end-to-end trainable. To improve the existing model, Fast R-CNN arrived, which promised better performance and speed compared to the first R-CNN model [32]. Instead of extracting CNN features from all the selected regions of interest, Fast R-CNN proposed generating the feature map of the entire image using a CNN feature extractor at the very beginning of the model. This entire feature map is then fed into a Region of Interest or ROI pooling layer, which generates regions of interest from the image. These pooled features are then sent to two sections, one which classifies the object inside the pooled region and one which corrects the bounding box coordinates using a regression algorithm. Even though Fast R-CNN was faster and more efficient than R-CNN, it was still not end-to-end trainable. Faster R-CNN arrives with a novel region proposal method which gives it advantages over the other models [33]. Faster R-CNN uses a Region Proposal Network or RPN, a CNN for generating regional proposals. Like Fast R-CNN, Faster R-CNN has a CNN feature extractor that first generates feature maps from a given image. These CNN feature extractors are already pre-trained for extracting features like shape, structures, textures, colors, etc., from images. After extracting the features from the first CNN, the feature map is given to the RPN as input which outputs the region proposals for the ROI layer. Later, the ROI layer sends the outputs to the classifier and the bounding box regressor, just like in Fast R-CNN. In contrast to earlier models, which relied on selective search to create regions of interest, Faster R-CNN employs a brand-new technique called the region proposal network, enabling it to detect objects from images more accurately while using less processing power. Fig. 3 represents a high-level diagram of the Faster R-CNN architecture.

Our research utilizes a Faster R-CNN model with a Resnet101 CNN as its feature extractor. The feature extractor is already pre-trained on the ImageNet dataset [34], hence good at detecting useful features from an image. ResNet is a popular CNN model introduced by [35], and since then, it has been used in various applications. The ResNet CNN architecture was mainly introduced from the notion that if we make a

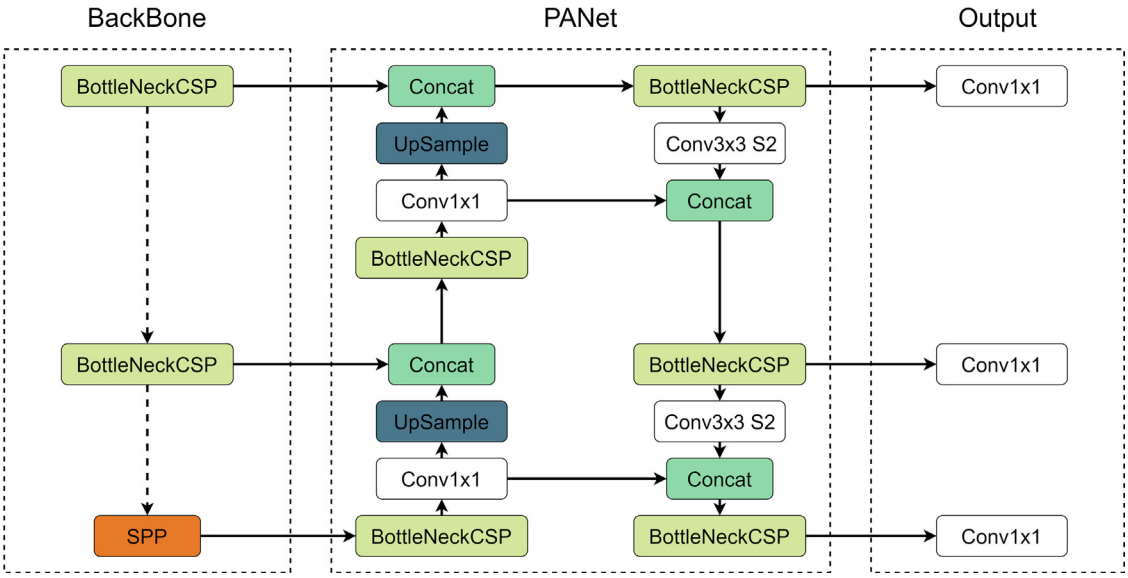

Fig. 4. YOLOv5 architecture.

model denser, the better accuracy we will get but as we make a model denser, the images become smaller due to the various convolutional operations. To tackle this issue, Resnet arrived with residual blocks with bottleneck layers that perform connection skipping and maintain a learnable image size throughout the flow of the dense model, resulting in better performance.

3.2.2. YOLO

"You Only Look Once" or YOLO is another widely used object detection model which is very popular for its speed and performance [36]. Unlike other object detection models like Fast R-CNN or Faster R-CNN, YOLO handles the entire task in a single CNN model and handles the problem of object detection like a regression problem. YOLO is an end-to-end model that looks at the entire input image and outputs vectors that represent the position of bounding boxes, confidence scores of the objects inside the bounding boxes, and class probabilities of the objects. First, each image is divided into $\boldsymbol{S}\times\boldsymbol{S}$ grid cells, and $\boldsymbol{B}$ number of bounding boxes is calculated for each grid. Each box outputs five values, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{w}$ and $\boldsymbol{h}$, along with the confidence score of the object. Non-max suppression is used to eliminate overlapping bounding boxes. Finally, the results are merged, and final bounding box coordinates are achieved from $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{w}$, and $\boldsymbol{h}$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ guide about the center of a bounding box in an image and $\boldsymbol{w}$ and $\boldsymbol{h}$ represent the width and height of that bounding box over an image. The YOLOv5 model's architecture is displayed in Fig. 4. The class probability provides the classifier result for the object inside the bounding box, and the confidence score provides the confidence level of the model for the bounded object. YOLO is mainly used in cases where speed is an important metric to consider along with accuracy. Another benefit of YOLO could be YOLO aims to develop a more generalized object detection model.

Since its arrival, various improvements have been made to the existing model, and newer models like YOLO v2, v3, v4, and finally, YOLOv5 have emerged. YOLOv5 is the most accurate and efficient model among these, developed by Ultralytics, and the entire repository is published on GitHub github.com/ultralytics/yolov5 [37]. YOLOv5 archives much higher accuracy in a faster time in comparison to other YOLO models. Also, YOLOv5 comes with various default augmentation features that help the model learn better. Augmentation is an oversampling technique that does various processing on an image to generate a slightly modified version of the image, which results in more training data for the model and as a result, the model learns a more generalized representation of the data. Among the augmentation techniques, YOLOv5 uses a method called mosaic augmentation. It combines four images into four tiles to generate training data and helps the model learn to detect significantly smaller objects. There are several variations of YOLOv5 like small, medium, large, etc. And each has different performance results. For example, a small variation will need the lowest time to train but the large model will provide the best accuracy.

3.2.3. EfficientDet

EfficientDet is a popular object detection model introduced by Google [38]. It uses a Bi-directional Feature Pyramid Network, BiFPN, and a compound scaling algorithm to generate accurate detection results. The model uses different versions of the EfficientNet models as the CNN backbone. EfficientNet is a famous CNN model architecture that Google introduced to support scalable CNN architectures [39]. The idea of EfficientNet arose from how researchers can find the proper combination of input data resolution along with the model's density and channel width to achieve the best output. EfficientNet introduced a scaling method that can uniformly scale resolution, model depth, and channel width using a compound coefficient, resulting in a balanced network design for optimal outputs. In EfficientDet, the compound scaling algorithm simultaneously scales the resolution, depth, and width of all the backbones, feature networks, and box/class prediction networks. The architecture is made up of three main parts. The backbone network is the initial part. The EfficientNet family serves as the foundation. A baseline model is produced by employing a neural architectural search (EfficientNet-B0). This base model's width, depth, and resolution can be increased using a scaling factor to match the target device's capabilities. The second essential component is the BiFPN. A weighted bi-directional feature pyramid network is known as BiFPN. It is a novel approach. Exclusive to the EfficientDet architecture, BiFPN was developed. Using multi-scale processing, it presents a set of learnable weights to fuse the information extracted from the input image. Compound scaling is the third major component. The network's dimensions are scaled using a compound coefficient Φ, which helps to properly scale the complete CNN design to the intended processing capability. Compound scaling was discovered to use the extra memory and processing power efficiently. There can be 8 different variations of EfficientDet models (EfficientDet-D0 to EfficientNet-D7) which change according to the compound coefficient Φ. Each model variant uses its corresponding backbone EfficientNet models (EfficientNet-B0 to EfficientNet-B6) except the EfficientDet-D7, which requires EfficientNet-B6. The higher

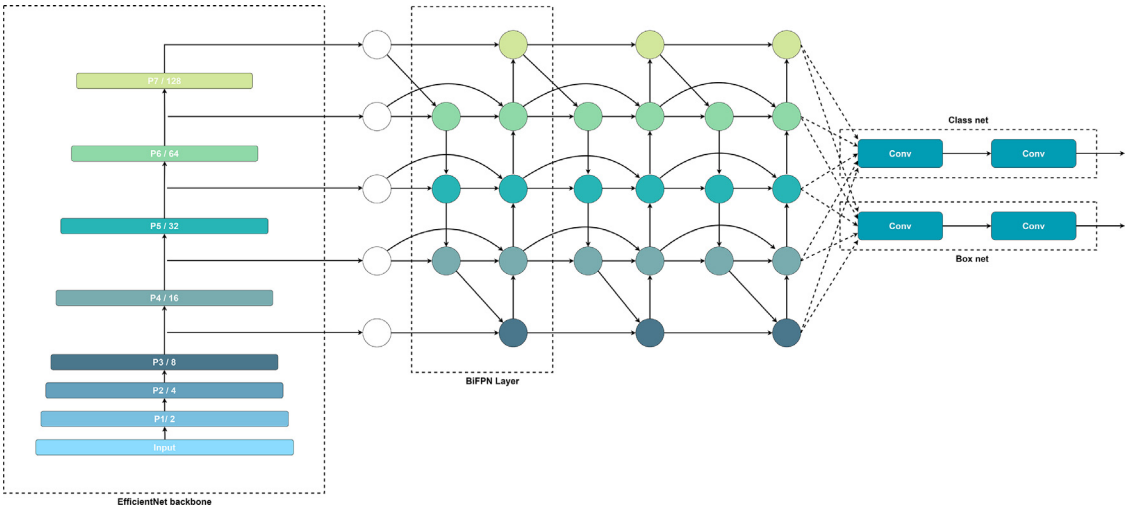

Fig. 5. Architecture of EfficientDet model.

Fig. 6. Before and after applying non-max suppression on a given image.

the compound coefficient value, the more layers, and channels in the BiFPN layer and box/class layers are presented. The architecture of the EfficientDet model is shown in Fig. 5. In our work, we have only used EfficientDet-D0 as its default input size is 512 × 512 sized images, expanding which might create artifacts in our dataset images and cause false results.

3.2.4. Non-max suppression

A model conducts classification and localization simultaneously in object detection tasks. The model can generate numerous bounding boxes of varying dimensions to localize an object in an image. However, we should expect a single bounding box for each object with the highest probability score. In this scenario, the object detection model employs non-max suppression strategies to eliminate all but the best bounding boxes. Fig. 6 demonstrates the application of non-max suppression on an image and its bounding boxes. The bounding box's confidence score and the value of Intersection Over Union (IOU) of the bounding boxes are used to accomplish this. The overlap between bounding boxes is measured using the IOU metric, calculated by comparing the bounding boxes' ground truth label and anticipated coordinates. Generally, a score of 0.5 is considered the IOU threshold value, which helps eliminate unnecessary bounding boxes from an image.

3.2.5. Precision and recall

Precision and Recall are two common evaluation metrics. They are used together to assess models. Precision is the percentage of correctly anticipated positive outcomes out of all positive predictions (Eq. (1)). Whereas recall refers to the portion of positive labels that were correctly identified as such out of all the positive labels (Eq. (2)).

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

3.2.6. F1 score

The F1 score measures a model's ability to identify positive examples while avoiding false positives correctly. It is viewed as a harmonic mean of precision and recall. The F1 score has a range between 0 and 1, where 1 is the best score, and 0 is the worst score. F1 score is determined using Eq. (3).

$$F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

3.2.7. Intersection over union

IOU or Intersection over Union is an evaluation metric used to calculate the precision and recall of an object detection system. It is the proportion of the ground truth labels and the prediction label's areas of union and overlap. The Equation is illustrated in Fig. 7. In particular, the metric is used to determine if a prediction is true positive or false positive. A precision and recall plot is generated for a single classifier at various IOU thresholds following the calculation of precision and recall

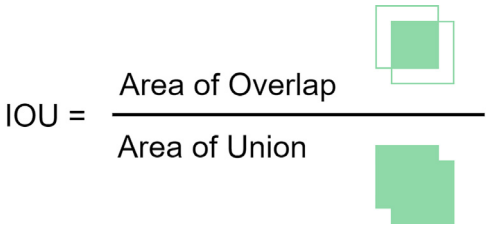

Fig. 7. Intersection over the union.

for different IOU thresholds. The precision–recall curve is then used to calculate the average precision.

3.2.8. Mean average precision

The general metrics used to classify images cannot be applied in an object detection system, as each image may contain multiple objects of different classes. Here, a model's both localization and classification need to be assessed. mAP or Mean Average Precision is determined by taking the mean of the average precision (AP) across all classes. The average precision summarizes the precision–recall curve into a single value representing the average of all precision. AP is determined using Eq. (4). Here, n represents the number of thresholds. The difference between the current and subsequent recalls is determined for each precision–recall and multiplied by the current precision. The mAP is calculated by taking the mean of the AP for all classes. Eq. (5) depicts the equation to calculate mAP. Here, AP_k represents the average precision of class k and n represents the number of classes. Different IoU thresholds are used to evaluate the object detection models. Depending on the threshold, each prediction may differ from the others.

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k+1)] * Precision(k) \tag{4}$$

$$mAP = \frac{1}{n} \sum_{k=n}^{k=1} AP_k \tag{5}$$

3.3. Transfer learning

Transfer learning is a popular and effective method for training large deep learning and machine learning models. Training a model necessitates a large number of computational resources and can be costly to the hardware. This problem, however, can be handled by utilizing transfer learning. In transfer learning, a model is utilized that has been previously trained to do similar tasks using a different or similar dataset. The pre-trained model's layers and learnable parameters are already initialized with proper weights. During transfer learning, certain layers, the last few layers are generally dropped and retrained to teach the model to detect newer classes. In this work, we employed Faster R-CNN, EfficientDet, and YOLOv5 models pre-trained on the MS COCO [40] dataset. Common Objects in Context, COCO is a large annotated dataset that contains 1.5 million object instances in 330k images of 80 distinct object classes such as people, cars, cats, airplanes, household objects, etc. The models are already good at recognizing diverse things from an image because they were pre-trained on the COCO dataset, and using transfer learning, we taught the models to detect certain classes of objects that we have defined in our dataset as visual pollutants.

3.4. Application

To demonstrate the usefulness of visual pollution detection, we have developed a heatmap generation system that combines a mobile application with a trained model. The system is divided into two parts, the mobile application, and the local server.

3.4.1. Android application

Android is the most popular mobile platform in the world. Due to the modern development of smartphones, Android-based smartphones have become the most popular medium of computation. In our work, we have developed a simple application that can be used to collect data from the real world and send it to cloud storage with minimal effort. The application is developed using Flutter, a popular cross-platform development framework. The only function of this application is to let a person capture images through their smartphone and send them to the cloud server along with their GPS coordinates. Upon launching the application for the first time, the application will ask for the user's permission to use the camera, local storage, and GPS sensor. The application uses the device's default camera API to capture the image, hence the user can select any of his preferred lenses or settings to capture a picture. The GPS coordinates are collected from the device's GPS sensor using Fused Location Provider API. The image is uploaded to the cloud server as the user captures and taps the "Upload" button. We have used Firebase as our cloud backend which is very popular for these types of applications. As a user uploads an image, the image is first stored in Firebase's cloud storage and then the image link along with the user's latitude and longitude is stored in Firebase's Real Time database. Fig. 8 displays the data collection method using the Android application.

3.4.2. Local server and heatmap

While the Android application is used for collecting the field data, the local server is used for utilizing the collected data with the help of the trained model to generate the desired application outputs. The local server or the local machine is used for downloading and utilizing the collected data. A Python script-based minimal application is used for fetching the data from the Firebase Real-Time Database to the local machine. As all the data is fetched from the database, a Dataframe is generated which maps each image path with its corresponding latitude and longitude values. Then the collected image paths are sent to the previously trained model which is already good at detecting the visual pollutants from the images. All the images are continuously inferred by the model and the model returns if the given image contains visual pollutants or not. If any image contains visual pollutants then the type of pollutant is also returned by the model. Upon receiving the model's prediction, the information is then again saved in the CSV (comma-separated values) file. So after passing all the images through the model, the newly generated CSV file contains not only the GPS coordinates of each image but also the information on available visual pollutants in that image. This specific functionality is simply achieved by converting the CSV file into a Pandas Dataframe. The converted Dataframe can be treated as a data table and rows or columns can be added depending on the necessity of the task. During the model inference process (where we pass each image into the model) the program adds 7 new columns to the data frame to define each class of pollutant along with the total pollutant count. If an image contains a certain pollutant, the corresponding field in that particular row will get the value of 1, otherwise, all the fields are set as 0. The total pollutant count field is used to store how many different types of pollutants are present in the image and this particular field is important for the heatmap generation process. As the inference process finishes, the data frame is then saved as a CSV file. For heatmap generation, we have used a library called Folium. Folium uses Leaflet js to visualize the Python-processed data on a map. The heatmap generation program loads the CSV file as a data frame and plots the heatmaps on the map. The data frame already contains location coordinates along with the intensity value of available visual pollution in that location. The intensity value is previously detected by the trained visual pollution detection model. A high-level view of the data analytics pipeline is shown in Fig. 9.

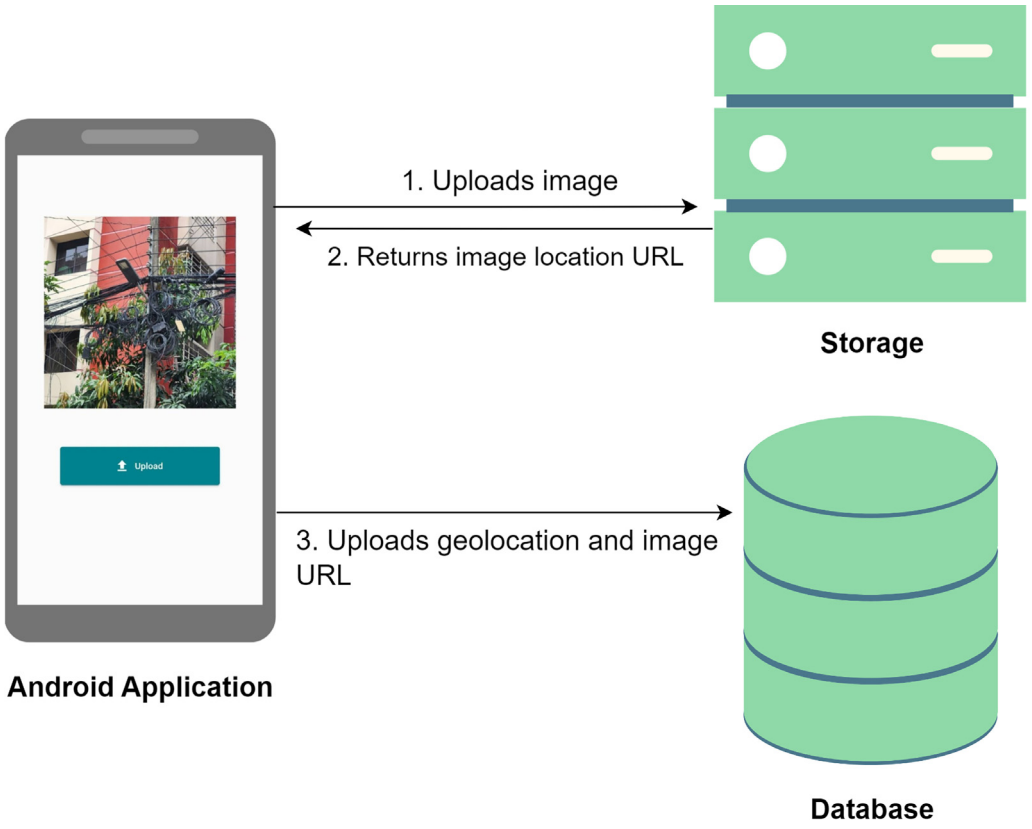

Fig. 8. Flow of data between Android device, cloud storage, and the database.

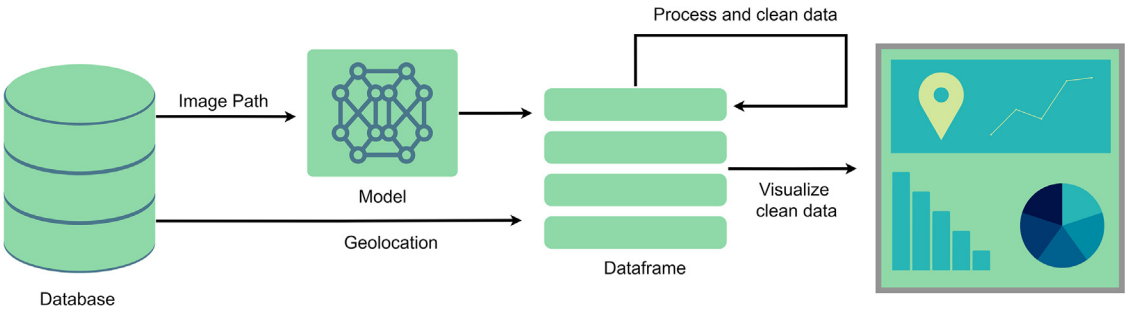

Fig. 9. A high-level view of the data analytics pipeline.

3.4.3. Continuous data generation and retraining

As the users collect and submit data to the cloud database, new data is generated on a frequent basis. Besides running analytics on the collected data, we designed a system to utilize the continuous flow of data to enhance the quality of the entire system. We applied two machine learning concepts called Active Learning and Incremental Learning to design the proposed concept (Fig. 10). The concept of Active Learning arises as data labeling is a tedious process. In Active Learning, the model is trained on a small portion of the labeled dataset, and then new data is labeled with the help of the model's inference [41,42]. In our system, the downloaded data is inferred with the help of the visual pollution detection model at first. If the model is confident about all the objects found on a given image, the image and its annotation are stored in the local database along with the previous dataset. On the other hand, if the model finds low confidence in any detected pollutant, the data is then sent to a human annotator for fixing the annotation. The human annotator gets to fix the annotation and submit the image along with its fixed annotation into the local database. In both cases, the newly arrived data are inferred by the model and depending on the model's confidence, the data is fixed and merged with the previously stored dataset. The minimum confidence required to pass the data is set to 60% in our system, so if any image contains any object where the recognition confidence is less than 60%, the data is re-validated with the help of a human annotator. Otherwise, it is directly sent to the database for retraining the model.

To retrain the model, we applied another concept called Incremental Learning. Incremental Learning is a set of strategies applied for retraining a model in periodic intervals to improve the overall system maintaining the previously learned knowledge [43,44]. As our system gets new data on the flow of its operation, we need to retrain the model to utilize the newly collected and annotated data. In some way, this can now be compared to the previously mentioned transfer learning strategy. Incremental Learning facilitates training models with updated versions of datasets and changes the model's architecture to adapt new classes if needed. But as we are using an Active Learning method in labeling the newly arrived data, we kept the model's architecture the same for this study. As our model is already trained on 6 different classes of visual pollutants, we do not need to retrain the model from the ground. The model is already good at detecting the classes we are about to train it on. For that, we only need to train the model for a

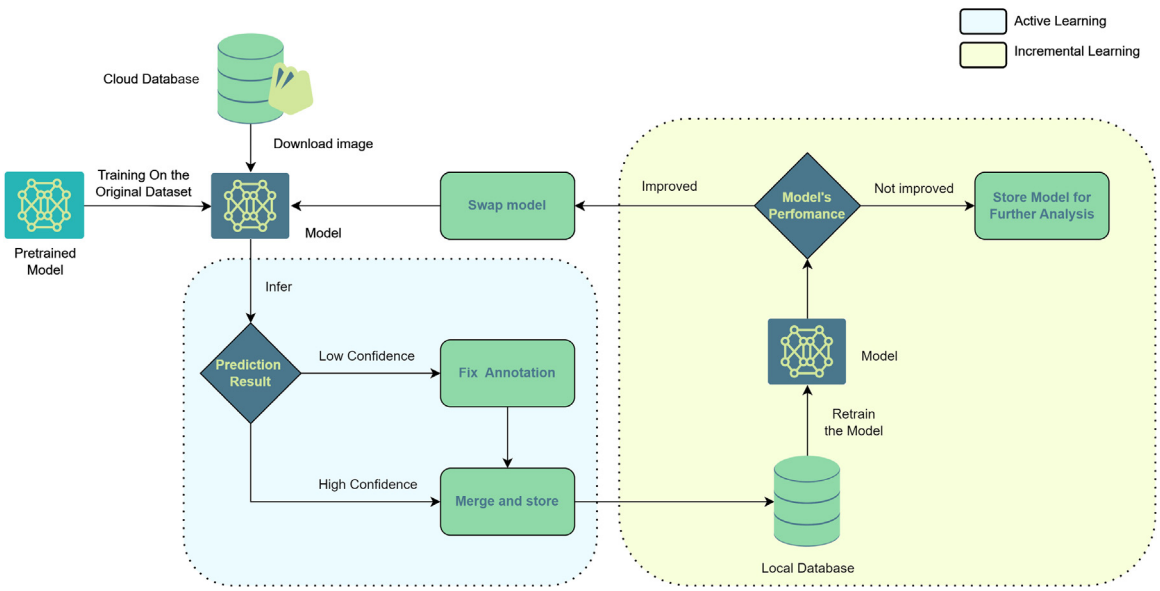

Fig. 10. A representation of the developed system with respect to the implementation of Active Learning and Incremental Learning. Active Learning is used in the semi-autonomous data annotation system where newly added data are inferred by the model and then depending on the confidence score re-checked by an admin. The Incremental Learning part is associated with the re-training of the model depending on the newly merged dataset on certain intervals.

Table 1
Performance of the trained models in mAP, Precision, Recall and F1-score metrics. YOLOv5 takes the advantage of the inbuilt augmentation methods and outperforms the others.

Model	mAP	Precision	Recall	F1-Score
YOLOv5	0.80	0.813	0.748	0.78
YOLOv5 with augmentation	0.85	0.883	0.807	0.84
Faster R-CNN	0.78	0.866	0.704	0.77
EfficientDet	0.77	0.840	0.699	0.76

very short amount of time and treat the operation as transfer learning. In our case, upon the addition of new 100 data, we run the training for 20% of the previously trained epochs. And finally, if the newly trained model performs better than the previously deployed model, we simply replace the model with the new, improved version of the model.

3.5. Setup

The machine we used for our experiments consisted of an Intel Core i7 8700K, 32 GB DDR4 memory, Nvidia RTX 2060 (6 GB), and Kubuntu operating system. On the software side we used both PyTorch and TensorFlow as the deep learning frameworks and various other Python libraries like OpenCV, Numpy, Pandas, TensorBoard, Matplotlib, etc, for various purposes. The android application was developed using Flutter and the cloud backend is implemented using Firebase. On the data side, we trained the models using 256 × 256 sized images and used a small batch size of 8 to optimize the usage of our GPU and CPU resources.

4. Results

As previously mentioned, we used three different object detection models to train our dataset. We split the dataset into 80:20 ratio meaning 80% of the dataset from each class goes to the training set and the rest 20% of the dataset goes to the validation set. We did not explicitly create a test set due to the shortage of dataset and it did not hamper the training process as the models never saw or used any metrics from the validation set during the training. First, we trained our dataset on the YOLOv5 model. The implementation of YOLOv5 comes with various default image augmentation techniques. Initially, on our first training, we turned off all the augmentation techniques, including the mosaic augmentation, and trained the model for 100 epochs. We trained the model on 256 × 256-sized images and kept the batch size of 8. We figured out the epoch number based on how fast the model converged to its optimal mAP score. After running the training for 100 epochs the YOLOv5 model achieved an mAP score of 0.80. Then we moved to the Faster R-CNN model and kept the image and batch size the same as before. As the training set contained 1120 images (80% of 1400) and the batch size was 8, 140 iterations were required by the model to complete 1 epoch. Hence we trained the model for 14,000 iterations which is the same as 100 epochs and the model achieved an mAP score of 0.78. On EfficientDet B0 we trained the model for 14,000 iterations maintaining the batch size of 8 and earned an mAP score of 0.77. Then we retrained the dataset on the YOLOv5 model turning all of its augmentations on. We kept the same training setup as we did on our initial training on YOLOv5. The augmentations applied to this training contained various augmentation techniques like HSV shifting, translation, scaling, flipping, and mosaic augmentation. We kept the values the same as it comes on the default YOLOv5 setup. This time, the model achieved an mAP score of 0.85, the highest among all of these training. Table 1 compares the results of different models on the dataset. It can be seen that YOLOv5 achieved the highest mAP among all the models, followed by Faster R-CNN. Similar dominance of the YOLO models is also seen across all other metrics. Though Faster R-CNN and EfficientDet achieved better precision compared to the default YOLOv5 model, they failed to maintain it in the case of mAP and F1 scores. The YOLOv5 that utilizes the in-built augmentation methods outperforms all the other models by a fair margin. Some demonstrations of our best-performed model can be seen in Fig. 11.

After training the models we deployed our best-performed YOLOv5 model into our application. We chose a popular residential area in Dhaka to run our data collection experiments. Selected volunteers were provided with the Android application to capture and upload images containing visual pollutants. Our volunteers randomly captured and uploaded the photos from various locations inside the designated area. Users could simply launch the application, capture the image and press the upload button to upload the image. Users could also select images from the gallery if they ever needed to. After selecting or capturing the image, users can see the image and then press the upload button to upload the image to the cloud data storage. During the uploading process, an animated loading icon is displayed to assist the users with

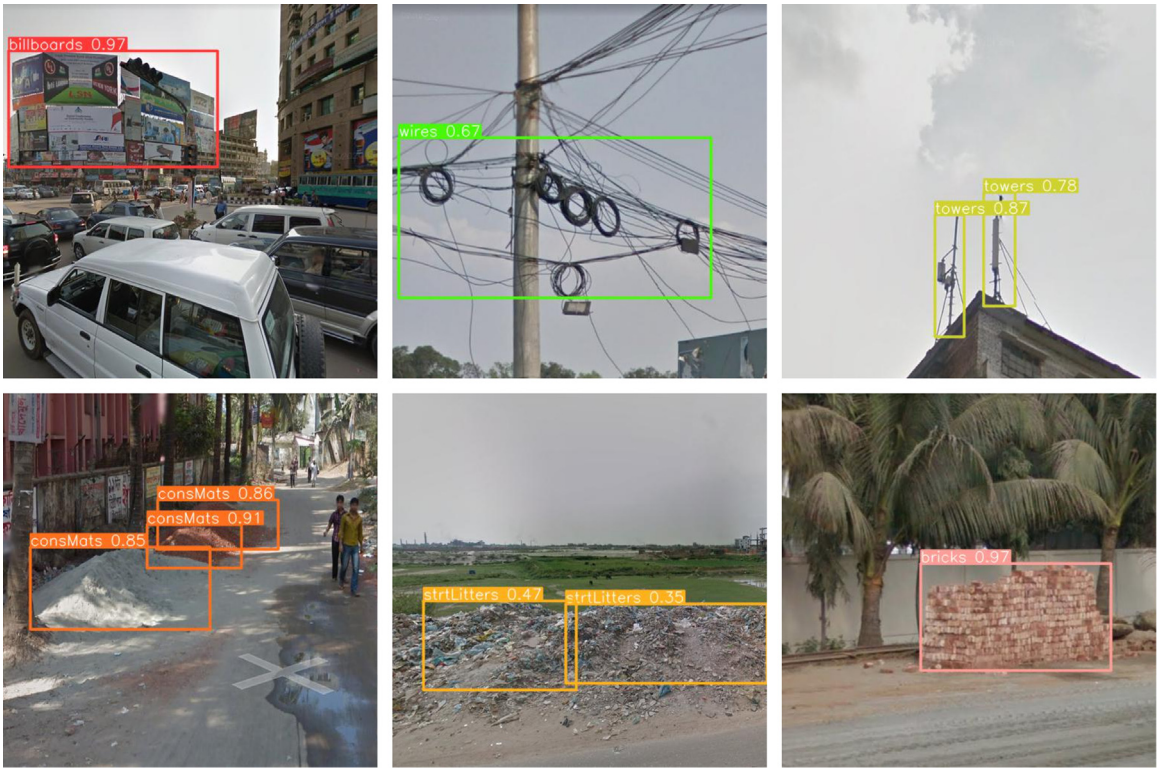

Fig. 11. Visual pollution detection of the best-performed YOLOv5 model on some given unseen images. The model draws bounding boxes around the pollutants and classifies the pollutants along with their confidence score.

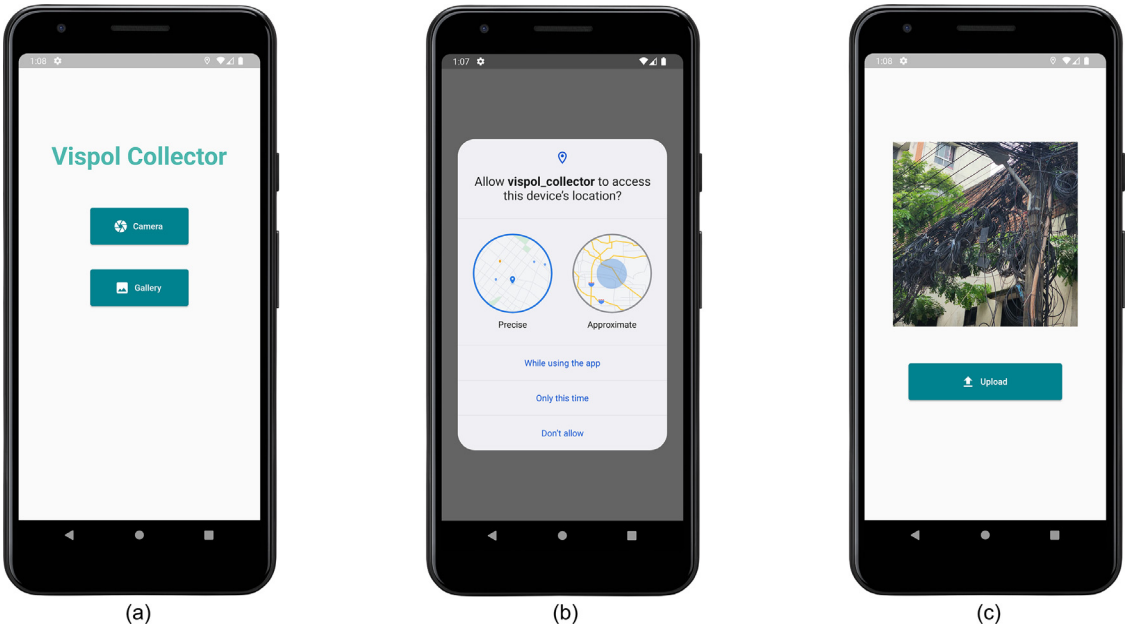

Fig. 12. Screenshots of the android application. (a) Home page (b) The app is asking for the user's permission to access the location (c) Image upload page.

the state of the operation. Fig. 12 illustrates the Android application workflow.

As our volunteers completed submitting the images, we moved on with further analyses of the collected data. We downloaded the collected data from the Firebase on the local machine using Python scripts. Then with the help of our previously designed method, we sent each collected image into the model to acquire the detection results. As the model finished generating results for all the images, we utilized the newly generated information to plot the heat maps of visual pollutants on the map. The generated heat map of visual pollutants is shown in Fig. 13. According to the map, visual pollutants are scattered around the residential area. Particularly, the area on the west, which is a prime business location contains many types of visual pollutants. Although the other parts are entirely residential areas, multiple key areas such as Block-B, Kaji Haj Abdus Sobhan Road, are affected by almost all types of visual pollution. From the pie chart in Fig. 14, we can see that the

Fig. 13. The amounts of visual pollutants in a selected region is plotted on the heatmap. These are based on the pollutants detected by the trained model on the uploaded images.

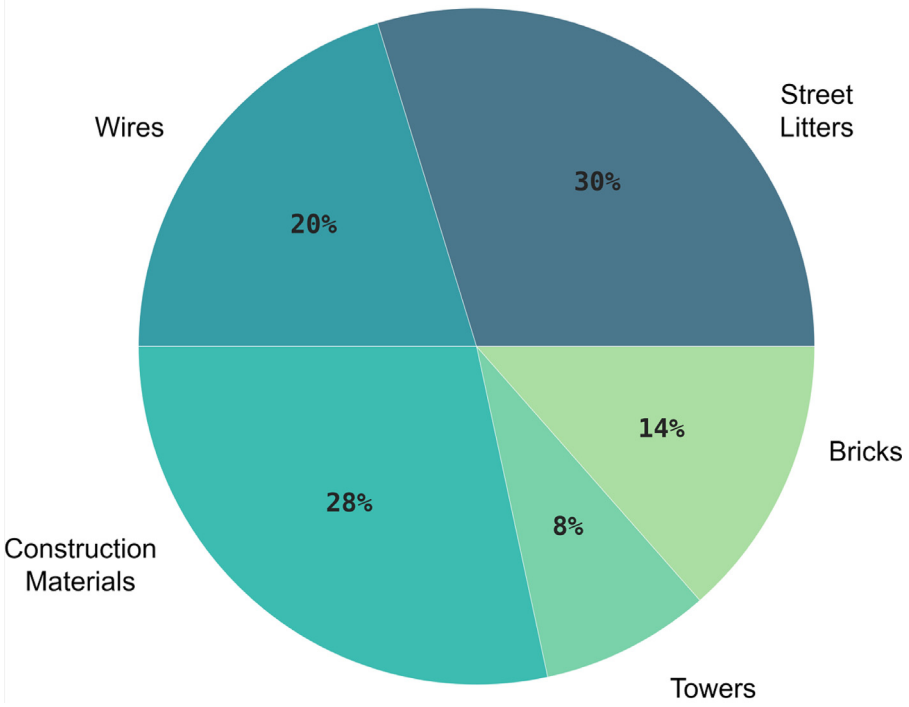

Fig. 14. A pie chart visualization shows the overall distribution of the detected pollutants from the collected images on the specific region.

Bashundhara residential is mainly influenced by Street Litter, followed by Construction materials; in contrast, least impacted by billboards as there are not any.

As for the Active Learning process, the model merged the images along with their annotation files with the previous dataset. When the model found less confidence score on the pollutants detection on certain images, the images were then stored in a buffer space for the human annotator to re-validate. A web-based image annotation tool was integrated with the system that could visualize the bounding boxes predicted by the model and let human annotators change things according to their choice. As the annotator validates or fixes the annotations, the images are passed from the buffer space and merged with the previous dataset. Upon the arrival of a certain amount of new data, the model retrained itself with the new dataset with only 20 epochs

which is less than its initial training. As we collected and utilized a very low amount of new data, the newly trained model did not perform any better than the existing one, hence the current model is kept in the system as it is and the newly trained one is stored in another location for future analyses.

5. Discussion

We utilized transfer learning on three popular object detection models so that they can work as visual pollutants detectors. We utilized a dataset that contained images along with annotations of the presenting visual pollutants in them. We also randomly split the dataset into 80:20 ratio and the same training and validation sets were used for training and validating all the models. From our experiments, we discovered that the YOLOv5 model outperformed the other object detection models in the case of visual pollutants detection on the dataset. The performance of the model also improves when the inbuilt augmentations are applied. As it is a single-stage model, YOLOv5 is also faster during the inference process and the deployment process is simpler compared to the others. As our system requires model swapping at certain intervals, the overall architecture and the usability of YOLOv5 make the overall development and deployment process much more feasible. The Android application is easy to use and anyone could capture and send pictures with minimal effort. During the image submission, the user's GPS coordinates are also captured upon receiving their permission. This helped us to plot the intensity of visual pollution in a geospatial manner. Along with displaying a heat map of visual pollutants in a given location, the system shows other information like a pie chart of the found pollutants. Using such a system, associated authorities can monitor the condition of visual pollution in a given region. This will help them to address the issues that we often miss out as they are a kind of a hidden pollution that we do not necessarily understand. Various organizations that work on the study and development of urban and regional planning can utilize our proposed system to understand and mitigate the problem of visual pollution efficiently. In the future, a real-time video-based analysis system can be implemented on the existing architecture of our proposed system. In fact, the trained YOLOv5 model can already detect visual pollutants from video feeds. A possible utilization of such a system can be recording video feeds from vehicles and running automatic analyses on them instead of engaging manual laborers. The proposed system also addresses the issue of dataset generation for problems that require more and more data to solve efficiently. As we previously mentioned, the subject of visual pollution that utilizes machine learning lacks work and one of the prime reasons for it is the dataset itself. We have implemented and demonstrated a system that can continuously handle incoming data, and with a semi-supervised approach, the data can be merged to generate larger datasets. Also, with the help of Incremental Learning, the model gets better day by day by utilizing the newly collected datasets. As the model gets better day by day, the model will be generating more and more accurate automatic annotations of the incoming data, lowering the labor of the human annotators. Incremental Learning also helps the model to adapt to newly added classes in the dataset. If a new class is needed to be added to the dataset, following various Incremental Learning techniques the model can be retrained to learn the new class without forgetting previously learned knowledge.

6. Conclusion

Since the industrial revolution, mankind always focused mainly on the development of a better economic society. To achieve this, they had to go through a lot, and engage in a lot of activities like scientific discoveries and implementing challenging engineering concepts. We invented better ways of transportation, better ways of communication, and overall better ways of living. But on the way, we disrupted nature's order. We contaminated the water, air, and almost everything around us. We are the primary cause of environmental pollution. Even though it took a long time to understand the importance of maintaining the proper order of nature and environment, numerous studies have been performed in this field by now. Even complex sophisticated computational systems are regularly studied and developed to mitigate and manage various environmental pollutants. Even though visual pollution is a very important subject of environmental pollution management, not much work has been done on this topic that implements modern computational algorithms like machine learning, deep learning, etc. With the fast-growing development of machine learning-based research and technologies, it is obvious that every sector would like to get facilitated by such automated systems in near future. There have been many works performed on various domains of environmental science and its management that utilizes machine learning, but the visual pollution domain lacks such efforts. We have tried to introduce how machine learning concepts like object detection can be used to automatically detect and recognize visual pollutants from our environment and how useful applications can be built regarding this. To this extent, we experimented with three popular object detection models and applied various useful machine-learning concepts to analyze and build a complete system that can help authorities control and manage visual pollution efficiently. We used an object detection dataset that was collected and annotated using Google Street View that addresses six different types of visual pollutants seen on streets around Dhaka city. Among all the models, the YOLOv5 model performed the best in detecting visual pollutants on the dataset beating Faster R-CNN and EfficientDet. After that, we developed a system that can be used to collect and analyze visual pollution with the help of our trained model. We have built an android application using which volunteers or authorities can simply capture and submit pictures that contain visual pollutants along with their GPS location to the cloud storage and these collected data can be downloaded and analyzed on the local machine anytime. The trained model can infer the collected data and provide analytics based on the detection results. These can be then plotted on a map as a heat map to visualize a location's condition regarding visual pollution. Also, other important visual representations can be generated that can aid the admins to understand the condition of a particular region or location. Besides this, the system uses a concept called Active Learning to re-label the newly collected dataset. With the help of an Active Learning strategy, we have developed a semi-autonomous data annotation system that helps the admins to annotate new images that contain visual pollutants with minimal effort. Also to retrain the model on regular basis with a new dataset we have applied another concept called Incremental Learning that helps the system to utilize both the newly added data and previously trained knowledge of the model to build a much better one. Overall, we have gone through how machine learning and its applications can aid us in easily detecting and managing the visual pollutants present in our environment. It is an issue that causes problems in this modern world every day but is not necessarily well understood by all. We hope that our work inspires many other researchers and scientists to utilize sophisticated computational studies like machine learning to shape a better future for all.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

References

[1] G. Ceballos, P.R. Ehrlich, A.D. Barnosky, A. García, R.M. Pringle, T.M. Palmer, Accelerated modern human–induced species losses: Entering the sixth mass extinction, Sci. Adv. 1 (5) (2015) e1400253.

[2] G. Ceballos, P.R. Ehrlich, R. Dirzo, Biological annihilation via the ongoing sixth mass extinction signaled by vertebrate population losses and declines, Proc. Natl. Acad. Sci. 114 (30) (2017) E6089–E6096.

[3] W. Ripple, C. Wolf, T. Newsome, P. Barnard, W. Moomaw, P. Grandcolas, World scientists' warning of a climate emergency, BioScience (2019).

[4] R. Nawaz, K. Wakil, Visual Pollution: Concepts, Practices and Management Framework, Emerald Group Publishing, 2022.

[5] J.C. Nagle, Cell phone towers as visual pollution, Notre Dame JL Ethics Pub. Pol'Y 23 (2009) 537.

[6] A. Portella, Visual Pollution, Routledge, 2016, http://dx.doi.org/10.4324/9781315547954.

[7] K. Wakil, M.A. Naeem, G.A. Anjum, A. Waheed, M.J. Thaheem, M.Q.u. Hussnain, R. Nawaz, A hybrid tool for visual pollution assessment in urban environments, Sustainability 11 (8) (2019) 2211.

[8] D. Yilmaz, A. Sagsöz, In the context of visual pollution: effects to trabzon city center silhoutte, Asian Soc. Sci. 7 (5) (2011) 98.

[9] T. Wibble, U. Södergård, F. Träisk, T. Pansell, Intensified visual clutter induces increased sympathetic signalling, poorer postural control, and faster torsional eye movements during visual rotation, PLoS One 15 (1) (2020) e0227370.

[10] L. Tyrväinen, A. Ojala, K. Korpela, T. Lanki, Y. Tsunetsugu, T. Kagawa, The influence of urban green environments on stress relief measures: A field experiment, J. Environ. Psychol. 38 (2014) 1–9.

[11] W. Jeremy, Air pollution and brain health: an emerging issue, Lancet 390 (2017) 1345–1422.

[12] M.C. Turner, Z.J. Andersen, A. Baccarelli, W.R. Diver, S.M. Gapstur, C.A. Pope, D. Prada, J. Samet, G. Thurston, A. Cohen, Outdoor air pollution and cancer: An overview of the current evidence and public health recommendations, CA: Cancer J. Clin. 70 (2020) 460–479.

[13] D. Hou, Y.S. Ok, Soil pollution — speed up global mapping, Nature 566 (2019) 455–456.

[14] M.K. Hasan, A. Shahriar, K.U. Jim, Water pollution in Bangladesh and its impact on public health, Heliyon 5 (8) (2019) e02145.

[15] P.K. Gupta, A. Saxena, B. Dattaprakash, R.S. Sheriff, S.H. Chaudhari, V. Ullanat, V. Chayapathy, Applications of artificial intelligence in environmental science, Artif. Intell. (AI) (2021) 225–240.

[16] Z. Tian, Z. Yu, Y. Li, Q. Ke, J. Liu, H. Luo, Y. Tang, Prediction of river pollution under the rainfall-runoff impact by artificial neural network: A case study of shiyan river, shenzhen, China, Front. Environ. Sci. 10 (2022) 810.

[17] S. Bindal, C.K. Singh, Predicting groundwater arsenic contamination: Regions at risk in highest populated state of India, Water Res. 159 (2019) 65–76.

[18] Y.-S. Chang, H.-T. Chiao, S. Abimannan, Y.-P. Huang, Y.-T. Tsai, K.-M. Lin, An LSTM-based aggregated model for air pollution forecasting, Atmosp. Pollut. Res. 11 (2020) 1451–1463.

[19] N. Ahmed, M.N. Islam, A.S. Tuba, M. Mahdy, M. Sujauddin, Solving visual pollution with deep learning: A new nexus in environmental management, J. Environ. Manag. 248 (2019) 109253.

[20] M.Y. Hossain, I.R. Nijhum, A.A. Sadi, M.T.M. Shad, R.M. Rahman, Visual pollution detection using google street view and YOLO, in: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON, IEEE, 2021, pp. 0433–0440.

[21] T.-D. Hoang, N.M. Ky, N.T.N. Thuong, H.Q. Nhan, N.V.C. Ngan, Artificial intelligence in pollution control and management: Status and future prospects, Artif. Intell. Environ. Sustain.: Chall. Solut. Era Ind. 4.0 (2022) 23–43.

[22] Z. Ye, J. Yang, N. Zhong, X. Tu, J. Jia, J. Wang, Tackling environmental challenges in pollution controls using artificial intelligence: A review, Sci. Total Environ. 699 (2020) 134279.

[23] S. Kumar, D. Yadav, H. Gupta, O.P. Verma, I.A. Ansari, C.W. Ahn, A novel YOLOv3 algorithm-based deep learning approach for waste segregation: Towards smart waste management, Electronics 10 (2020) 14.

[24] H. Panwar, P. Gupta, M.K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, S. Sharma, I.H. Sarker, AquaVision: Automating the detection of waste in water bodies using deep transfer learning, Case Stud. Chem. Environ. Eng. 2 (2020) 100026.

[25] A. Nazerdeylami, B. Majidi, A. Movaghar, Smart Coastline environment management using deep detection of manmade pollution and hazards, in: 2019 5th Conference on Knowledge Based Engineering and Innovation, KBEI, IEEE, 2019, pp. 332–337, http://dx.doi.org/10.1109/KBEI.2019.8735012.

[26] R. Janarthanan, P. Partheeban, K. Somasundaram, P.N. Elamparithi, A deep learning approach for prediction of air quality index in a metropolitan city, Sustainable Cities Soc. 67 (2021) 102720.

[27] S. Kundu, U. Maulik, Vehicle pollution detection from images using deep learning, Intell. Enabled Res.: DoSIER 2019 (2020) 1–5.

[28] S.A. Bakar, A. al Sharaa, S. Maulan, R. Munther, Measuring visual pollution threshold along kuala lumpur historic shopping district streets using cumulative area analysis, 2019.

[29] M. Cvetković, A. Momčilović-Petronijević, Visual pollution of the historical city core–a case study, the city of Niš, in: Proceedings of the 6th International Conference Contemporary Achievements in Civil Engineering, Subotica, Serbia, Vol. 20, 2018, pp. 495–504.

[30] N.H. Tasnim, S. Afrin, B. Biswas, A.A. Anye, R. Khan, Automatic classification of textile visual pollutants using deep learning networks, Alex. Eng. J. 62 (2023) 391–402.

[31] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2014, pp. 580–587, http://dx.doi.org/10.1109/CVPR.2014.81.

[32] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, IEEE, 2015, pp. 1440–1448, http://dx.doi.org/10.1109/ICCV.2015.169.

[33] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.

[36] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[37] ultralytics, Github.com/ultralytics/yolov5, 2022, URL: https://github.com/ultralytics/yolov5, Accessed December 15, 2022.

[38] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.

[39] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 6105–6114, URL https://proceedings.mlr.press/v97/tan19a.html.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755, http://dx.doi.org/10.1007/978-3-319-10602-1_48.

[41] B. Settles, Active Learning Literature Survey, Technical Report 1648, University of Wisconsin–Madison, 2009.

[42] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B.B. Gupta, X. Chen, X. Wang, A survey of deep active learning, ACM Comput. Surv. 54 (9) (2021) 1–40.

[43] R. Polikar, L. Upda, S.S. Upda, V. Honavar, Learn++: An incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern. C (Appl. Rev.) 31 (4) (2001) 497–508.

[44] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 374–382.