Sci

MDPI

Article

Deep-Learning-Based Real-Time Visual Pollution Detection in Urban and Textile Environments

Md Fahim Shahoriar Titu , Abdul Aziz Chowdhury , S. M. Rezwanul Haque and Riasat Khan *

Electrical and Computer Engineering, North South University, Dhaka 1229, Bangladesh; fahim.shahoriar@northsouth.edu (M.F.S.T.); ac.abdulaziz98@gmail.com (A.A.C.); rezwanul.haque18@northsouth.edu (S.M.R.H.)
* Correspondence: riasat.khan@northsouth.edu; Tel.: +880-1879992680

Abstract: The environmental physiognomy of an area can significantly diminish its aesthetic appeal, rendering it susceptible to visual pollution, the unbeaten scourge of modern urbanization. In this study, we propose using a deep learning network and a robotic vision system integrated with Google Street View to identify streets and textile-based visual pollution in Dhaka, the megacity of Bangladesh. The issue of visual pollution extends to the global apparel and textile industry, as well as to various common urban elements such as billboards, bricks, construction materials, street litter, communication towers, and entangled electric wires. Our data collection encompasses a wide array of visual pollution elements, including images of towers, cables, construction materials, street litter, cloth dumps, dyeing materials, and bricks. We employ two open-source tools to prepare and label our dataset: LabelImg and Roboflow. We develop multiple neural network models to swiftly and accurately identify and classify visual pollutants in this work, including Faster SegFormer, YOLOv5, YOLOv7, and EfficientDet. The tuna swarm optimization technique has been used to select the applied models' final layers and corresponding hyperparameters. In terms of hardware, our proposed system comprises a Xiaomi-CMSXJ22A web camera, a 3.5-inch touchscreen display, and a Raspberry Pi 4B microcontroller. Subsequently, we program the microcontroller with the YOLOv5 model. Rigorous testing and trials are conducted on these deep learning models to evaluate their performance against various metrics, including accuracy, recall, regularization and classification losses, mAP, precision, and more. The proposed system for detecting and categorizing visual pollution within the textile industry and urban environments has achieved notable results. Notably, the YOLOv5 and YOLOv7 models achieved 98% and 92% detection accuracies, respectively. Finally, the YOLOv5 technique has been deployed into the Raspberry Pi edge device for instantaneous visual pollution detection. The proposed visual pollutants detection device can be easily mounted on various platforms (like vehicles or drones) and deployed in different urban environments for on-site, real-time monitoring. This mobility is crucial for comprehensive street-level data collection, potentially engaging local communities, schools, and universities in understanding and participating in environmental monitoring efforts. The comprehensive dataset on visual pollution will be published in the journal following the acceptance of our manuscript.

Keywords: artificial intelligence; deep learning; EfficientDet; Raspberry Pi; SegFormer; visual pollution

Citation: Titu, M.F.S.; Chowdhury, A.A.; Haque, S.M.R.; Khan, R. Deep-Learning-Based Real-Time Visual Pollution Detection in Urban and Textile Environments. *Sci* **2024**, *6*, 5. https://doi.org/10.3390/sci6010005

Academic Editor: Yudong Zhang

Received: 29 November 2023
Revised: 4 January 2024
Accepted: 5 January 2024
Published: 11 January 2024

1. Introduction

Our environment is undergoing significant transformations due to the amalgamation of detrimental elements and the advancement of human civilization [1]. With more people congregating in cities, towns, and rural areas than ever before, various human-created pollutants are surreptitiously filling the environment due to these activities [2]. Beyond their environmental impact, these contaminants can also have adverse effects on our physical and mental well-being. At times, we encounter these contaminants inadvertently, causing disruptions to our visual and aesthetic experiences [3]. These unfamiliar and unpleasing

visual elements are collectively known as visual pollutants, contributing to the occurrence of visual pollution. Visual pollution results from several environmental factors [4] and can manifest in various forms, including extensive billboards, wind and nuclear power facilities, electrical lines, industrial structures, construction sites, street litter, and roadside billboards displaying advertisements [5]. Despite its glaring presence, visual pollution often receives minimal attention in the face of more conventional forms of environmental contamination [6]. Visual pollution poses a threat to the aesthetics of landscape physiognomy, diminishing the overall environmental appeal. Excessive deployment of numerous unsightly elements in urban environments can result in visual blight and eyesores for city residents [7].

It is estimated that approximately 20% of global water pollution can be attributed to the dyeing and finishing processes involved in textile production [8]. The residual materials generated during the textile manufacturing process are referred to as textile waste [9]. This waste can be generated at various stages of textile production, including spinning, weaving, dyeing, finishing, and even after the final product is formed. Textile waste can be produced unintentionally or intentionally as part of efforts to enhance efficiency [10]. The categorization of textile contaminants has become a subject of increasing interest in recent times. In 2018, World Bank research reported that 242 million tons of plastic waste, constituting 12% of all solid waste, were generated globally in 2016 [11]. By 2050, the world is projected to produce 3.40 billion tons of plastic waste annually, accounting for 12% of total solid waste. This scenario represents a significant increase from the current 2.01 billion tons of waste produced annually. Despite the relatively high prevalence of disease, hazardous working conditions can result in infections affecting the skin, respiratory system, and digestive system [12]. Visual pollution encompasses the adverse effects of pollutants that impair vision and mental health, thereby reducing overall quality of life [13]. This pollution originates from various environmental factors and can manifest in different ways. Examples include large billboards, wind and nuclear power plants, electrical cables, industrial structures, construction sites, street litter, and roadside billboards displaying advertisements [14].

Every day, our efforts are directed toward modernizing the world; however, the consequences of this modernization have a detrimental impact on the environment. Rapidly expanding cities around the globe are becoming cluttered with undesirable visual elements. The pollution resulting from these visible objects, often referred to as visual pollutants, in our surroundings is termed visual pollution. Visual pollutants encompass any objects that are aesthetically displeasing and intrusive to the observer. These can include billboards, tangles of electric wires, street litter, construction materials, graffiti, cellphone towers, and worn-out buildings, as well as instances such as industrial clothes dumps, industrial textile billboards, and industrial textile dye.

Visual pollution has received relatively limited attention, overshadowed by the prevalence of traditional environmental pollution in today's world. However, it poses a significant threat to environmental aesthetics, diminishing the overall landscape quality. To address the categorization of visual pollution in the streets and textile industries of Dhaka, we propose a novel system combining Google Street View data with a deep learning network and a robotic vision system. This study aims to raise awareness of the detrimental effects of visual pollution in both urban and industrial settings. This work's significant contributions are as follows:

- Combining and developing a comprehensive dataset focused on textile-based and conventional urban visual wastes of nine categories is a significant contribution to this research. The process of labeling was executed with precision using the LabelImg and Roboflow API, which involved manual labeling, sorting, renaming, and meticulous categorization of each image. We employed data augmentation techniques to enhance the diversity and volume of the employed dataset.
- Subsequently, we implemented a range of neural network models, Faster SegFormer, YOLOv5, YOLOv7, and EfficientDet, to automatically detect and classify visual con-

taminants. The tuna swarm optimization (TSO) technique was employed to select the applied models' final layers and corresponding hyperparameters. Various experiments were conducted, scrutinizing precision, recall, mean average precision (mAP), intersection over union (IoU), and loss metrics encompassing classification, localization, and regularization losses, as well as overall accuracy, to ensure the reliability and effectiveness of the applied models.

- Finally, the proposed automatic visual pollution detection system was assembled into a robust hardware setup, comprising a Raspberry Pi 4B microcontroller, a 3.5-inch touchscreen display, and a Xiaomi-CMSXJ22A web camera. The selected model, YOLOv5, was seamlessly integrated into the microcontroller.
- To the best of our knowledge, this is the first time a deep learning technique has been integrated with a Raspberry Pi 4-based edge device to instantly detect nine distinct categories of visual pollutants in real time.

This article is structured into four sections. Section 2 reviews related works on automatic visual pollution detection and identifies existing gaps in the literature. Section 3 depicts the proposed system, where we describe a comprehensive overview of the applied deep learning models, various system components, and the comprehensive dataset that we utilized. The overall workflow of our proposed system is described in this section. Section 4 delves into the demonstrated results and the challenges that we encountered during the study. Finally, in Section 5, we present our conclusions and outline potential avenues for future enhancements to this system.

2. Related Works

Visual pollution, characterized by disorderly and displeasing urban environments, is inherently subjective and challenging to quantify precisely. In recent years, substantial research efforts have been initiated to identify and categorize various forms of visual pollution by applying artificial intelligence and computer vision techniques. The automated recognition of visual disturbances using advanced deep learning methods can aid governmental bodies and relevant authorities in taking proactive measures. This section provides a brief overview of recent endeavors in automated visual pollution detection.

Ahmed et al. [15] demonstrated the automated detection of a wide range of visual pollution using deep learning convolutional neural networks. The authors classified their data into four categories and employed a convolutional neural network with multiple layers of artificial neurons. The implemented customized CNN model attained a training accuracy of 95% and a validation accuracy of 85%. Andjarsari and her team [16] reported that the presence of billboards and street graffiti along the route could impact the area's aesthetics and potentially obstruct vision. To investigate visual pollution, the authors employed an AHP-based SBE technique. Furthermore, a combination of SWOT, AHP, and QSPM techniques was used.

Hossain et al. [17] introduced artificial intelligence techniques for identifying visual contaminants using images from Google Street View. The authors selected the different roads of Dhaka, the capital city of Bangladesh, as their test subject due to its recent ranking as one of the world's most polluted cities. The image dataset was manually curated, with photos collected from various perspectives, focusing on frames containing visual pollution. These images were meticulously tagged using the CVAT framework and utilized for model training. Notably, they leveraged the object detection model YOLOv5 for detection and classification purposes. Yang et al. [18] developed WasNet, a distinctive and lightweight neural network approach for trash classification. This network stands out due to its efficiency, with only 1.5 million parameters on the ImageNet dataset, which is half the parameters of mainstream neural networks. Despite its lightweight nature, it exhibits reasonable performance. It operates at 3 million floating-point operations per second (FLOPs), making it one-third as resource-intensive as other well-known lightweight neural networks. Notably, it achieves a 64.5% accuracy on the ImageNet dataset, an 82.5% accuracy on the Garbage Classification dataset, and an impressive 96.10% accuracy on the TrashNet dataset.

Mittal et al. [19] introduced a novel smartphone app, SpotGarbage, leveraging a deep architecture based on fully convolutional networks for garbage detection. After training on the newly released Garbage In Images (GINI) dataset, the model attained a mean accuracy of 87.69%. Furthermore, they optimize the network architecture, resulting in a significant reduction of 96.8% in prediction time and an 87.9% decrease in memory consumption, all without compromising accuracy. Marin and his team [20] explored three distinct feature extraction schemes for underwater marine debris using six well-established deep convolutional neural networks (CNNs) and other features. They conducted a comprehensive analysis comparing the performance of a neural network (NN) classifier constructed on top of deep CNN feature extractors in various configurations: when the feature extractor is fixed, fine-tuned on the task, fixed during the initial training phase, and fine-tuned afterward. Their findings reveal that the improved Inception-ResNetV2 feature extractor outperforms others, achieving an impressive accuracy of 91.40% and an F1 score of 92.08%.

Tasnim et al. [21] delved into the application of computer vision techniques to develop an innovative approach for the automatic detection and categorization of visual pollutants associated with the textile industry. Their research focused on three categories of textile-based visual pollutants: cloth litter, advertising billboards, signs, and textile dyeing waste materials. Deep learning algorithms, including Faster R-CNN, YOLOv5, and EfficientDet, were employed to classify the collected dataset automatically. Bakar et al. [22] utilized a standard cumulative area technique to address the issue of assessing visual pollution. Using a photo booklet, the authors conducted surveys with respondents in an architectural and urban zone in Kuala Lumpur, Malaysia. The findings of the study, which took demographic factors into account, revealed insights regarding visual pollutants based on the respondents' varying tolerance levels.

Setiawan et al. [23] utilized the SIFT technique to distinguish between photos of organic and non-organic waste. The input image dimensions were adjusted to meet specific requirements. Finally, the SIFT algorithm achieved an impressive accuracy level of 89%. Ahmed et al. [24] introduced computer vision systems and intelligent cameras or optical sensors for object detection and tracking. They employed SSD (Single Shot MultiBox Detector) and YOLO (You Only Look Once) models for object detection and localization, complemented by visual processing intelligent robotic cameras. Their system demonstrated a remarkable performance, with a maximum valid detection rate ranging from 90% to 93%.

The literature reviews highlight the extensive research conducted on automated visual pollution identification using deep learning techniques, demonstrated in Table 1. However, notable gaps persist in the integration of detecting and classifying visual pollution in both the textile industry and urban streets by leveraging advanced deep learning techniques. Furthermore, most of the articles have not explored the use of modern edge devices, such as the Raspberry Pi 4B, for automated identification processes in real time.

Table 1. A comparative analysis of related works on AI-based visual pollution detection.

Ref	Sample Size	Number of Classes	Limitations
[15]	1200	4	Low accuracy, limited classes of objects
[17]	1400	6	Low accuracy
[20]	2395	6	Detects underwater pollutants only
[21]	2663	3	Detects textile-based pollutants only, limited classes of objects
[25]	34,460	3	Detects road-based pollutants only, limited classes of objects

A comprehensive analysis of the developed devices is lacking in most existing works. This study, in contrast, delves into a comparative analysis of automatic detection methods

using the Raspberry Pi 4B microcontroller and advanced computer-vision-based library functions. The Raspberry Pi 4B microcontroller utilizes the YOLOv5 tiny deep learning approach for object identification, leading to efficient detection outcomes and quicker response times than previous studies. This research extensively investigates the proposed automatic device, providing insights into its accuracy, implementation cost, and other relevant aspects related to detecting and classifying visual pollution in the textile industry and urban streets.

3. Proposed System

3.1. Dataset and Its Preprocessing

Dataset Collection: This work makes a substantial contribution by utilizing an extensive visual pollution database based on Google Street View and the textile industry of Dhaka, Bangladesh. This section briefly describes the data acquisition and its corresponding preprocessing methods.

- Our dataset was compiled by gathering a selection of images from Google and Microsoft Bing search tools, along with images from two prominent knit garment industries in Bangladesh. Through web crawling, we accumulated more than 1400 photographs, consisting of 200 depicting textile billboards, 200 bricks, 300 construction equipment, 300 street trash, 200 towers, and 200 overhead cables. We removed several irrelevant photographs collected during the data crawling process to refine our dataset.
- Additionally, we collected 600 photographs of textile advertising billboards from Original Marines Ltd., a major apparel manufacturer in Gazipur, Bangladesh. Another 350 photos of building supplies were obtained from Beximco Industrial Park in Gazipur, a significant clothing supplier in Bangladesh. Finally, we gathered 200 pictures of bricks, towers, and wires from various locations throughout Dhaka, Bangladesh, including roadside scenes. The combined dataset, comprising both open-source and locally acquired images related to textile-based visual pollution, encompasses photos of various categories, as presented in Table 2. Figure 1 illustrates photographs representing different classes, including clothing trash, textile billboards, and textile dye, which were collected from Google and Microsoft Bing search engines.

Dataset Classes: Visual pollutants encompass any objects that have the potential to affect an individual's visual perception adversely. These pollutants can be attributed to various factors, including excessive billboards and signage, hanging cables, roadside litter, communication infrastructure such as grid and cellphone towers, and several other sources. In this article, we prioritize the analysis of common visual pollutants frequently observed on the streets of Dhaka, Bangladesh, with particular attention paid to the presence of urban elements and textile factories. Specifically, our study focuses on the visual pollutants listed in Table 2, which includes billboards, street litter, construction materials, bricks, wires, towers, industrial clothes dump, industrial textile billboards, and industrial textile dye.

Data Preprocessing: After the initial data collection from web scrapping and nearby industries and roadsides, the raw dataset exhibited limitations due to the inherent constraints of automated search engine operations, which resulted in the inclusion of outliers and undesirable photographs. To ensure the reliability of our dataset, we conducted a manual review and removed these undesirable images through careful arbitration. We leveraged two open-source applications for data preparation and labeling, namely LabelImg and Roboflow. LabelImg, a widely used graphical image annotation tool [26], played a crucial role in this work. Using LabelImg, we meticulously renamed the images and added text-based labels, as exemplified in Figure 2. Furthermore, we incorporated COCO JSON and segmentation-masks-based labels to accommodate the requirements of the EfficientDet and SegFormer models.

Figure 1. Sample images of the employed visual pollution dataset.

Table 2. Number of images for each category of the dataset.

Class	Google Street View	Collected	Augmented	Total
Billboards "0"	205	-	318	523
Bricks "1"	212	-	350	562
Construction Materials "2"	342	-	350	692
Street Litters "3"	310	-	350	660
Towers "4"	214	-	350	564
Wires "5"	212	-	350	562
Industrial Clothes Dump	200	281	350	831
Industrial Textile Billboard	100	117	350	567
Industrial Textile Dye	145	200	350	692
Total	1940	598	3018	5653

Subsequently, we utilized a model to expand our dataset significantly, complementing it with synthetic augmentation techniques. Deep learning models tend to perform more effectively as the dataset size increases, and data augmentation serves as a valuable strategy for augmenting the training dataset. In this context, the introduction of image variations through data augmentation enhances the model's capacity to generalize its learning to new images.

Our approach to data augmentation in this study adhered to conventional techniques [27], including operations such as horizontal or vertical flips, 90° rotations (both clockwise and counter-clockwise), and image flipping upside down, illustrated in Table 3. Additionally, we applied saturation adjustments within the range of −15%.

Figure 2. Images annotation using Roboflow framework.

Table 3. Applied augmentation approaches and corresponding parameters.

Augmentation Technique	Parameter
Flip	Horizontal and Vertical
Rotation	Clockwise, Counter-Clockwise, Upside Down by 90°
Saturation	[−15% to +15%]
Brightness	[−10% to +10%]
Exposure	[−6% to +6%]

3.2. Software Tools

Visual Studio: Visual Studio is a source code editor compatible with all operating systems. It is known for its speed and excellent readability, aided by a syntax highlighting feature that helps users understand the code's execution sequence. Visual Studio is chosen for its user-friendliness.

Anaconda: To streamline the setup process, we have opted for Anaconda, a platform that provides all the necessary packages for this work. Anaconda not only exhibits increased speed but also requires less storage space.

Python: Python, a widely adopted high-level programming language, is at the core of this research. With a vast array of standard libraries, Python grants users access to the necessary libraries for our proposed system.

Google Colab: For the execution and development of this work, we harnessed Google Colab, a free online platform that offers a Jupyter-Notebook-style environment for running Python code. Google provides this platform for machine learning and data science projects.

Roboflow: Roboflow is a comprehensive software platform that empowers users to create and manage autonomous robots. It encompasses an array of tools and features that facilitate the design, testing, and deployment of robots for diverse applications.

PyTorch: PyTorch, a free machine learning library for Python, is a valuable component of this study. Developed and maintained by Facebook's artificial intelligence research group, PyTorch is popular for its flexibility and modularity, enabling users to construct and train neural networks for a variety of applications, such as computer vision, natural language processing, time series prediction, etc.

3.3. Hardware Tools

Raspberry Pi 4B 2GB: A quad-core 64-bit ARM Cortex-A72 CPU running at 1.5 GHz, 2 GB of LPDDR4 SDRAM, Gigabit Ethernet, Bluetooth 5.0, and two USB 3.0 ports are all included in the Raspberry Pi 4 Model B. Additionally, it contains a microSD card slot, an HDMI connector, a display port, and a 40-pin extension header.

LM2596 Buck Converter: The LM2596 is a step-down switching regulator that can convert a DC input voltage to a lower DC output voltage. It has a wide input voltage range and can operate with input voltages from 3.5 V to 40 V. It has a built-in frequency compensation and can operate at a fixed frequency of 150 kHz or an adjustable frequency of up to 500 kHz.

Web Camera: We have used a full HD 1080P USB web camera. The model of our webcam is Xiaomi-CMSXJ22A. It has been used for capturing video.

3.5-inch LCD Display: Real-time Raspberry Pi monitoring and control have been performed using a 3.5-inch LCD in this work. It has a built-in touch system. So, Raspberry Pi can be easily controlled using this touchscreen display.

Figure 3 demonstrates the entire design of the proposed hardware visual pollution detection system. The system's total cost is approximately USD 160.

Figure 3. Design of the proposed hardware visual pollution detection system.

3.4. Applied Deep Learning Models

To accurately identify objects, it is essential to train an object detection model. This process assigns a class name to each object of interest within the image and outlines it with a bounding box. We conducted training on four distinct deep learning models—EfficientDet, SegFormer, YOLOv5, and YOLOv7—using our custom dataset, which contains visual pollutant images from nine different classes. The subsequent paragraphs describe the applied deep learning techniques for detecting and classifying various visual pollutants.

3.4.1. SegFormer

Transformers are employed in developing SegFormer [28], a computer vision framework for semantic segmentation tasks. Semantic segmentation refers to the model type, while transformers pertain to the architectural aspect, and the framework used is PyTorch.

As soon as the vision transformer (ViT) demonstrated substantial promise as a backbone, numerous studies began to build upon this concept and innovate to address issues related to poor resolution and high computational costs. Notably, these studies seemed to concentrate primarily on enhancing the design of the transformer encoder while neglecting the decoder, despite observing performance improvements with each new technique. An innovative encoder can handle any resolution without compromising performance.

In contrast to ViT, the encoder in SegFormer can produce both high- and low-resolution features. The decoder's design combines regional and global focus to generate high-quality representations efficiently. Thanks to these innovative advancements, SegFormer establishes a new state-of-the-art (SOTA) performance on prominent semantic segmentation datasets, including ADE20K, Cityscapes, and COCO-Stuff.

3.4.2. EfficientDet

EfficientDet stands out as a highly scalable design, mainly when operating with limited resources, delivering exceptional performance within minimal training epochs [29]. It falls under the category of object detection models and is implemented using the PyTorch framework with the COCO JSON annotation format. This work employs the PyTorch version of EfficientDet, originally developed in Tensorflow and Keras for real-time object recognition. It incorporates a custom detection and classification network combined with an EfficientNet backbone.

This foundation grants EfficientDet the capability to scale efficiently, even from its smallest model size, known as EfficientDet-D0, which comprises 4 million weight parameters. Despite its compact size, EfficientDet-D0 offers efficient inference, taking only 30 ms, and consumes a mere 17 megabytes of storage. This combination of compactness and speed makes it an excellent choice for various applications.

The compound scaling approach of the applied EfficientDet model involves maintaining a consistent ratio when scaling the dimensions of width, depth, and resolution, thereby achieving a balanced adjustment across these parameters, which is illustrated as:

$$\begin{gathered} Depth = j^{\theta}; Width = k^{\theta}; Resolution = l^{\theta} \\ \sqrt{j} \times k \times l \approx \sqrt{2} \end{gathered} \tag{1}$$

where the value of parameter (θ) increases with the increment in computational resources.

3.4.3. YOLOv5

For achieving high detection accuracy and rapid detection speed, the YOLOv5 model is highly preferable. YOLOv5 offers four models: YOLOv5x, YOLOv5l, YOLOv5s, and YOLOv5m. Among these, YOLOv5s and YOLOv5m are predefined simplified variants. These variants feature reductions in the size and number of model parameters, along with variations in the number of feature extraction modules and convolution kernels at specific network locations [29]. The input terminal of this model leverages adaptive image scaling, adaptive anchor generation, and mosaic data augmentation.

The input image is processed through an input layer and then directed to the backbone for feature extraction. This layer yields feature maps of varying sizes, fused through a feature fusion network (neck) to produce three final feature maps: P3, P4, and P5. Following the transmission of these feature maps to the prediction head, each pixel achieves confidence calculation and bounding-box regression using predefined prior anchors. This process generates a multi-dimensional array containing information on visual pollutant class, class confidence, box coordinates, width, and height. Finally, the array is filtered using specified thresholds to eliminate irrelevant information, and a non-maximum suppression process is applied to output the final detection information. The architecture of the applied YOLOv5 model is illustrated in Figure 4.

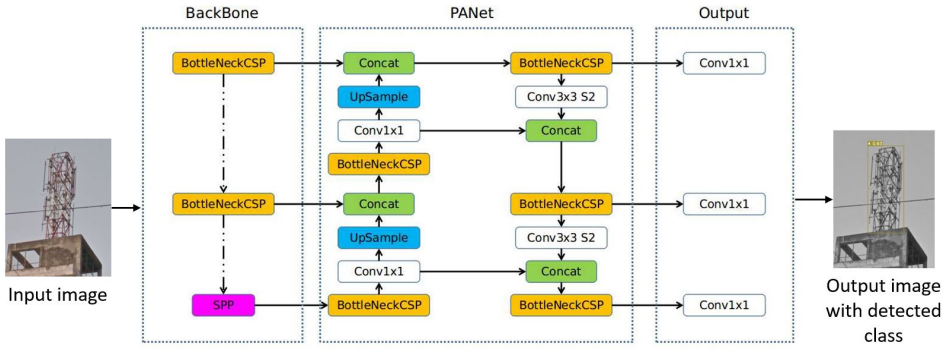

Figure 4. Architecture of the applied YOLOv5 model.

3.4.4. YOLOv7

WongKinYiu and AlexeyAB introduced YOLOv7, a cutting-edge object detection model, in 2022 [30]. YOLOv7 demonstrates exceptional performance on the MS COCO dataset and is specially designed for real-time object recognition across 80 different classes. This model is available in six distinct variants, ranging from the fastest and smallest with reduced accuracy to the potent YOLOv7-E6E, which is the slowest, largest, and most accurate among them. Various variants offer flexible solutions to meet a wide range of needs and requirements in object detection, i.e., input resolution for images, number of anchors, trainable parameters and layers, etc.

Table 4 displays important hyperparameters employed in the applied YOLOv5 and YOLOv7 models. Hyperparameters play a pivotal role in fine-tuning the model's accuracy and performance. The table presents these hyperparameters alongside their respective values.

Table 4. Hyperparameters with their corresponding values for the applied YOLO models.

Parameter	Value	Parameter	Value
lr0	0.01	warmup_epochs	3.0
lrf	0.01	warmup_momentum	0.8
momentum	0.937	box	0.05
weight_decay	0.0005	lou_t	0.2
hsv_s	0.7	anchor_t	4.0

3.5. Software and Hardware System Design

3.5.1. Software System Workflow

In this work, a robust algorithm for identifying industrial and street pollutants, including wires, billboards, bricks, street litter, fabric rubbish, industrial dye, and industrial billboards, categorizing visual contaminants associated with the textile industry, has been developed. Figure 5 illustrates the functional flowchart of the proposed PC-based software system. As depicted in Figure 5, we initially compiled nine categories of visual pollution images to train a model capable of performing various tasks. These categories will be used to train our model to recognize the different types of pollution present in those images and to learn the common characteristics within each image. To create a diverse dataset, we aggregated photos from search engines, various textile factories, and local streets in Bangladesh. After acquiring raw images, we utilized LabelImg to annotate the photographs. Subsequently, Roboflow was employed for the artificial preprocessing and enhancement of the images. Finally, the YOLOv5, YOLOv7, SegFormer, and EfficientDet deep learning models were utilized for the categorization of the visual pollutants.

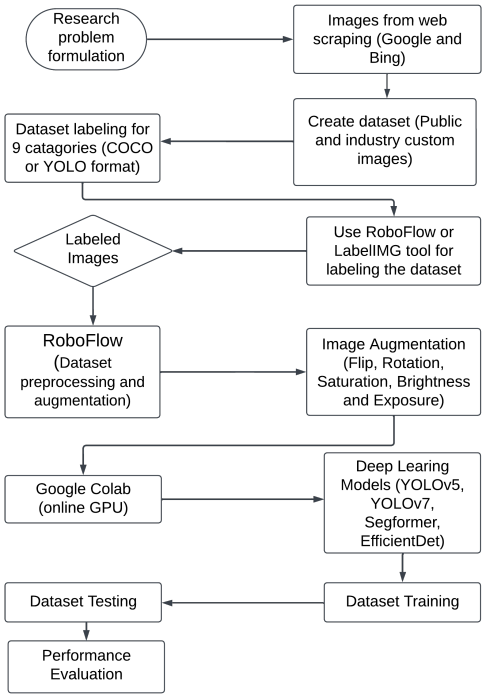

Figure 5. Proposed software system's working progressions.

3.5.2. Hardware System Workflow

The Raspberry Pi 4B, webcam, and YOLOv5 model are initialized. Following the training of the proposed YOLOv5 model with various visual pollutants, it is deployed on the Raspberry Pi 4B. The webcam captures a live video feed, which is then processed through the YOLOv5 framework for the detection and classification of visual pollutants. Subsequently, the detected pollutants are labeled, and their corresponding confidence scores are obtained. Working sequences of the proposed Raspberry Pi 4 and YOLOv5-based visual pollution detection hardware system are demonstrated in Figure 6.

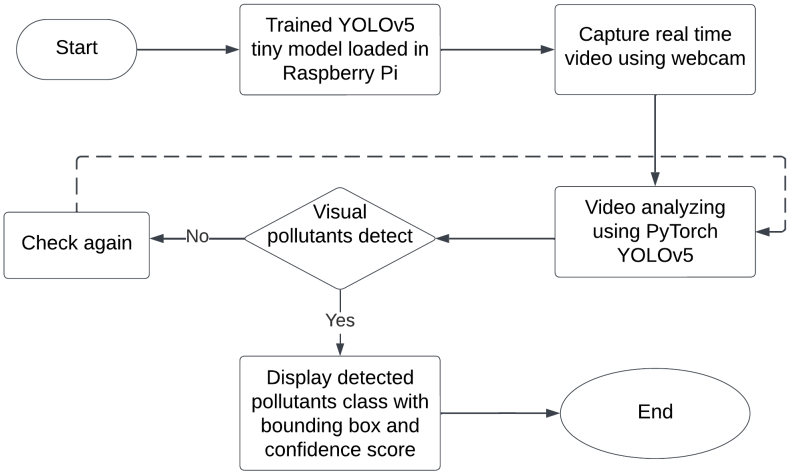

Figure 6. Working sequences of the proposed hardware system.

4. Result and Discussion

4.1. Result and Discussion of the Proposed Software System

The effectiveness of the proposed visual pollution detection and categorization system is carefully examined in this section. To construct a diverse dataset, this study collected

data from web searches, two nearby textile factories, roadside locations, and Bangladeshi retail centers. The visual pollution dataset has been categorized into nine target classes for image classification. The dataset was subdivided into training, validation, and testing. For the training session, 70% images of the comprehensive dataset were employed, while 20% and 10% of the samples were allocated to the validation and test datasets, respectively. To enhance the versatility of our model in generating three distinct neural network models, we incorporated several crucial elements discussed below.

- Loss Function—Much like the cross-entropy loss, this function is applied to detect errors or discrepancies in the model's learning process. The construction and classification of models employ sigmoid focal loss to estimate the probabilities for the YOLOv5, YOLOv7, EfficientDet, and SegFormer approaches. In multi-class classification problems, assigning greater weight to classes with the most pronounced imbalances is customary.
- Optimizer—In machine learning, optimization is a critical step that involves evaluating the prediction against the loss function to determine the optimal input weights. Stochastic gradient descent (SGD) has been employed for optimization instead of Adam to facilitate a more comprehensive comparison of the applied models. While the default setup options are effective in most scenarios, the configuration is straightforward.
- Epochs—The number of times that the model must be evaluated during the training phase varies. The YOLOv5 model has been trained for 200 epochs, while the YOLOv7 and EfficientDet models have undergone up to 10,000 epochs.
- Batch size refers to the number of training samples used in a single iteration. Each of the models has been configured with a batch size of 16.
- Various evaluation metrics are employed to assess how well our model performs. We demonstrated consistent evaluation criteria, including precision, recall, mean average precision, and intersection over union, for classification accuracy in the case of EfficientDet, Segformer, YOLOv5, and YOLOv7.
- For the training of each model, we employed annotated images in both COCO JSON and TXT formats.

For SegFormer, the IoU (intersection over union) is 0.698, while the loss is 0.256. The SegFormer model utilizes a learning rate of 0.08. In terms of validation accuracy, SegFormer achieves 0.678, and its validation precision is 0.782 in the context of semantic segmentation.

The classification loss function works as a measure of dissimilarity between the predicted probabilities and the actual labels. A lower classification loss corresponds to higher accuracy. As shown in Figure 7, after 10,000 epochs, we achieved a classification loss of 0.3. Additionally, across an average of 2000 to 10,000 epochs, the classification loss ranged from 0.2 to 0.4.

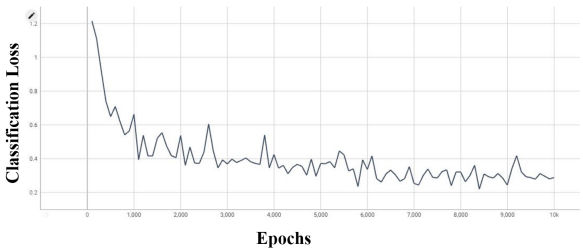

Figure 7. Classification loss in 10k epochs for the EfficientDet approach.

Figures 7 and 8 depict the classification and localization losses associated with the EfficientDet approach. These figures reveal a consistent reduction in losses as the training progresses. Notably, the overall and classification losses in the test curve reach their lowest values, approximately 0.10 and 0.20, respectively, reflecting the model's improved performance over time.

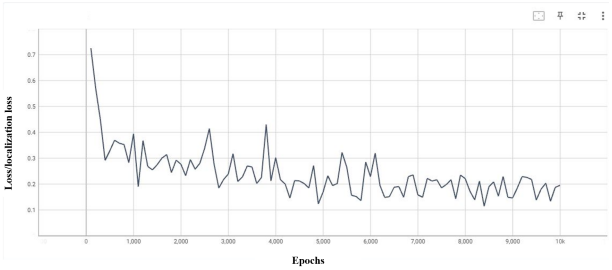

Figure 8. Localization loss in 10k epochs for the EfficientDet approach.

The total loss is a composite measure encompassing classification, localization, and regularization loss. In Figure 9, we observe a total loss of 0.50 for the EfficientDet model. A lower total loss indicates that the model makes fewer errors and better fits the data.

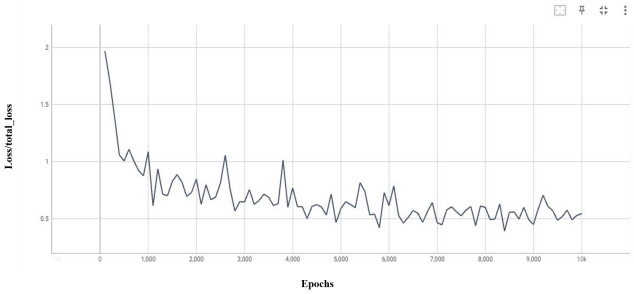

Figure 9. Total loss in 10k epochs for the EfficientDet approach.

Lastly, the EfficientDet framework achieved its highest accuracy, with an average performance of 87% on the advertising billboard and signage dataset, 83% on the textile dataset, and 89% on the street view dataset. Notably, the model demonstrated its best accuracy on the collected photographs of roadside and textile factory visual pollution.

For the YOLOv5 model, the employed dataset utilizes an average amount of GPU RAM. Classification losses show a significant decrease as training progresses. Figure 10 illustrates the training and validation classification losses of the YOLOv5 model over epochs. It is noteworthy that, at 200 epochs, we achieve the highest accuracy of 0.712 and the highest recall of 0.91. Figure 10 represents all performance metrics obtained with the YOLOv5 model.

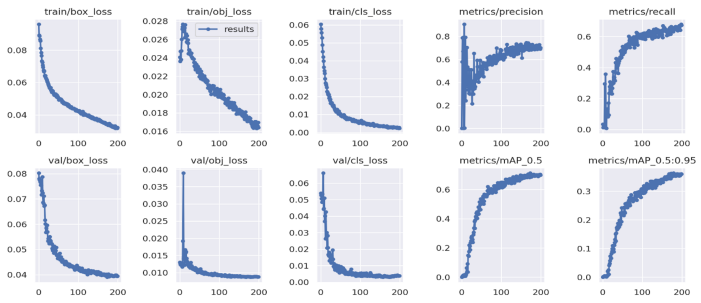

Figure 10. Various performance metrics with the change in epochs for the YOLOv5 model.

The normalized confusion matrix, representing the performance of the YOLOv5 model across the nine visual pollution categories, is illustrated in Figure 11. According to this figure, the YOLOv5 model achieved the highest accuracy in detecting the tower pollutant category. In contrast, it recorded the lowest accuracy for the entangled wire class.

Figure 12 illustrates that all classes achieve an F1 score of 0.68 at a confidence threshold of 0.296. This metric indicates that the model performs reasonably well in terms of both accuracy and completeness when detecting all classes of objects. In other words, the model correctly identifies 68% of the objects while minimizing false positive and false negative errors.

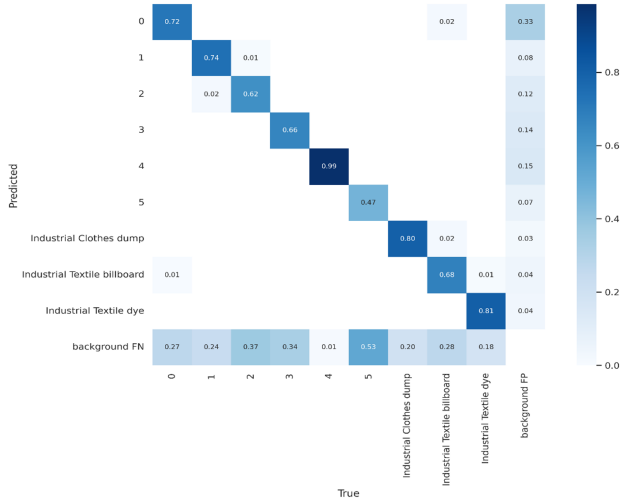

Figure 11. Normalized confusion matrix for the YOLOv5 model.

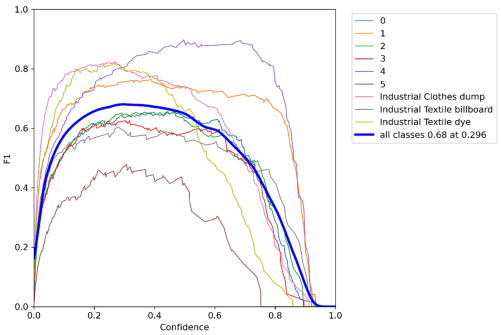

Figure 12. F1 confidence curve for the YOLOv5 model.

According to Figure 13, it is evident that all classes achieve a precision of 1.00 at a confidence threshold of 0.961. This scenario signifies that the model is highly accurate and confident in detecting all classes of objects at that threshold, effectively eliminating false positive errors, which are bounding boxes that do not match the ground truth labels. A higher confidence threshold indicates that the applied model is precise and confident in its predictions.

Figure 14 demonstrates that all classes have a precision and recall of 0.712 at a confidence threshold of 0.50. This scenario implies that the model is moderately accurate in detecting all categories of objects at the corresponding threshold. It correctly identifies 71.2% of the objects while minimizing false positive errors. However, it also indicates that the model misses 28.8% of the objects and makes some false negative errors.

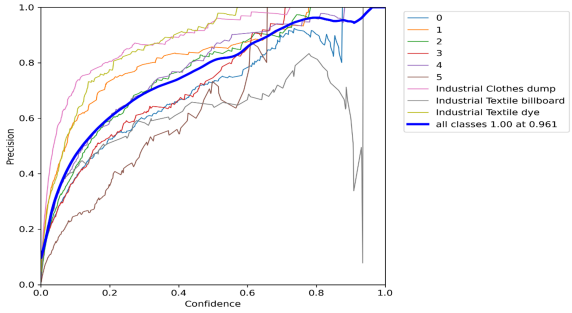

Figure 13. Precision vs. confidence for the YOLOv5 model.

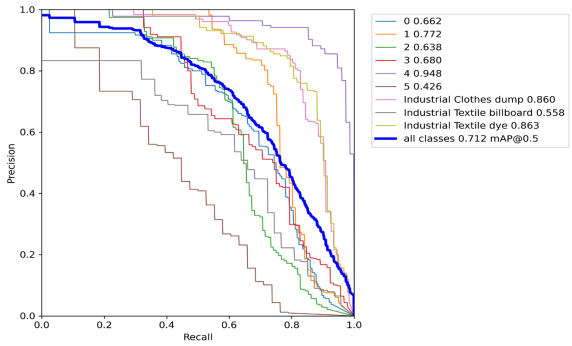

Figure 14. Precision vs. recall for the YOLOv5 model.

Figure 15 demonstrates that, at a confidence threshold of 0.00, all classes achieve a recall of 0.91. This indicates the model's comprehensiveness in detecting objects, as it captures 91% of objects while minimizing misclassification. The confidence threshold serves as the minimum probability assigned by the model to validate a bounding box as a detection. On the other hand, precision quantifies the model's accuracy in detecting objects, measuring the ratio of true positives to all positives.

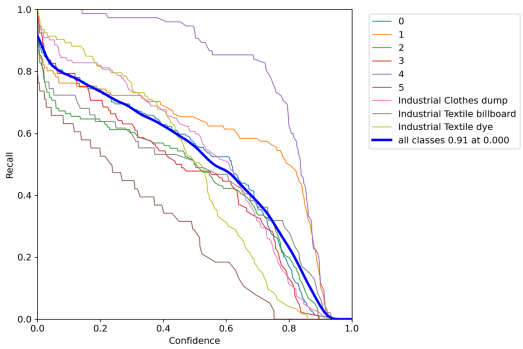

Figure 15. Recall vs. confidence for the YOLOv5 model.

Figures 12–15 offer insights into precision, recall, F1 score, and confidence curves as they evolve over epochs. Notably, the model achieves an average performance of 0.712 over 200 epochs. Figure 10 illustrates the mean average precision (mAP) of the YOLOv5 model at a fixed threshold of 0.50 intersection over union (IoU).

During the training of the employed dataset, we leveraged the Google Colab online GPU RAM provider infrastructure for running the YOLOv7 model. As the duration of the training increased, there was a substantial reduction in classification losses. This training progress is visually depicted in Figure 16, illustrating the evolution of training and validation classification losses for the YOLOv7 model over different epochs. Notably, the YOLOv7 model achieved its peak accuracy of 0.667 and the highest recall of 0.657 after 200 training epochs.

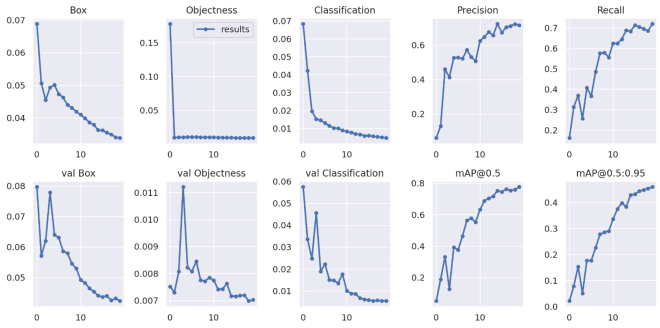

Figure 16. Various performance metrics with the change in epochs for the YOLOv7 model.

Figure 17 presents the normalized confusion matrix for the YOLOv7 model for various visual pollution categories. Similar to the applied YOLOv5 architecture, the YOLOv7 technique attained the highest accuracy for detecting the tower class.

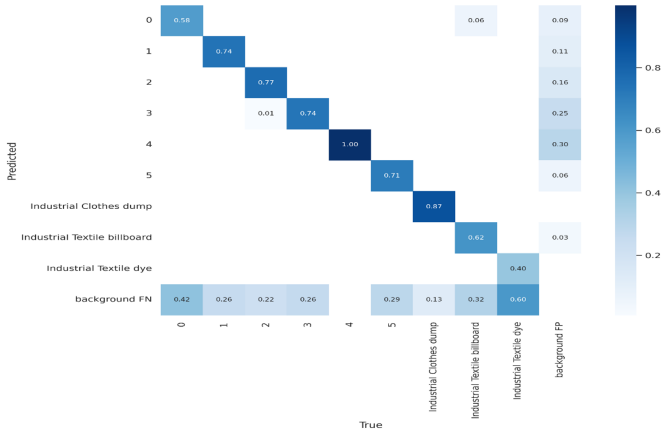

Figure 17. Normalized confusion matrix for the YOLOv7 model.

Table 5 displays a range of performance metrics for the deep learning models that we employed in this research. The results in this table indicate that the YOLOv5 model outperforms all other models, achieving the highest mean average precision (mAP) score of 0.712 at IoU 0.50.

Table 5. Various performance metrics for the applied models.

Model	mAP	IoU	Precision	Recall
YOLOv7	0.667	0.50	0.636	0.657
YOLOv5	0.712	0.50	0.703	0.666
SegFormer	0.597	0.698	0.782	0.546
EfficientDet	0.689	0.50	0.734	0.798

Table 6 presents the testing accuracies achieved by the various deep-learning-based classification models. Notably, the YOLOv5 approach stands out with the highest accuracy, achieving 98%. As a result of its outstanding performance, YOLOv5 has been selected for deployment on the edge device. In contrast, the SegFormer framework demonstrated the lowest accuracy, achieving a score of 73%.

Table 6. Accuracies of the applied models.

Model	Detection Accuracy Rate
YOLOv7	92%
YOLOv5	98%
EfficientDet	86%
SegFormer	73%

4.2. Hardware Proposed System

The proposed automatic visual pollution detection system has been integrated into a robust hardware setup, featuring a Raspberry Pi 4B microcontroller, a 3.5-inch touchscreen display, and a Xiaomi-CMSXJ22A web camera. This hardware configuration is optimized to execute the YOLOv5 model efficiently. Extensive testing has been conducted at various

locations in Dhaka, Bangladesh, to validate the performance and reliability of the proposed automatic visual pollution detection hardware device. The tests encompassed diverse lighting conditions, backgrounds, and variations in the distances between objects and the camera. The device consistently and instantaneously detects various visual pollutants, effectively identifying the detected class, as demonstrated in Figure 18. These rigorous tests confirm the robustness and adaptability of the proposed system in real-world scenarios, making it a valuable tool for addressing visual pollution concerns.

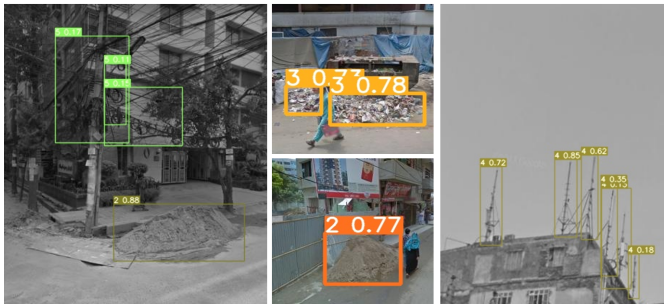

Figure 18. Real time detection of various visual pollutants using Raspberry Pi.

Table 7 provides a comparison between the proposed visual pollution classification system and existing works. A significant distinction is that most of the other works utilized publicly available images and did not deploy deep learning techniques on embedded devices. In contrast, our research leverages a comprehensive dataset of 3000 images spanning nine visual pollution classes. Furthermore, we successfully implemented the proposed detection system on the Raspberry Pi edge device, enabling real-time classification.

Table 7. Comparison of the proposed visual pollution system with other similar works.

Ref	Dataset (# of Images)	Model	Hardware Implementation	Accuracy
[15]	Public (1200)	CNN	No	85.09%
[17]	Public (1400)	YOLOv5	No	83.17%
[18]	Public	Neural Network	No	96.10%
[28]	Public	Customized CNN	No	87.70%
[29]	Public	ResNetV2	No	91.40%
This work	Public and custom (3000)	YOLOv5	Yes with Raspberry Pi 4	98.40%

4.3. Complexity and Limitations of the Proposed System

The proposed system for visual pollution detection using a deep learning network and a robotic vision system integrated with an edge device involves various complexities. For instance, precision in labeling using tools like LabelImg and Roboflow API requires meticulous categorization and annotation of each image. The seamless integration of the selected model (YOLOv5) into the hardware setup presents additional technical challenges, potentially leading to reduced performance.

The dataset employed in this work primarily focused on visual pollution in Dhaka, Bangladesh. Generalizing the findings to other cities or regions may be challenging due to variations in urban landscapes, infrastructure, and cultural contexts. The detection performances of the implemented deep learning models could be affected by changes in weather conditions, lighting, and seasonal variations. The proposed hardware setup using the Raspberry Pi 4B-embedded system may involve processing power and image resolution limitations.

5. Conclusions and Future Work

The relatively recent term in environmental management, visual pollution, encompasses various unappealing visual elements that disrupt the spatial aesthetics of urban environments. This aesthetic deterioration includes billboards, bricks, construction materials, street litter, communication towers, and entangled electric wires. Our research addresses the automatic identification and categorization of nine distinct visual pollutants. The initial phase involved collecting open-source images of visual pollutants through web crawling performed by Google and Bing. Subsequently, we gathered photographs of aesthetic pollution from diverse sources within Dhaka, the capital of Bangladesh, including textile factories, retail malls, and street scenes. To tackle the classification challenges presented by this dataset, we employed various deep-learning approaches, namely SegFormer, EfficientDet, YOLOv5, and YOLOv7. The selected model for deployment on our hardware setup is YOLOv5, which operates on a Raspberry Pi 4 microcontroller. With this integrated hardware setup, we conducted comprehensive testing and data collection to assess the model's accuracy and overall performance. The results of this study hold the potential to assist relevant authorities in identifying and mitigating instances of environmental visual pollution. As a future direction, this method can be further enhanced to develop an Android application capable of real-time visual pollution detection through streaming video cameras. Additionally, exploring the utilization of more powerful embedded systems for visual pollution identification represents another avenue for potential research and development. Enhancements to the suggested system's performance can be achieved by incorporating advanced image processing techniques and expanding the dataset of pollution data to enrich the model's capabilities.

Author Contributions: M.F.S.T. wrote the manuscript text; A.A.C. and S.M.R.H. analyzed the data and prepared the figures; M.F.S.T. and R.K. reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by North South University, Dhaka, Bangladesh, under the grant CTRG-21-SEPS-25.

Institutional Review Board Statement: All authors have read, understood, and have complied as applicable with the statement on "Ethical responsibilities of Authors" as found in the Instructions for Authors.

Data Availability Statement: Data will be made available upon request.

Conflicts of Interest: The authors have no competing financial and/or non-financial interests.

References

1. Salem, A.A.; Lau, K.Y.; Rahiman, W.; Al-Gailani, S.A.; Abdul-Malek, Z.; Rahman, R.A.; Rahman, R.A.; Sheikh, U.U. Pollution Flashover Characteristics of Coated Insulators under Different Profiles of Coating Damage. *Coatings* **2021**, *11*, 1194 . [CrossRef]
2. Gu, K.; Liu, H.; Liu, J.; Yu, X.; Shi, T.; Qiao, J. Air Pollution Prediction in Mass Rallies With a New Temporally-Weighted Sample-Based Multitask Learner. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–15. [CrossRef]
3. Hulagu, S.; Celikoglu, H.B. Environment-Friendly School Bus Routing Problem With Heterogeneous Fleet: A Large-Scale Real Case. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 3461–3471. [CrossRef]
4. Zhang, H.; Zhou, Z.; Ding, L.; Wu, C.; Qiu, M.; Huang, Y.; Jin, F.; Shen, T.; Yang, Y.; Hsu, L.; et al. Divergent and Convergent Imaging Markers Between Bipolar and Unipolar Depression Based on Machine Learning. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 4100–4110. [CrossRef] [PubMed]
5. Ren, K.; Wu, Y.; Zhang, H.; Fu, J.; Qu, D.; Lin, X. Visual Analytics of Air Pollution Propagation Through Dynamic Network Analysis. *IEEE Access* **2020**, *8*, 205289–205306. [CrossRef]
6. Deng, Z.; Weng, D.; Chen, J.; Liu, R.; Wang, Z.; Bao, J.; Zheng, Y.; Wu, Y. AirVis: Visual Analytics of Air Pollution Propagation. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 800–810. [CrossRef] [PubMed]
7. Lyu, C.; Chen, Y.; Alimasi, A.; Liu, Y.; Wang, X.; Jin, J. Seeing the Vibration: Visual-Based Detection of Low Frequency Vibration Environment Pollution. *IEEE Sens. J.* **2021**, *21*, 10073–10081. [CrossRef]
8. Zhang, X.; Wang, D.; Jiang, F.; Lin, T.; Xiang, H. An Optimal Regulation Method for Parallel Water-Intake Pump Group of Drinking Water Treatment Process. *IEEE Access* **2020**, *8*, 82797–82803. [CrossRef]

9. Ajayi, O.O.; Bagula, A.B.; Maluleke, H.C.; Gaffoor, Z.; Jovanovic, N.; Pietersen, K.C. WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes. *IEEE Access* **2022**, *10*, 48318–48337. [CrossRef]
10. Saad, A.; Benyamina, A.E.H.; Gamatié, A. Water Management in Agriculture: A Survey on Current Challenges and Technological Solutions. *IEEE Access* **2020**, *8*, 38082–38097. [CrossRef]
11. Tiyasha, T.; Bhagat, S.K.; Fituma, F.; Tung, T.M.; Shahid, S.; Yaseen, Z.M. Dual Water Choices: The Assessment of the Influential Factors on Water Sources Choices Using Unsupervised Machine Learning Market Basket Analysis. *IEEE Access* **2021**, *9*, 150532–150544. [CrossRef]
12. Wu, D.; Wang, H.; Mohammed, H.; Seidu, R. Quality Risk Analysis for Sustainable Smart Water Supply Using Data Perception. *IEEE Trans. Sustain. Comput.* **2020**, *5*, 377–388. [CrossRef]
13. Chopade, S.; Gupta, H.P.; Mishra, R.; Kumari, P.; Dutta, T. An Energy-Efficient River Water Pollution Monitoring System in Internet of Things. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 693–702. [CrossRef]
14. Wan, L.; Sun, Y.; Lee, I.; Zhao, W.; Xia, F. Industrial Pollution Areas Detection and Location via Satellite-Based IIoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1785–1794. [CrossRef]
15. Ahmed, N.; Islam, M.N.; Tuba, A.S.; Mahdy, M.; Sujauddin, M. Solving visual pollution with deep learning: A new nexus in environmental management. *J. Environ. Manag.* **2019**, *248*, 109253. [CrossRef] [PubMed]
16. Andjarsari, S.; Subadyo, A.T.; Bonifacius, N. Safe Construction And Visual Pollution Of Billboards Along Main Street. *IOP Conf. Ser. Earth Environ. Sci.* **2022**, *999*, 012015. [CrossRef]
17. Hossain, M.Y.; Nijhum, I.R.; Sadi, A.A.; Shad, M.T.M.; Rahman, R.M. Visual Pollution Detection Using Google Street View and YOLO. In Proceedings of the Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, New York, NY, USA , 1–4 December 2021; pp. 433–440. [CrossRef]
18. Yang, Z.; Li, D. WasNet: A Neural Network-Based Garbage Collection Management System. *IEEE Access* **2020**, *8*, 103984–103993. [CrossRef]
19. Mittal, G.; Yagnik, K.B.; Garg, M.; Krishnan, N.C. SpotGarbage: Smartphone App to Detect Garbage Using Deep Learning. In Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 940–945.
20. Marin, I.; Mladenović, S.; Gotovac, S.; Zaharija, G. Deep-Feature-Based Approach to Marine Debris Classification. *Appl. Sci.* **2021**, *11*, 5644. [CrossRef]
21. Tasnim, N.H.; Afrin, S.; Biswas, B.; Anye, A.A.; Khan, R. Automatic classification of textile visual pollutants using deep learning networks. *Alex. Eng. J.* **2023**, *62*, 391–402. [CrossRef]
22. Bakar, S.A.; al Sharaa, A.; Maulan, S.; Munther, R. Measuring Visual Pollution Threshold along Kuala Lumpur Historic Shopping District Streets Using Cumulative Area Analysis. In Proceedings of the Visual Resource Stewardship Conference, Lemont, IL, USA, 27–30 October 2019.
23. Setiawan, W.; Wahyudin, A.; Widianto, G. The use of scale invariant feature transform (SIFT) algorithms to identification garbage images based on product label. In Proceedings of the International Conference on Science in Information Technology, Bandung, Indonesia, 25–26 October 2017; pp. 336–341.
24. Ahmed, I.; Din, S.; Jeon, G.; Piccialli, F.; Fortino, G. Towards Collaborative Robotics in Top View Surveillance: A Framework for Multiple Object Tracking by Detection Using Deep Learning. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1253–1270. [CrossRef]
25. AlElaiwi, M.; Al-antari, M.A.; Ahmad, H.F.; Azhar, A.; Almarri, B.; Hussain, J. VPP: Visual Pollution Prediction Framework Based on a Deep Active Learning Approach Using Public Road Images. *Mathematics* **2023**, *11*, 186. [CrossRef]
26. Sun, Y.; Loparo, K. Context Aware Image Annotation in Active Learning with Batch Mode. In Proceedings of the Annual Computer Software and Applications Conference, Milwaukee, WI, USA, 15–19 July 2019; Volume 1, pp. 952–953. [CrossRef]
27. Wang, S.; Yang, Y.; Wu, Z.; Qian, Y.; Yu, K. Data Augmentation Using Deep Generative Models for Embedding Based Speaker Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2598–2609. [CrossRef]
28. Qiu, C.; Li, H.; Guo, W.; Chen, X.; Yu, A.; Tong, X.; Schmitt, M. Transferring Transformer-Based Models for Cross-Area Building Extraction From Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4104–4116. [CrossRef]
29. Mekhalfi, M.L.; Nicolò, C.; Bazi, Y.; Rahhal, M.M.A.; Alsharif, N.A.; Maghayreh, E.A. Contrasting YOLOv5, Transformer, and EfficientDet Detectors for Crop Circle Detection in Desert. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
30. Wang, C.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]