Construction Vehicle Activity Detection in Low-Frequency Surveillance Imagery and Its Relationship to Local Air Quality

DAN-CRISTIAN PLOESTEANU, University of Twente, The Netherlands

Construction vehicles are both central to construction site workflows and major contributors to local air pollution. This paper develops a 3-stage machine learning pipeline that uses sparse on-site surveillance imagery to detect and classify construction vehicle activity and quantify its relationship to ambient air quality. The pipeline comprises a detection model based on the YOLOv9 architecture, a construction vehicle activity classification model (for which two contrasting architectures are tested, including a ViT-based method and an SVM-based model) and a linear regression analysis between detected vehicle activities and local air quality indicators. Despite operating on low-frequency (5-minute interval) imagery under real-world conditions, the proposed models achieve state-of-the-art performance in both detection and activity inference. Regression analysis reveals a statistically significant but limited correlation between vehicle activity and local pollutant concentrations, suggesting the presence of dominant external sources. These findings demonstrate the feasibility of passive, vision-based environmental sensing in constrained urban deployments and open new avenues for integrating ubiquitous computing with sustainable construction monitoring.

1 INTRODUCTION

Construction sites are hidden hotspots of urban pollution, yet their dynamic operations remain poorly captured by traditional monitoring systems. In urban areas, they are one of the most important contributors to particulate matter (PM) pollution [22] as well as significant contributors to NO_x and greenhouse gases (GHG) pollution [32].

Monitoring air pollution on construction sites is important for assessing its impact on the population, monitoring regulatory compliance, and deploying countermeasures in a timely manner [34]. However, the task of monitoring air pollution is complex. Accurate monitoring typically requires expensive sensors, which limits their wide-scale deployment [34]. In recent years, low-cost sensors have become more widely available; however, they are often associated with lower accuracies and are more prone to interference caused by environmental conditions [5].

To address the limited scalability of sensor-based monitoring, this work explores whether surveillance imagery, already captured by many sites, can offer reliable pollution indicators. One possible way of using such imagery is to monitor the activity of construction vehicles (e.g. dump trucks, excavators, mobile cranes) on the site. Construction vehicle activity may serve as a predictor of air pollution, since these vehicles are not only a significant source of pollution in themselves (mainly due to their heavy-duty diesel engines), but their use can also be indicative of construction activities that generate pollution (such as excavations or movement across unpaved roads) [2]. However, for construction vehicle activities to be used as a predictor of air pollution levels, it must first be established if there exists a clear link between air pollutant levels and construction vehicle activity and whether that link can reliably be observed using computer vision techniques.

Existing literature on computer vision techniques for the identification of construction vehicle activities has predominantly concentrated on video data, as shown by Sherafat et al. in their review of activity recognition techniques [27] and by the more recent work of Kim et al. [18] and Küpers et al. [19] which have focused on enabling real-time analysis of video data. Nonetheless, common camera infrastructure deployed on construction sites is typically configured to capture images at substantial time intervals (e.g., every 5 minutes), instead of videos, thereby leading to suboptimal performance of existing computer vision algorithms when integrated with real-world systems.

The goal of this paper is therefore to investigate whether construction vehicle activity detected in camera images taken at substantial time intervals can be correlated with changes in air pollutant levels, as measured by nearby sensors. This goal motivates the following research question:

"To what extent can construction vehicle activity, detected via computer vision in images captured at set time intervals, be used to model variations in local air quality near construction sites?"

To better answer this research question, it can be further subdivided into four smaller sub-questions:

(1) To what extent are state-of-the-art computer vision model architectures suitable for detecting and tracking construction vehicles in temporally sparse surveillance camera images?
(2) What computer vision approaches can reliably classify construction vehicle activity (stationary, operating, moving) from temporally sparse image pairs?
(3) Which detected construction vehicle activities show statistically significant relationships with measured air quality metrics, under a linear modelling assumption?

To answer these questions, a 3-stage machine learning pipeline that detects and classifies construction vehicle activity from low-frequency surveillance imagery and statistically relates this activity to local air quality metrics will be developed. The remainder of this paper is structured as follows. In Section 2, a literature review which covers the state of the art related to these tasks is presented. Section 3 then explains the methodology of this study in detail. Section 4 presents the results. Section 5 interprets the results of this research, comparing them with existing literature, and discussing the limitations of the proposed methodology. Finally, Section 6 concludes the paper by concisely stating the answers to the posed research questions.

TScIT 43, July 4, 2025, Enschede, The Netherlands

2 BACKGROUND AND RELATED WORKS

To mitigate the issues associated with traditional sensing techniques, machine learning (ML) has been used to predict air pollution levels from camera images [29, 16, 39, 38]. Camera images are an attractive way of monitoring construction site pollution because many sites already have surveillance cameras installed to prevent trespassing and remotely monitor construction activities. These studies employ deep neural networks to extract information about pollutant levels from images, combining that information with data on atmospheric conditions obtained from local sensors. Although these studies demonstrate that computer vision can provide a reliable alternative to traditional pollution sensors, the techniques used offer no insight into the causes of the pollution that is being recorded.

The problem of construction vehicle detection is discussed in literature as early as 2016, with one of the first approaches being based on foreground detection [14]. However, newer studies almost exclusively use deep learning models, thanks to their superior performance in complex environments. For instance, Arabi et al. [3] applied a MobileNet-based [10] single-shot detector to identify construction vehicles with over 90% mean average precision. Küpers et al. [19] employed the single-shot YOLOv8 (You Only Look Once) model [31] to optimise the detection of construction vehicles on devices with limited resources, achieving 80% precision and real-time video processing. Zhang et al. [41] further extended this paradigm by using a self-supervised training approach to improve the accuracy of a YOLOv4 model beyond that achieved with classical supervised training. These models share a common anchor-based structure, where predefined regions (anchor boxes) are used to localise and classify objects [4]. In contrast, anchor-free methods (e.g., Guo et al. [9]) offer faster inference while maintaining comparable accuracy, which is particularly relevant for embedded edge deployment. A review of the various detection models available in early 2024 was conducted by Chen et al. [4], which lists the advantages and disadvantages of all detection model architectures available then, though it is not specifically focused on detecting construction vehicles.

Object tracking (i.e., the task of maintaining identity across frames) has typically been tackled using algorithms designed for high frequency video. ByteTrack [40] and BoT-SORT [1] are state-of-the-art methods, which both rely on differences in an object's bounding box coordinates, with the latter also integrating visual appearance for robust re-identification. However, since both assume video-like continuity, they may not generalise well to low-frame-rate imagery, such as the 5-minute interval images commonly captured on construction sites.

Activity recognition, in contrast, is less explored in the context of construction equipment. Küpers et al. [19] applied logistic regression on bounding box centroid and area differences to classify activity as idle or non-idle. More expressive temporal models like Bi-LSTMs were used by Kim et al. [18] to classify excavator states from video, while Slaton et al. [28] used LSTMs to classify accelerometer data from sensors mounted on an excavator and a roller.

The current study aims to supplement this existing literature by evaluating a new model architecture for construction vehicle detection and two novel approaches for classifying construction vehicle activity, which are adapted to images taken at significant time intervals instead of video data. Furthermore, we extend this by linking detected activities with air quality sensor data, providing insight into the environmental footprint of machine operation, an aspect underrepresented in prior work.

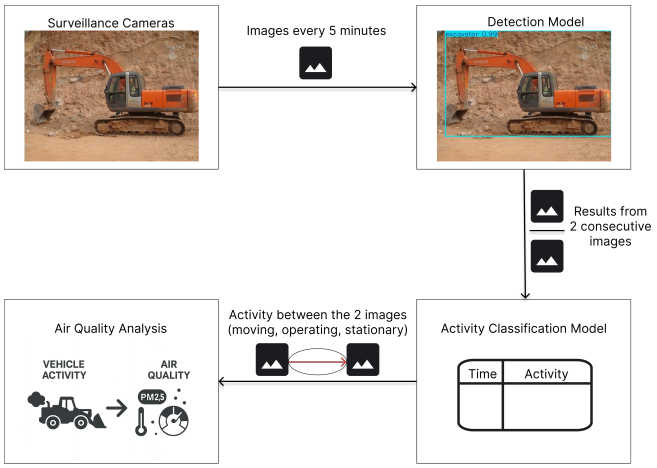

Fig. 1. Simplified overview of the project methodology

3 METHODOLOGY

A high-level overview of the methodology is presented in Figure 1. The study begins with pictures taken every 5 minutes by surveillance cameras positioned on a construction site. These images are first processed by the detection model, which is responsible for drawing bounding boxes around each construction vehicle and for identifying its type. Two consecutive images, along with the information produced by the detection model for these images, form the input of the activity classification model. Two different architectures of activity classification models are tested; for one of these two architectures, tracking information is also passed to the activity classification model along with the information produced by the detection model. The activity classification model is responsible for determining what activity a construction vehicle undertook in the interval between two images. There are three possible activities: *moving* (the vehicle has changed its location considerably between pictures), *operating* (the vehicle remained in roughly the same location, but part of the vehicle was moved, such as an excavator's bucket) and *stationary* (the vehicle was not in use). In the final stage of the study, the information about vehicle activity over time is combined with air quality sensor readings and data about atmospheric conditions to analyse the relationship between vehicle activity and air quality. The following subsections will explain each of these stages in greater detail and describe the data that is used for each stage.

3.1 Datasets

3.1.1 The Amsterdam Dataset. All stages of this study require data, either for training or as input for inference. The first dataset employed by this study is a dataset of 47850 images taken every 5 minutes by surveillance cameras from a construction site in Amsterdam, the Netherlands (which will be referred to as the *The Amsterdam*

Dataset). The surveillance camera images come from 4 different cameras, positioned around the perimeter of the site, on poles at a height of 6.2m. The construction site is lit up in green at night, and the cameras are subject to varying natural light conditions, which can lead to colour distortions in some images (images turning greyscale, or having a pink or green hue). This dataset is confidential due to privacy concerns, and no images from it will be displayed in this paper.

3.1.2 ACID & MOCS Datasets. To aid in training the detection model, two additional open datasets are used. These are the *Alberta Construction Image Dataset* (ACID) [35] and the *Moving Objects in Construction Sites* (MOCS) dataset [36]. These datasets contain a varied and numerous collection of images taken on construction sites around the world, along with manually verified bounding box annotations meant for construction vehicle detection. Of the vehicle classes present in these two datasets, only excavators, trucks, concrete mixers, and mobile cranes are present in the Amsterdam dataset, so only these will be used for training. In total, after selecting only these classes, the combined ACID-MOCS detection training dataset is made up of 22369 images containing 47575 instances of construction vehicles. The combined dataset exhibits a moderate class imbalance, with excavators being the largest class and bulldozers the smallest. The full class distribution is available in Figure 3 of Appendix C.

3.1.3 Air Quality Dataset. The final dataset that is used is a dataset of air quality sensor readings measured on the same construction site and at the same times as the pictures in the Amsterdam Dataset. This dataset contains readings for the concentrations of the following air quality metrics/pollutant levels measured every five minutes: NO_2, CO_2, O_3, PM_1, $PM_{2.5}$, PM_{10}. In addition, the dataset also records the values of the following atmospheric conditions: temperature, relative humidity, and pressure.

3.2 Data Preprocessing

The Amsterdam dataset was first anonymised to protect the privacy of the workers on the construction site by using a detection model similar to the one described in Subsection 3.3 to mask people and license plates (see Appendix D for an example). A low confidence threshold (0.08) was chosen to ensure that no sensitive information would remain after anonymisation. The anonymised pictures were manually checked by a reviewer to validate the results of the detection model.

After anonymisation, some of the images in the Amsterdam dataset are annotated using the online computer vision annotation tool Roboflow [26] for future use in model training and validation. To aid in training the detection model, 300 images, chosen to include a diverse selection of vehicles in various lighting conditions, are manually annotated with bounding boxes. The resulting annotations contain a moderate class imbalance, with excavators and trucks again being the largest classes. The full class distribution of these annotations is available in Appendix C, Figure 4 (note that many images contain more than one vehicle, so the total number of annotated vehicles is greater than the number of images).

Similarly, for activity classification, 2861 randomly selected pseudo-bounding box pairs are annotated. A pseudo-bounding box pair is the association between a bounding box in one image and the same coordinates in the following image of the same camera, regardless of whether the detection model would have still identified a vehicle at the same coordinates in that image. This annotation process resulted in an imbalanced dataset, with the following activity class distribution: 57.7% stationary, 21.7% moving and 20.6% operating.

Another preprocessing step that has to be taken is converting the annotations of the ACID and MOCS datasets into the YOLO format that the detection model will require. The ACID dataset is provided in the COCO format [6], while the MOCS dataset uses a third annotation format: PASCAL VOC [7]. Converting both types of annotations to the YOLO format is done using a Python script which makes use of the pylabel library [8].

Both the Amsterdam dataset and the combined ACID-MOCS dataset are split into train, validation and test sets, at a ratio of 7:2:1, using a stratified random split. In small data datasets with class imbalance, such as those of this study, a stratified random split is preferable to a simple random split as it prevents random chance from affecting the class distribution across dataset splits, thereby ensuring more reliable model training and evaluation.

All values of air pollution metrics in the air quality dataset are cleaned by removing outliers (values beyond three standard deviations of the mean). Finally, to allow them to be used for statistical inference, each observation's time of day in the air quality dataset must be encoded. While timestamps are initially provided for each observation, only the hour and minute are retained. Adopting the naive approach of encoding each 5-minute interval as a dummy variable would lose the ordering and cyclicity of the times of day. Instead, time is first converted to a decimal value (hour + minutes/60) to preserve ordering. To capture cyclicity, a trigonometric transformation is then applied, representing each time as two features:

$$\sin\left(\frac{2\pi t}{24}\right) \quad \text{and} \quad \cos\left(\frac{2\pi t}{24}\right),$$

where t is the decimal time.

3.3 Detection Model

3.3.1 Model Selection. The detection model's task is to identify the position (bounding box coordinates) and class of a construction vehicle in a picture. A YOLO model is chosen thanks to the architecture's proven performance in detecting construction vehicles [19, 41]. The architecture's latest version is YOLOv12 [30]; however, a recent review conducted by Jegham et al. [13] shows that this version largely underperforms previous YOLO iterations. Furthermore, this review indicates that YOLOv9 [33] outperforms all other YOLO versions for detecting large objects, using small training datasets and in complex situations where there are multiple overlapping objects. This makes YOLOv9 a promising candidate for this study, since many images in the Amsterdam dataset exhibit these characteristics. There are five versions of the YOLOv9 model available, with parameter counts ranging from 2M (YOLOv9t) to 58.1M (YOLOv9e), which present a trade-off between inference speed and higher accuracy. For this study, the largest and most accurate version (YOLOv9e) is chosen, since real-time inference is outside the scope of the current study,

Table 1. Detection Model Training Hyper-parameters

Hyperparameter	First Round	Fine-tuning
Epochs	100	50
Patience	15	10
Batch size	12	2
Optimiser	SGD	AdamW
Learning rate	10^{-2}	10^{-4}
Momentum	0.9	0.9
Image size	640	640

and even if the model were deployed for real-time inference, the long interval between pictures taken by cameras such as the ones used for the Amsterdam dataset (5 minutes) affords ample time for inference, even in resource-constrained environments.

3.3.2 Model Training. The YOLOv9 model is trained using the Ultralytics Python library [15]. The model is first initialised with weights pre-trained on the MS COCO dataset [21]. This has the advantage of reducing model training time and improving its performance, since the model is already capable of recognising basic shapes and patterns before it starts training on the specialised task needed for this project.

Training the model is done in two rounds. During the first round, the model is trained on the combined ACID-MOCS dataset. For this round, the model is trained using the hyperparameters specified in Table 1. All hyperparameters not specified in Table 1 retain their default values from the Ultralytics model training function.

After the initial round of training, the model is fine-tuned during a second round of training, on the 300 labelled pictures from the Amsterdam dataset. This is necessary because the angle from which pictures in the Amsterdam dataset are taken is unusual, being neither from ground level nor from high above (as would be the case for images taken using a UAV). Because of this unusual angle, the model trained only on the ACID-MOCS dataset does not generalise well to the Amsterdam dataset without further fine-tuning. The hyperparameters used in this second round of training are also shown in Table 1.

To address potential model bias caused by the class imbalance of both training sets, the YOLO Ultralytics implementation offers two solutions, which are used in this study. The first is the use of a custom loss function: distributed focal loss (DFL). DFL is a loss function which aims to address class imbalance by assigning higher weights to challenging instances and therefore penalising models that are biased against the minority classes [20]. The second measure taken to address the imbalanced class distribution is to make use of the on-the-go data augmentation techniques built into the Ultralytics library. All images are augmented in every training epoch, before being used as input to the model, by randomly changing their hue, saturation, brightness or scale, by making mosaics of crops from 4 different images and by horizontally flipping the images. The default probabilities for these augmentations are used. Data augmentation helps combat class imbalance by artificially creating more variability for the under-represented classes and therefore helping the model better generalise for examples of these classes.

Table 2. Modified Tracker parameters

Parameter	Value
track_buffer	7
match_thresh	0.9
proximity_thresh	0.3
appearance_thresh	0.6
with_reid	True

3.3.3 Vehicle Tracking. To calculate features to be used by the vehicle activity classification model described in Section 3.4.1, it is important that detected vehicles can also be tracked. This can be achieved through a small addition to the detection model. The Ultralytics library allows a tracking model to be seamlessly integrated into the detection phase. The BoT-SORT-ReID tracker [1] was chosen for this purpose. The tracking model associates detections in different images based on two characteristics: similarity in bounding box location and, if the ReID (Re-Identification) mechanism is enabled, similarity in appearance, determined based on the features extracted by the YOLO detection model. The presence of the ReID mechanism is the reason the BoT-SORT tracker was chosen over its alternatives, such as the Byte-Track algorithm [40]. All tracking algorithms are designed to track objects across video frames, where differences in position between two instances of the same object are much more subtle than in the case of pictures taken at a significant time interval; therefore, object appearance is likely to be a significantly more reliable method of associating detected objects than their position. To better adjust the Bot-SORT algorithm to pictures taken at a significant time interval, instead of video frames, some of its configuration parameters were adjusted. Table 2 shows the values of the modified parameters; any parameters not shown in the table have the default values specified in the *botsort.yaml* file of the Ultralytics library.

3.4 Activity Classification Model

Two different construction vehicle activity classification models are developed and tested. The first uses manually extracted features and a support vector machine (SVM), while the second uses a vision transformer (ViT) backbone — a type of deep learning model — to automatically extract features.

3.4.1 SVM Model. The SVM model relies on four different features to represent the transition between an image and the next image of the same camera and to link that transition to each construction vehicle's activity. For each detection, the extracted features are:

(1) **Bounding box centroid difference**: The euclidean distance between the centroid of a vehicle's bounding box and the centroid of that same vehicle's bounding box in the next image.
(2) **Absolute bounding box area difference**: The absolute value of the difference in area between a vehicle's bounding box and the corresponding bounding box in the next image.
(3) **Bounding box intersection over union (IoU)**: The intersection over union of a vehicle's bounding box and its corresponding bounding box in the next image.

Table 3. Evaluated SVM hyperparameters

Hyper-parameter	Evaluated Values
Kernel	{'linear', 'rbf'}
C	{0.1, 0.2, ... , 10}
Gamma	{0.05, 0.1, 0.2, ... , 2.5}

(4) **Absolute per-pixel difference mean**: The region of the vehicle's bounding box is cropped out of the first image. The region with the same coordinates is cropped out of the second image, regardless of whether the vehicle's bounding box is still determined to be at those coordinates. The per-pixel absolute difference is computed between these cropped regions (see the OpenCV documentation [24] for all implementation details). The mean of all per-pixel differences is used as the feature.

For features 1 and 2, if there is no corresponding bounding box in the second image (which can happen, for instance, if the vehicle leaves the camera's field of view), the feature's value is assumed to be the maximum of all recorded values for that feature, since a movement outside of the frame is likely to be at least as large as any movement inside the camera's frame. In this case, the bounding box IoU (feature 3) is also assumed to be 0.

Features 1 and 2 are chosen because the previous study by Küpers et al. [19] shows they have the potential to accurately distinguish between idle and active construction vehicles. Feature 3 is introduced to better account for small changes in the vehicle's position or orientation (which would likely be common especially for "operating" vehicles), while feature 4 is introduced to account not only for vehicle movement, but also for vehicle appearance, which if the vehicle is operating, may be the only feature that shows any change. Detailed information about the distribution of all features used for the SVM model is available in Appendix E.

The extracted features are standardised using their Z-score to prevent features with larger variance from dominating the model. The standardised features are then reduced to only 2 dimensions by applying principal component analysis (PCA). This step is taken to eliminate noise in the data, prevent multicollinearity from affecting the model and to speed up the model's training time.

The activity-annotated version of the Amsterdam dataset is used for training and evaluation of the model. The training and validation splits are merged for this model's training process. To tune the model's hyperparameters, an exhaustive grid search is performed on the parameter values shown in Table 3.

Each hyperparameter combination is evaluated by performing a 5-fold cross-validation on the merged train and validation splits. The model with the highest cross-validation F1 score is selected and re-evaluated on the test split to produce an unbiased estimation of the model's real-world performance.

3.4.2 Vision Transformer (ViT) Model. Manually extracted features are unlikely to be capable of fully representing the range of possible changes in an image that can be described as a vehicle "operating" or "moving". Therefore, deep learning is a promising solution to accurately classify even these more complex situations. Deep learning

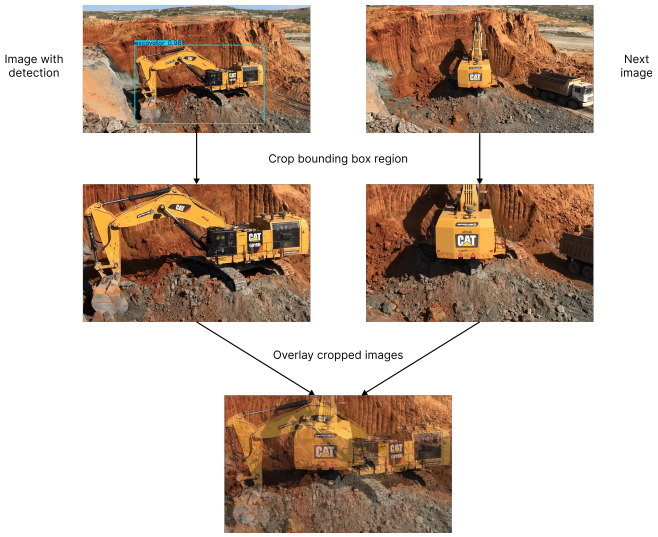

Fig. 2. Image pre-processing for transformer model

image classifiers usually consist of two main parts: a feature extractor (traditionally, a convolutional neural network or, more recently, a vision transformer) and a classification head. Vision Transformers (ViT) are a type of deep learning model that applies the transformer architecture, originally designed for natural language processing, to computer vision tasks. They split images into patches, treat each patch like a token, and process them using self-attention mechanisms to capture global relationships.

The DINOv2 pre-trained ViT [25] is used as the feature extractor in this study because it has shown very good generalisation abilities without additional fine-tuning [25]. Furthermore, it has also been successfully applied to classifying non-natural images, outperforming the well-known convolutional feature extractor ResNet50 in a study examining medical image classification [11]. Features extracted by the DINO backbone will be classified using a single linear layer.

While the vision transformer model does not require any tracking information (unlike the SVM model), the results of the detection model require a different pre-processing step before they can be used as input to the transformer. Figure 2 shows a graphical representation of this process. For any detected construction vehicle, the area of its bounding box is cropped from the picture. The area with the same coordinates is also cropped from the next picture taken by the same camera, regardless of whether the vehicle is still detected at those coordinates. The two cropped areas are overlaid on each other, with each crop being set to 50% transparency, resulting in a non-natural-looking hybrid of the two images, which is what makes DINOv2's performance with non-natural images important.

The linear layer is the only part of the model that has to be trained. Its training is done on the training split of the activity-annotated images of the Amsterdam dataset, pre-processed according to the procedure described above. The Hugging Face Transformers Python library [12] is used for this training process. The validation split is used to check model performance on each epoch and to adjust model hyperparameters. The best-performing hyperparameters are

Table 4. ViT Model Hyperparameters

Hyperparameter	Value
Epochs	100
Image size	512
Batch size	12
Optimiser	AdamW
Learning rate	10^{-5}
Momentum	0.9

recorded in Table 4. After training is complete, the test split is used for the final model evaluation.

3.5 Air Quality Data Analysis

The output of the activity classification model is, for every detected vehicle, what its activity is over each interval of 5 minutes. If there are multiple vehicles detected in the same time interval, either by the same camera or by different cameras, these outputs must first be aggregated. For each 5 minute time interval, a variable A_xC_y (with x∈{0/"moving", 1/"operating", 2/"stationary"} and y∈{0/"cement_truck", 1/"excavator", 2/"mobile_crane", 3/"truck"}) is computed, representing the number of vehicles of type y doing activity x during that time interval summed across all cameras.

To examine the relation between vehicle activities and local air quality, ordinary least squares (OLS) linear regressions are performed using the A_xC_y variables as independent variables and the NO_2, CO_2, O_3, PM_1, $PM_{2.5}$, PM_{10} air quality metrics as dependent variables. Ambient temperature, relative humidity, atmospheric pressure and (trigonometrically transformed) time of day are also introduced as independent variables in the regressions, since they are possible confounds that need to be controlled for in the analysis. Finally, the significance of the impact of each A_xC_y variable on the regression line is quantified using a t-test to determine which construction vehicle activities have a significant relationship with air quality metrics.

Exploratory data analysis has revealed that the air quality metrics time series show strong, long-lasting autocorrelations, likely due to physical factors—pollutant levels cannot change drastically within five minutes, so closely spaced measurements are highly correlated. As a result, regression residuals for all air quality metrics are also autocorrelated (Ljung-Box $p < 0.001$) and heteroscedastic (Breusch-Pagan $p < 0.001$). This violates the assumptions of standard OLS regression and increases the risk of type I errors, especially with large sample sizes such as those used in this study [37]. To address this, we use the Newey-West HAC estimator [23] to calculate robust covariance matrices. The estimator requires selecting a maximum lag parameter (m), which we determine individually for each regression. The value chosen is the number of lags past which the residual autocorrelations become statistically nonsignificant, as shown in their respective ACF plots (all plots available in Appendix F).

Table 5. Detection Model Performance

Metric	First round	Fine-tuned
mAP50	74.8%	95.3%
mAP50-95	60.5%	74.4%
Precision	69.8%	94.9%
Recall	66.7%	82.8%
F1	68.2%	88.4%

4 RESULTS

This section begins by presenting the performance of the proposed detection model. Following that, the performance of the two construction vehicle activity classification models is outlined and compared. Finally, this section shows the results of the linear regression analysis of vehicle activity and air quality data.

4.1 Detection Model Performance

Table 5 presents the performance of the detection model on the Amsterdam dataset test split, both after initial training on the ACID-MOCS dataset and after fine-tuning on the Amsterdam training split. The results show that fine-tuning significantly improved the model's ability to detect construction vehicles in the Amsterdam dataset images, confirming that the process was effective in adapting the model to the target dataset.

The model seems to perform similarly for all vehicle classes (see Appendix G, Figure 19 for per-class precision-recall curves and mAP metrics). The measures taken to address class imbalance appear effective, as the minority classes (cement trucks and mobile cranes) achieved higher mAP scores than the more common excavators and trucks. The lower performance for excavators may stem from frequent occlusion by dirt or other vehicles, while the high intra-class variance of trucks (due to differing cargo and viewing angles) likely contributed to reduced accuracy in their case. The normalised confusion matrix is made available for further analysis in Appendix G, Figure 20.

4.2 Activity Classification Performance

Table 6 shows the performance metrics of both the SVM and the ViT activity classification models on the test split of the Amsterdam dataset. The ViT model outperforms the SVM in all but one of the evaluated metrics. The "operating" class appears to be the most difficult activity class to accurately predict for both models, though the ViT model offers the most significant improvement compared to the SVM in this class, with a 16 p.p. increase in F1. The challenge associated with the "operating" class can be attributed to its considerable intra-class variability. For instance, the operation of an excavator can entail moving its component parts in many different ways, each of which being substantially different from the operation of a cement truck. Additionally, the class exhibits relative semantic ambiguity, as even human observers may find it difficult to determine clear boundaries between a vehicle operating, moving, or remaining stationary. Although these categories are treated as discrete in this study, they appear to constitute a continuum in reality.

Table 6. Activity Classification Models Performance

Metric	SVM Model	ViT Model
Overall Accuracy	82.1%	87.1%
Overall Precision	81.8%	87.2%
Overall Recall	82.1%	87.1%
Overall F1 Score	81.8%	87.0%
"Moving" Precision	73.9%	75.0%
"Moving" Recall	85.7%	84.9%
"Moving" F1 Score	79.4%	79.6%
"Operating" Precision	62.7%	77.6%
"Operating" Recall	52.5%	69.1%
"Operating" F1 Score	57.1%	73.1%
"Stationary" Precision	90.8%	93.8%
"Stationary" Recall	90.8%	93.3%
"Stationary" F1 Score	90.8%	93.5%

4.3 Relation between Vehicle Activity and Air Quality

Full tables of results for the regressions conducted to test the relationship between construction vehicle activity and air quality are available in Appendix H.

Regressions for all six air quality metrics indicate a relatively poor fit, with R^2 values between 0.076 (for NO_2) and 0.484 (for CO_2). However, all regression models nonetheless indicate that the included independent variables are jointly statistically significant predictors of air quality metrics (F-statistic p<0.001), despite the low explanatory power indicated by the R^2 value.

In predicting NO_2, statistically significant predictors are operating (p=0.032; $\beta = 2.72$) and stationary cement trucks (p=0.019; $\beta = 2.58$), air temperature (p=0.039; $\beta = 0.61$), air pressure (p=0.014; $\beta = 0.24$) and cosine-transformed time of day (p=0.003; $\beta = 3.14$).

In predicting O_3, statistically significant variables are relative humidity (p<0.001; $\beta = -0.63$), air pressure (p=0.001; $\beta = -0.41$), and cosine-transformed time of day (p=0.006; $\beta = -3.68$). No vehicle activities have a statistically significant relation with O_3 levels.

In predicting CO_2, statistically significant vehicle activity variables are operating (p=0.008; $\beta = 15.32$) and moving excavators (p=0.012; $\beta = 9.48$), and stationary cement trucks (p=0.037; $\beta = 6.64$). Among environmental variables, air pressure (p=0.000; $\beta = 2.87$) was positively associated with CO_2 levels, while air temperature (p=0.009; $\beta = -3.84$) showed a negative association.

In predicting PM_1 concentrations, significant predictors were the number of operating trucks (p=0.001; $\beta = 6.02$), operating mobile cranes (p=0.001; $\beta = -6.29$), stationary mobile cranes (p=0.001; $\beta = -6.20$), and moving mobile cranes (p=0.006; $\beta = -5.02$). Environmental factors (temperature, relative humidity, or air pressure) and the time of day were not statistically significant in this model.

The significant predictors of $PM_{2.5}$ concentration are the same as those of PM_1 concentrations: operating trucks ($p = 0.001$; $\beta = 8.12$), operating mobile cranes (p=0.002; $\beta = -6.97$), stationary mobile cranes (p=0.001; $\beta = -7.15$), and moving mobile cranes (p=0.006; $\beta = -5.48$). As with PM_1, environmental factors (temperature, relative humidity, and air pressure) and the time of day were not statistically significant.

The pattern seen with smaller particulate matter sizes repeats itself in the case of PM_{10} concentrations. Significant predictors of PM_{10} are, once again, the number of operating trucks ($p = 0.012$; $\beta = 6.33$), operating mobile cranes (p = 0.021; $\beta = -6.91$), stationary mobile cranes (p = 0.010; $\beta = -7.66$), and moving mobile cranes (p = 0.026; $\beta = -5.84$). As with PM_1 and $PM_{2.5}$, environmental factors and the time of day were not statistically significant in this model.

5 DISCUSSION

This section will first provide an interpretation of the key results and discuss their probable causes. Then, it will compare the results obtained in this paper with those of related works introduced in Section 2. Finally, it will examine the limitations of this study's methodology and it will propose directions for future work to address these limitations.

5.1 Interpretation of Key Findings

The performance results of the detection and activity classification models indicate that, under the tested conditions, 5-minute interval imagery is sufficient for reasonably accurate construction vehicle activity classification. The main implication for industry is that existing data and camera configurations can be used directly to analyse vehicle activities on their sites, not only for air quality monitoring, but also for other applications, such as productivity estimation or safety monitoring. An important implication for research is that real-time model inference is not as crucial as previously believed for construction site monitoring, since accurate inferences can be performed even when there is a large time interval between frames. Furthermore, the ViT overlay method of recognising transitions/activity between two images introduced in this study is a promising development for many other low-frame-rate computer vision tasks, since it has been shown to provide largely accurate results on temporally sparse imagery, in a complex environment, with little annotated training data.

Analysis of air quality data shows some link between vehicle activity and pollutant levels, though low R^2 values suggest major influences beyond vehicles and weather. Ozone (O_3) levels are unaffected by vehicle activity, with atmospheric conditions being the highest observed influence. Other pollutants show varying relationships: cement truck presence correlates with higher NO_2, though the cause is unclear. CO_2 levels rise with excavator use, as expected given the previously attested emissions of their diesel engines [17], but also with stationary cement truck presence—likely reflecting overlap with other pollution-heavy activities rather than a direct effect. Particulate matter (PM) levels correlate positively, across all PM sizes, with operating trucks (likely due to the trucks raising dust when being loaded or unloaded), while mobile crane presence consistently shows a negative correlation. This is likely because cranes appear mainly during wall assembly, not during dustier earth-moving phases.

Table 7. Detection Model Comparison

Study	Model Architecture	mAP50
Arabi et al. (2020) [3]	MobileNet	91.2%
Küpers et al. (2025) [19]	YOLOv8	70.4%
Zhang et al. (2022) [41]	Self-supervised YOLOv4	92.9%
Guo et al. (2023) [9]	Anchor-free network	71.0%
Current study	**YOLOv9**	**95.3%**

5.2 Comparison with Related Works

Although not directly comparable due to dataset differences, Table 7 shows that the mAP50 score of the proposed detection model exceeds those reported in the literature discussed in Section 2. Metrics shown in the table are those reported in the original studies. The performance improvement can be explained by several factors. One is the use of the newer and more powerful YOLOv9 architecture. Another is the use of the high-quality and relatively sizeable ACID and MOCS datasets. Additionally, most other models for detecting construction vehicles are optimised for inference on edge devices, whereas the model developed in this study was under no such constraints.

A quantitative comparison of the activity classification model with other works would be meaningless, since each study discussed in Section 2 uses different activity classes and deals with different vehicles. However, a key point of novelty is that the model proposed in the current paper only requires two frames for inference, while being able to distinguish between different types of non-idle activity (moving and operating). A table highlighting key qualitative differences between the current model and other works is available in Appendix G, Table 6.

5.3 Limitations and Future Work

The 3-stage (detection, activity classification, pollution analysis) design of the study is inherently liable to cascading errors. Neither the detection model nor the activity classification models are perfect, and any errors made by a previous pipeline stage are likely to degrade the performance of all stages that follow after it. Furthermore, since the evaluation of the activity classification models and the OLS regressions is performed using data coming from previous stages of the pipeline, the reported performance metrics may also be affected by such cascading errors. Since the extent to which the reported performance metrics are affected by cascading errors is currently unknown, future work could aim to quantify it by isolating activity classification and regression performance evaluation from previous stages through the use of ground truth labels created independently of previous stage results.

The 5-minute time interval between pictures in the Amsterdam dataset limits the precision of the air quality data analysis. A vehicle may perform multiple different activities within 5 minutes, but only one of them can be detected and considered in the analysis. Future work should aim to validate the relations between air quality and vehicle activities described in this paper using video data, which offers higher granularity.

The method of aggregating activity classification results coming from different cameras in the same time interval (by simply summing all results together) is also prone to introducing errors in the analysis. Some vehicles may be visible by multiple cameras (and counted twice), while others may sometimes not be visible by any cameras (and not counted, even though they are still influencing the pollution metrics). Future work should aim to prevent errors caused by camera fields of view by ensuring full camera coverage of the construction site, accurately documenting camera placement, and treating detections in overlapping camera angles separately. Techniques for combining results from overlapping camera angles (by, for instance, using confidence levels or confidence-weighted majority voting) should also be developed and evaluated.

A linear regression analysis was performed between construction vehicle activity and air quality, yet potential non-linear relations have not been investigated. Future work could examine whether vehicle activity levels can serve as useful features in a non-linear model, such as a neural network or a non-linear regression.

Finally, the relationships described in this study between vehicle activity and air quality may only be interpreted as correlational, but not as causational. A causal link may be established in future studies by controlling for any possible external influences on the measured pollution levels (such as the background pollution in the area) and by employing statistical techniques such as Granger causality tests or instrumental variables. Controlling for the level of background pollution was also initially considered for this study by examining the difference between pollution sensors positioned upwind of the construction site and those positioned downwind. However, this was found to be infeasible due to a lack of data on inter-sensor calibration and wind speed and direction, as well as due to a lack of specialised knowledge in atmospheric modelling. This limitation highlights a need for increased multi-disciplinary cooperation and early researcher involvement in data gathering for future studies of this topic.

6 CONCLUSION

This study highlighted that while construction activity is a visible and detectable phenomenon in computer vision pipelines, its contribution to localised air quality variance may be secondary to broader environmental and operational factors. The YOLOv9 architecture was found to be highly capable of reliably identifying and tracking construction vehicles in sparse surveillance camera imagery. A novel approach combining overlaid image crops with a deep neural network powered by a DINOv2 backbone was identified as the superior option for classifying vehicle activities in temporally sparse image pairs. However, most construction vehicle activities were found to have no linear relationship with local air quality metrics; of those activities that were found to be linearly related to local air quality, some revealed surprising and likely non-causal links. Therefore, effective air quality modelling will likely require multi-modal sensing and tighter integration with site-level operational data.

ACKNOWLEDGMENTS

The author wishes to express their gratitude to Rob Bemthuis and Arda Satici for supervising this research project. The author also

wishes to thank Deepak Yeleshetty for his helpful advice on computer vision techniques and Fulya Kula for validating the statistical techniques used in the air quality data analysis. This work was partly supported by the Dutch Ministry of Infrastructure and Water Management and TKI Dinalog under the ECOLOGIC project (case no. 31192090 and 5000006252). The author thanks the project partners for their involvement.

REFERENCES

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. *BoT-SORT: Robust Associations Multi-Pedestrian Tracking*. July 7, 2022. DOI: 10.48550/arXiv.2206.14651. arXiv: 2206.14651[cs]. URL: http://arxiv.org/abs/2206.14651 (visited on 06/13/2025).

[2] Dheeraj Alshetty and S. M. Shiva Nagendra. "Impact of vehicular movement on road dust resuspension and spatiotemporal distribution of particulate matter during construction activities". In: *Atmospheric Pollution Research* 13.1 (Jan. 1, 2022), p. 101256. ISSN: 1309-1042. DOI: 10.1016/j.apr.2021.101256. URL: https://www.sciencedirect.com/science/article/pii/S1309104221003184 (visited on 04/28/2025).

[3] Saeed Arabi, Arya Haghighat, and Anuj Sharma. "A deep-learning-based computer vision solution for construction vehicle detection". In: *Computer-Aided Civil and Infrastructure Engineering* 35.7 (2020), pp. 753–767. ISSN: 1467-8667. DOI: 10.1111/mice.12530. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12530 (visited on 04/28/2025).

[4] Wei Chen, Jinjin Luo, Fan Zhang, and Zijian Tian. "A review of object detection: Datasets, performance evaluation, architecture, applications and current trends". In: *Multimedia Tools and Applications* 83.24 (July 1, 2024), pp. 65603–65661. ISSN: 1573-7721. DOI: 10.1007/s11042-023-17949-4. URL: https://doi.org/10.1007/s11042-023-17949-4 (visited on 04/28/2025).

[5] Andrea Clements, Rachelle Duvall, Danny Greene, and Tim Dye. *The Enhanced Air Sensor Guidebook*. EPA/600/R-22/213. Research Triangle Park, NC: United States Environmental Protection Agency, Office of Research and Development, 2022. URL: https://www.epa.gov/air-sensor-toolbox.

[6] *COCO - Common Objects in Context*. URL: https://cocodataset.org/#format-data (visited on 06/10/2025).

[7] Mark Everingham and John Winn. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit*. May 18, 2012. URL: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/devkit_doc.pdf (visited on 06/15/2025).

[8] Jeremy Fraenkel, Alex Heaton, and Derek Topper. *pylabel-project/pylabel*. original-date: 2021-10-20T03:56:54Z. June 9, 2025. URL: https://github.com/pylabel-project/pylabel (visited on 06/10/2025).

[9] Yapeng Guo, Yang Xu, Jin Niu, and Shunlong Li. "Anchor-free arbitrary-oriented construction vehicle detection with orientation-aware Gaussian heatmap". In: *Computer-Aided Civil and Infrastructure Engineering* 38.7 (2023), pp. 907–919. ISSN: 1467-8667. DOI: 10.1111/mice.12940. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12940 (visited on 04/28/2025).

[10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Apr. 17, 2017. DOI: 10.48550/arXiv.1704.04861. arXiv: 1704.04861[cs]. URL: http://arxiv.org/abs/1704.04861 (visited on 04/28/2025).

[11] Yuning Huang, Jingchen Zou, Lanxi Meng, Xin Yue, Qing Zhao, Jianqiang Li, Changwei Song, Gabriel Jimenez, Shaowu Li, and Guanghui Fu. "Comparative Analysis of ImageNet Pre-Trained Deep Learning Models and DINOv2 in Medical Imaging Classification". In: *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*. 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC). ISSN: 2836-3795. July 2024, pp. 297–305. DOI: 10.1109/COMPSAC61105.2024.00049. URL: https://ieeexplore.ieee.org/abstract/document/10633612 (visited on 06/16/2025).

[12] Hugging Face. *Transformers*. URL: https://huggingface.co/docs/transformers/en/index (visited on 06/16/2025).

[13] Nidhal Jegham, Chan Young Koh, Marwan Abdelatti, and Abdeltawab Hendawi. *Yolo Evolution: A Comprehensive Benchmark and Architectural Review of Yolov12, Yolo11, and Their Previous Versions*. 2025. DOI: 10.2139/ssrn.5175639. URL: https://www.researchgate.net/profile/Nidhal-Jegham-2/publication/389264268_YOLO_Evolution_A_Comprehensive_Benchmark_and_Architectural_Review_of_YOLOv12_YOLO11_and_Their_Previous_Versions/links/67d860b32719090652bd034b/YOLO-Evolution-A-Comprehensive-Benchmark-and-Architectural-Review-of-YOLOv12-YOLO11-and-Their-Previous-Versions.pdf (visited on 04/28/2025).

[14] Wenyang Ji, Lingjun Tang, Dedi Li, Wenming Yang, and Qingmin Liao. "Video-based construction vehicles detection and its application in intelligent monitoring system". In: *CAAI Transactions on Intelligence Technology* 1.2 (Apr. 1, 2016), pp. 162–172. ISSN: 2468-2322. DOI: 10.1016/j.trit.2016.09.001. URL: https://www.sciencedirect.com/science/article/pii/S2468232216300282 (visited on 04/28/2025).

[15] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. *Ultralytics YOLO*. Version 8.0.0. Jan. 2023. URL: https://github.com/ultralytics/ultralytics.

[16] Jovan Kalajdjieski, Eftim Zdravevski, Roberto Corizzo, Petre Lameski, Slobodan Kalajdziski, Ivan Miguel Pires, Nuno M. Garcia, and Vladimir Trajkovik. "Air Pollution Prediction with Multi-Modal Data and Deep Neural Networks". In: *Remote Sensing* 12.24 (Jan. 2020). Number: 24 Publisher: Multidisciplinary Digital Publishing Institute, p. 4142. ISSN: 2072-4292. DOI: 10.3390/rs12244142. URL: https://www.mdpi.com/2072-4292/12/24/4142 (visited on 04/26/2025).

[17] Asmat Ullah Khan and Lizhen Huang. "Toward Zero Emission Construction: A Comparative Life Cycle Impact Assessment of Diesel, Hybrid, and Electric Excavators". In: *Energies* 16.16 (Jan. 2023). Number: 16 Publisher: Multidisciplinary Digital Publishing Institute, p. 6025. ISSN: 1996-1073. DOI: 10.3390/en16166025. URL: https://www.mdpi.com/1996-1073/16/16/6025 (visited on 06/27/2025).

[18] In-Sup Kim, Kamran Latif, Jeonghwan Kim, Abubakar Sharafat, Dong-Eun Lee, and Jongwon Seo. "Vision-Based Activity Classification of Excavators by Bidirectional LSTM". In: *Applied Sciences* 13.1 (Jan. 2023). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, p. 272. ISSN: 2076-3417. DOI: 10.3390/app13010272. URL: https://www.mdpi.com/2076-3417/13/1/272 (visited on 04/28/2025).

[19] Xander Küpers, Jeroen Klein Brinke, Rob Bemthuis, and Ozlem Durmaz Incel. *Towards Edge-Based Idle State Detection in Construction Machinery Using Surveillance Cameras*. June 3, 2025. DOI: 10.48550/arXiv.2506.00904. arXiv: 2506.00904[cs]. URL: http://arxiv.org/abs/2506.00904 (visited on 06/18/2025).

[20] Xiang Li, Chengqi Lv, Wenhai Wang, Gang Li, Lingfeng Yang, and Jian Yang. "Generalized Focal Loss: Towards Efficient Representation Learning for Dense Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.3 (Mar. 2023), pp. 3139–3153. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2022.3180392. URL: https://ieeexplore.ieee.org/abstract/document/9792391 (visited on 06/13/2025).

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. *Microsoft COCO: Common Objects in Context*. Feb. 21, 2015. DOI: 10.48550/arXiv.1405.0312. arXiv: 1405.0312[cs]. URL: http://arxiv.org/abs/1405.0312 (visited on 06/13/2025).

[22] Arideep Mukherjee and Madhoolika Agrawal. "World air particulate matter: sources, distribution and health effects". In: *Environmental Chemistry Letters* 15.2 (June 1, 2017), pp. 283–309. ISSN: 1610-3661. DOI: 10.1007/s10311-017-0611-9. URL: https://doi.org/10.1007/s10311-017-0611-9 (visited on 04/25/2025).

[23] Whitney K. Newey and Kenneth D. West. "A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix". In: *Econometrica* 55.3 (1987). Publisher: [Wiley, Econometric Society], pp. 703–708. ISSN: 0012-9682. DOI: 10.2307/1913610. URL: https://www.jstor.org/stable/1913610 (visited on 06/12/2025).

[24] *OpenCV: Operations on arrays*. URL: https://docs.opencv.org/4.11.0/d2/de8/group__core__array.html#ga6fef31bc8c4071cbc114a758a2b79c14 (visited on 06/15/2025).

[25] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. Feb. 2, 2024. DOI: 10.48550/arXiv.2304.07193. arXiv: 2304.07193[cs]. URL: http://arxiv.org/abs/2304.07193 (visited on 06/08/2025).

[26] *Roboflow: Computer vision tools for developers and enterprises*. URL: https://roboflow.com (visited on 06/10/2025).

[27] Behnam Sherafat, Changbum R. Ahn, Reza Akhavian, Amir H. Behzadan, Mani Golparvar-Fard, Hyunsoo Kim, Yong-Cheol Lee, Abbas Rashidi, and Ehsan Rezazadeh Azar. "Automated Methods for Activity Recognition of Construction Workers and Equipment: State-of-the-Art Review". In: *Journal of Construction Engineering and Management* 146.6 (June 1, 2020). Publisher: American Society of Civil Engineers, p. 03120002. ISSN: 1943-7862. DOI: 10.1061/(ASCE)CO.1943-7862.0001843. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29CO.1943-7862.0001843 (visited on 06/28/2025).

[28] Trevor Slaton, Carlos Hernandez, and Reza Akhavian. "Construction activity recognition with convolutional recurrent networks". In: *Automation in Construction* 113 (May 1, 2020), p. 103138. ISSN: 0926-5805. DOI: 10.1016/j.autcon.2020.103138. URL: https://www.sciencedirect.com/science/article/pii/S0926580519310234 (visited on 04/28/2025).

[29] Kyung-Suk Suh, Byung-Il Min, Byung-Mo Yang, Sora Kim, Kihyun Park, and Jiyoon Kim. "Machine learning method using camera image patterns for predictions of particulate matter concentrations". In: *Atmospheric Pollution Research* 13.3 (Mar. 1, 2022), p. 101325. ISSN: 1309-1042. DOI: 10.1016/j.apr.2022.101325. URL: https://www.sciencedirect.com/science/article/pii/S1309104222000125 (visited on 04/24/2025).

[30] Yunjie Tian, Qixiang Ye, and David Doermann. *YOLOv12: Attention-Centric Real-Time Object Detectors*. Feb. 18, 2025. DOI: 10.48550/arXiv.2502.12524. arXiv: 2502.12524[cs]. URL: http://arxiv.org/abs/2502.12524 (visited on 06/13/2025).

[31] Rejin Varghese and Sambath M. "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness". In: *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). Apr. 2024, pp. 1–6. DOI: 10.1109/ADICS58448.2024.10533619. URL: https://ieeexplore.ieee.org/document/10533619/citations (visited on 06/13/2025).

[32] Bao Zhen Wang, Zhen Hua Zhu, Ende Yang, Zhi Chen, and Xiang Hong Wang. "Assessment and management of air emissions and environmental impacts from the construction industry". In: *Journal of Environmental Planning and Management* 61.14 (Dec. 6, 2018), pp. 2421–2444. ISSN: 0964-0568. DOI: 10.1080/09640568.2017.1399110. URL: https://doi.org/10.1080/09640568.2017.1399110 (visited on 04/25/2025).

[33] Chien-Yao Wang, I.-Hau Yeh, and Hong-Yuan Mark Liao. *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*. Feb. 29, 2024. DOI: 10.48550/arXiv.2402.13616. arXiv: 2402.13616[cs]. URL: http://arxiv.org/abs/2402.13616 (visited on 06/13/2025).

[34] World Health Organization. *Overview of methods to assess population exposure to ambient air pollution*. 2023. ISBN: 978-92-4-007349-4. URL: https://iris.who.int/bitstream/handle/10665/373014/9789240073494-eng.pdf?sequence=1.

[35] Bo Xiao and Shih-Chung Kang. "Development of an Image Data Set of Construction Machines for Deep Learning Object Detection". In: *Journal of Computing in Civil Engineering* 35.2 (Mar. 1, 2021). Publisher: American Society of Civil Engineers, p. 05020005. ISSN: 1943-5487. DOI: 10.1061/(ASCE)CP.1943-5487.0000945. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000945 (visited on 06/10/2025).

[36] An Xuehui, Zhou Li, Liu Zuguang, Wang Chengzhi, Li Pengfei, and Li Zhiwei. "Dataset and benchmark for detecting moving objects in construction sites". In: *Automation in Construction* 122 (Feb. 1, 2021), p. 103482. ISSN: 0926-5805. DOI: 10.1016/j.autcon.2020.103482. URL: https://www.sciencedirect.com/science/article/pii/S0926580520310621 (visited on 06/10/2025).

[37] Kun Yang, Justin Tu, and Tian Chen. "Homoscedasticity: an overlooked critical assumption for linear regression". In: *General Psychiatry* 32.5 (Oct. 17, 2019), e100148. ISSN: 2517-729X. DOI: 10.1136/gpsych-2019-100148. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6802968/ (visited on 06/16/2025).

[38] Chao Zhang, Junchi Yan, Changsheng Li, Xiaoguang Rui, Liang Liu, and Rongfang Bie. "On Estimating Air Pollution from Photos Using Convolutional Neural Network". In: *Proceedings of the 24th ACM international conference on Multimedia*. MM '16. New York, NY, USA: Association for Computing Machinery, Oct. 1, 2016, pp. 297–301. ISBN: 978-1-4503-3603-1. DOI: 10.1145/2964284.2967230. URL: https://dl.acm.org/doi/10.1145/2964284.2967230 (visited on 04/26/2025).

[39] Ruichuan Zhang, Bing Chen, and Jenna Krall. "Predictive Learning for Air Pollution at Construction Sites Using Multimodal Data". In: 2024 ASCE International Conference on Computing in Civil Engineering. Pittsburgh, Pennsylvania, USA, July 2024. URL: https://www.researchgate.net/publication/388425735_Predictive_Learning_for_Air_Pollution_at_Construction_Sites_Using_Multimodal_Data (visited on 04/26/2025).

[40] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. Apr. 7, 2022. DOI: 10.48550/arXiv.2110.06864. arXiv: 2110.06864[cs]. URL: http://arxiv.org/abs/2110.06864 (visited on 06/21/2025).

[41] Ying Zhang, Xuyang Hou, and Xuhang Hou. "Combining Self-Supervised Learning and Yolo v4 Network for Construction Vehicle Detection". In: *Mobile Information Systems* 2022.1 (Jan. 1, 2022). Publisher: John Wiley & Sons, Ltd, p. 9056415. ISSN: 1574-017X. DOI: 10.1155/2022/9056415. URL: https://doi-org.ezproxy2.utwente.nl/10.1155/2022/9056415 (visited on 04/28/2025).

APPENDICES

A AI USAGE STATEMENT

During the preparation of this work, the author used ChatGPT to help with debugging code, LaTeX syntax, transcribing tables from Python and brainstorming. The author also used the PyCharm IDE, specifically its full-line AI code assist feature, to help with writing code faster. Grammarly was also used to check for any grammar or spelling mistakes in the paper. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

B OPEN CODE

All Python code used to create the results presented in this paper is available using the following link: https://gitlab.utwente.nl/s2913879/bachelor-open-code/

C IMAGE DATASETS CLASS DISTRIBUTIONS

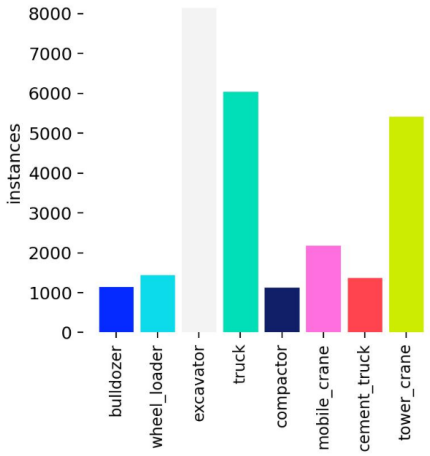

Fig. 3. Class distribution of combined ACID-MOCS dataset

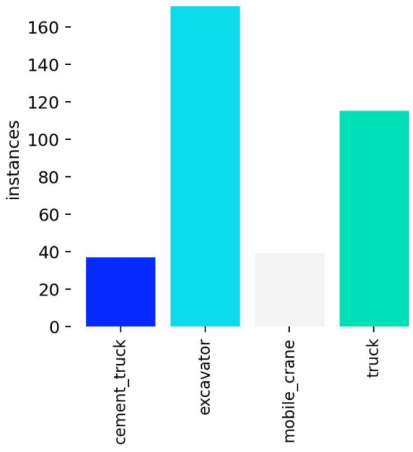

Fig. 4. Class distribution of the Amsterdam bounding-box annotations

D AMSTERDAM DATASET ANONYMISATION

Fig. 5. Example of anonymised picture (source of original image: https://www.greenwheels.nl/en-us/rent-a-car/amsterdam)

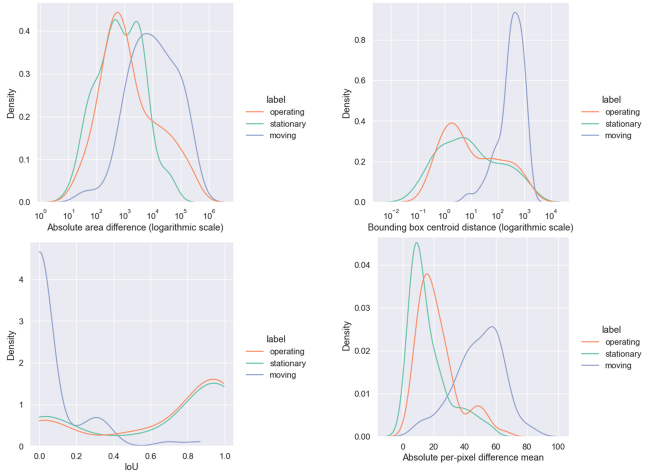

Fig. 7. KDE feature distribution plots for cement trucks

E SVM FEATURES DISTRIBUTION KERNEL DENSITY ESTIMATES (KDE)

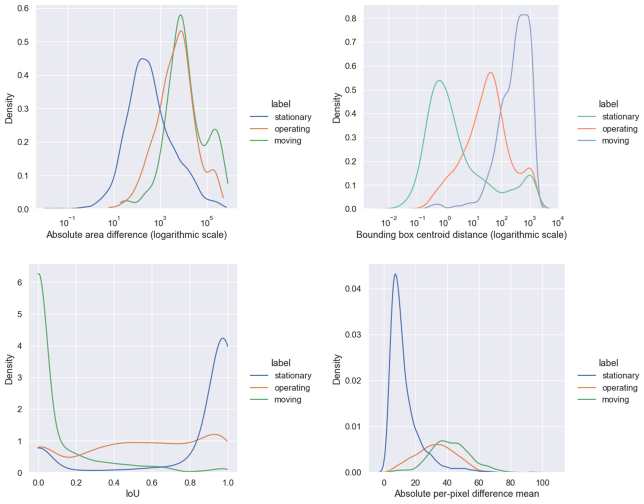

Fig. 6. Global KDE feature distribution plots

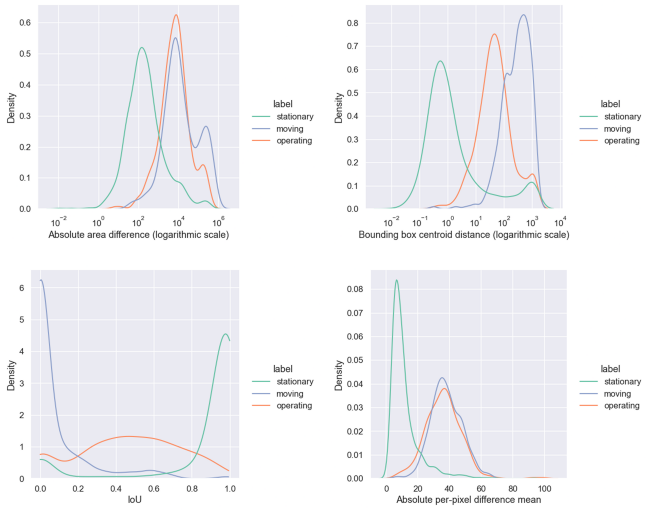

Fig. 8. KDE feature distribution plots for excavators

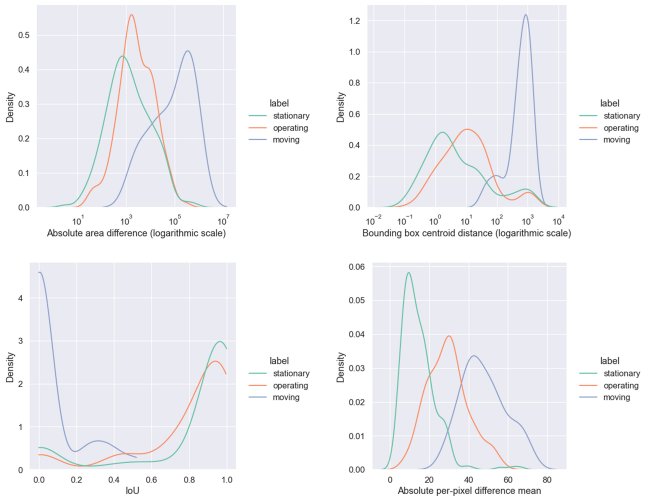

Fig. 9. KDE feature distribution plots for excavators

F ACF PLOTS OF THE REGRESSION RESIDUALS

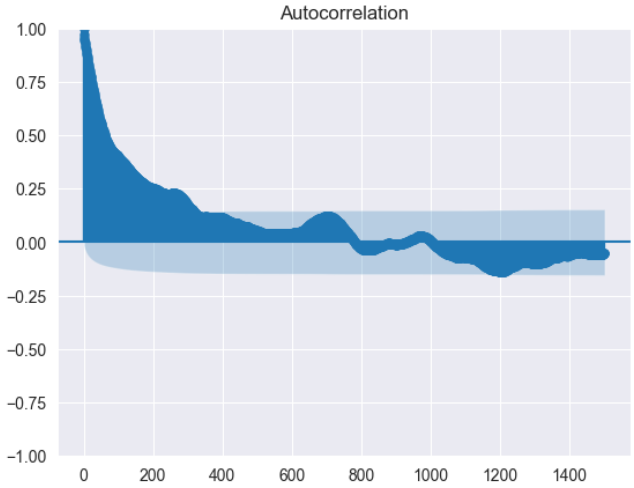

Fig. 11. NO2 Regression Residuals ACF Plot

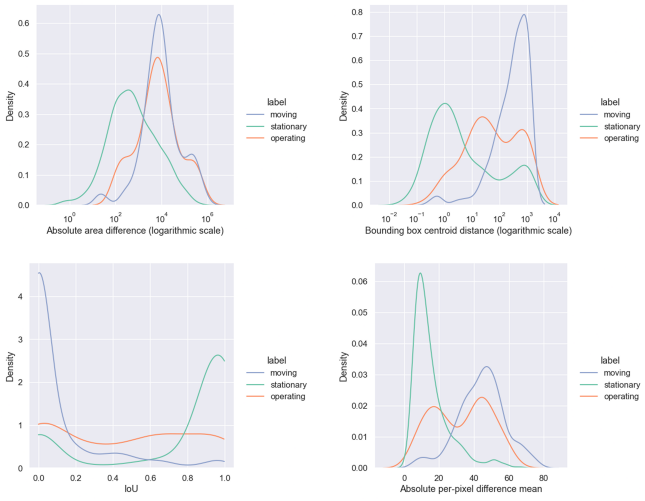

Fig. 10. KDE feature distribution plots for trucks

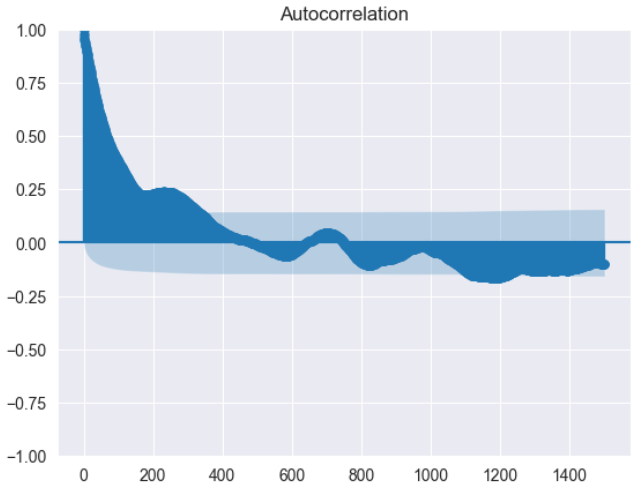

Fig. 12. O3 Regression Residuals ACF Plot

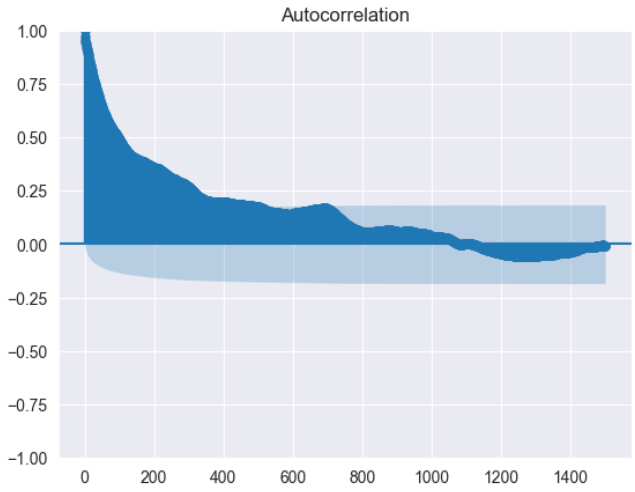

Fig. 13. CO2 Regression Residuals ACF Plot

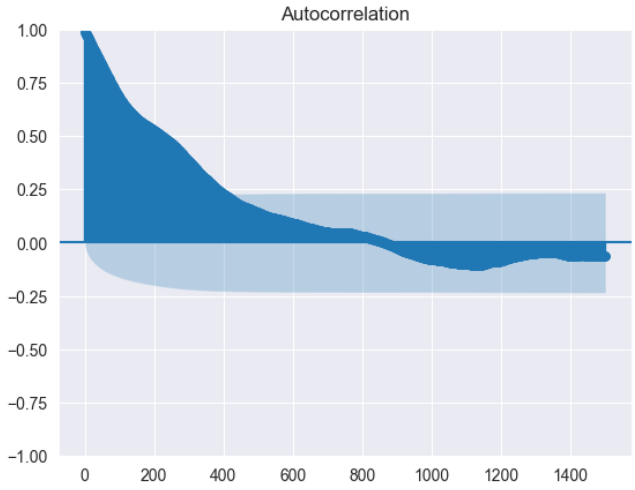

Fig. 14. PM1 Regression Residuals ACF Plot

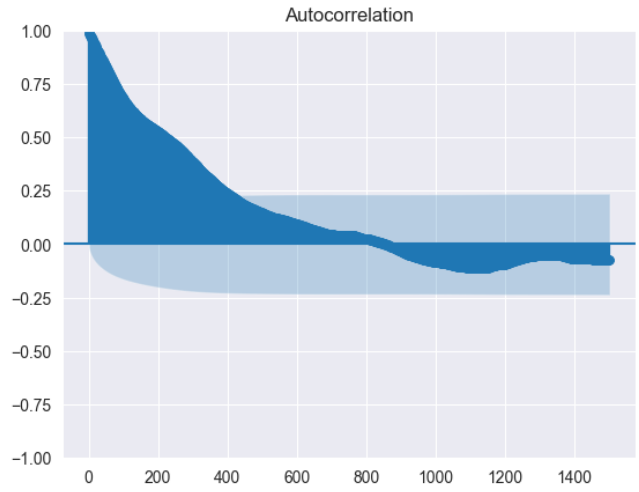

Fig. 15. PM2P5 Regression Residuals ACF Plot

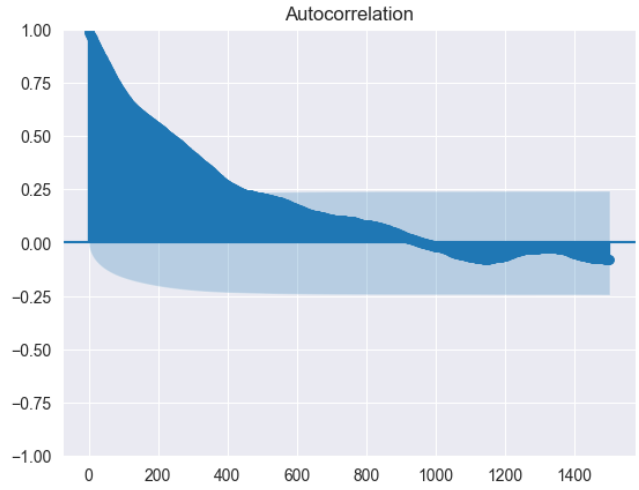

Fig. 16. PM10 Regression Residuals ACF Plot

G DETECTION MODEL PERFORMANCE

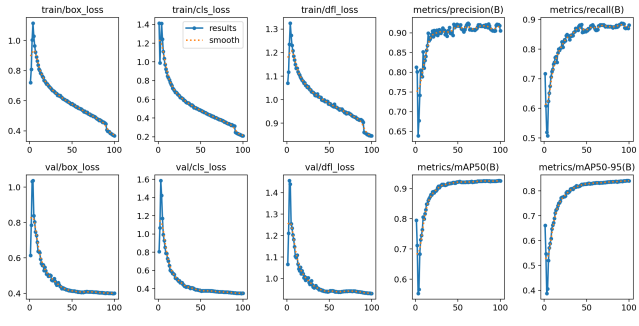

Fig. 17. Detection model performance during training on the ACID-MOCS dataset

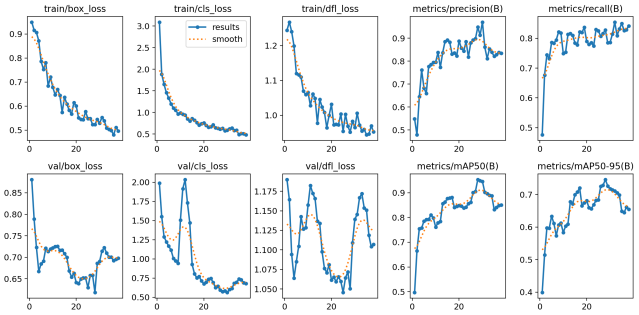

Fig. 18. Detection model performance during fine-tuning on Amsterdam dataset

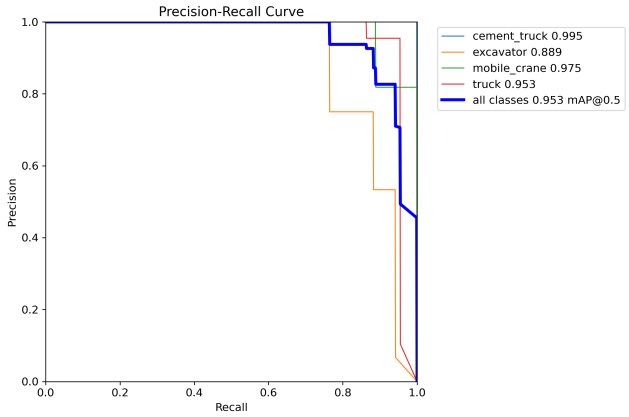

Fig. 19. Precision-recall curve of the fine-tuned detection model

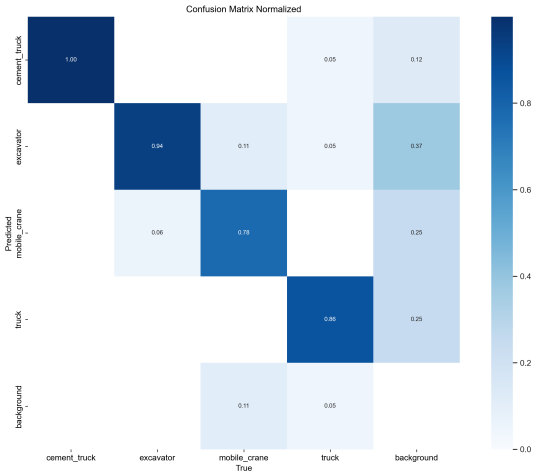

Fig. 20. Detection model normalised confusion matrix

H AIR QUALITY REGRESSION RESULTS

Table 8. Activity Classification Model Comparison

Study	Model	Vehicle Types	Activities	Frames Required	Real-time
Küpers et al. (2020) [19]	Logistic regression on bounding-box features	Excavators, Dump trucks, Concrete mixers	Idle/Non-idle	15	Yes
Kim et al. (2023) [18]	CNN feature extractor + Bi-LSTM	Excavators	Dumping, Excavation, Hauling, Swing	400	No
Current study	DINOv2 ViT on overlaid image crops	Concrete mixers, Excavators, Trucks, Mobile Cranes	Stationary, Operating, Moving	2	No

Table 9. NO2 Regression Results

Dep. Variable:	NO2	**R-squared:**	0.076
Model:	OLS	**Adj. R-squared:**	0.076
Method:	Least Squares	**F-statistic:**	3.660
Date:	Thu, 19 Jun 2025	**Prob (F-statistic):**	4.62e-07
Time:	12:50:22	**Log-Likelihood:**	-79783.
No. Observations:	19973	**AIC:**	1.596e+05
Df Residuals:	19955	**BIC:**	1.597e+05
Df Model:	17	**Max. lags (m):**	400
Covariance Type:	HAC		

	coef	std err	t	P> |t|	[0.025	0.975]
Intercept	-242.6032	100.796	-2.407	**0.016***	-440.172	-45.034
a_1_c_0	2.7222	1.268	2.146	**0.032***	0.236	5.208
a_1_c_1	1.7894	1.057	1.693	0.090	-0.282	3.861
a_1_c_3	1.3934	1.179	1.182	0.237	-0.918	3.704
a_1_c_2	-0.2883	2.075	-0.139	0.890	-4.356	3.780
a_2_c_0	2.5754	1.101	2.339	**0.019***	0.417	4.734
a_2_c_1	-0.7620	0.647	-1.177	0.239	-2.031	0.507
a_2_c_3	0.2954	0.651	0.454	0.650	-0.980	1.571
a_2_c_2	-0.2558	1.524	-0.168	0.867	-3.242	2.731
a_0_c_0	2.2043	1.611	1.368	0.171	-0.954	5.362
a_0_c_1	1.4997	0.900	1.666	0.096	-0.264	3.264
a_0_c_3	0.8239	1.069	0.770	0.441	-1.272	2.920
a_0_c_2	-1.1451	1.703	-0.672	0.501	-4.483	2.193
temperature	0.6138	0.297	2.068	**0.039***	0.032	1.196
relativeHumidity	0.1435	0.078	1.851	0.064	-0.008	0.295
pressure	0.2386	0.097	2.448	**0.014***	0.048	0.430
sin_time_of_day	-0.7800	1.071	-0.729	0.466	-2.878	1.318
cos_time_of_day	3.1405	1.066	2.946	**0.003***	1.051	5.230

Omnibus:	1938.696	**Durbin-Watson:**	0.054
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2546.114
Skew:	0.851	**Prob(JB):**	0.00
Kurtosis:	3.401	**Cond. No.**	1.01e+05

Note: * p-value significant at 95% confidence level

Table 10. O3 Regression Results

Dep. Variable:	O3	**R-squared:**	0.400
Model:	OLS	**Adj. R-squared:**	0.400
Method:	Least Squares	**F-statistic:**	20.10
Date:	Thu, 19 Jun 2025	**Prob (F-statistic):**	1.01e-61
Time:	13:22:33	**Log-Likelihood:**	-82335.
No. Observations:	19973	**AIC:**	1.647e+05
Df Residuals:	19955	**BIC:**	1.648e+05
Df Model:	17	**Max. lags (m):**	350
Covariance Type:	HAC		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	501.4584	128.681	3.897	**0.000**	249.234	753.683
a_1_c_0	-3.3008	2.157	-1.530	0.126	-7.529	0.927
a_1_c_1	-1.6188	1.172	-1.381	0.167	-3.916	0.678
a_1_c_3	-1.3463	1.307	-1.030	0.303	-3.908	1.215
a_1_c_2	1.9017	2.036	0.934	0.350	-2.089	5.893
a_2_c_0	-2.2736	1.207	-1.884	0.060	-4.640	0.092
a_2_c_1	-0.1585	0.902	-0.176	0.860	-1.926	1.609
a_2_c_3	-0.4119	0.757	-0.544	0.586	-1.896	1.072
a_2_c_2	1.7100	1.477	1.157	0.247	-1.186	4.606
a_0_c_0	-1.7210	1.727	-0.997	0.319	-5.105	1.663
a_0_c_1	-1.4092	1.127	-1.250	0.211	-3.619	0.800
a_0_c_3	-0.2768	1.218	-0.227	0.820	-2.663	2.110
a_0_c_2	2.3159	2.200	1.053	0.293	-1.997	6.629
temperature	0.1054	0.329	0.321	0.748	-0.539	0.749
relativeHumidity	-0.6272	0.090	-6.950	**0.000***	-0.804	-0.450
pressure	-0.4146	0.124	-3.335	**0.001***	-0.658	-0.171
sin_time_of_day	-2.3716	1.266	-1.873	0.061	-4.853	0.110
cos_time_of_day	-3.6795	1.338	-2.751	**0.006***	-6.302	-1.057

Omnibus:	311.017	**Durbin-Watson:**	0.049
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	183.980
Skew:	-0.052	**Prob(JB):**	1.12e-40
Kurtosis:	2.542	**Cond. No.**	1.01e+05

Note: * p-value significant at 95% confidence level

Table 11. CO2 Regression Results

Dep. Variable:	CO2	**R-squared:**	0.484
Model:	OLS	**Adj. R-squared:**	0.484
Method:	Least Squares	**F-statistic:**	11.48
Date:	Thu, 19 Jun 2025	**Prob (F-statistic):**	3.96e-32
Time:	13:26:35	**Log-Likelihood:**	-1.0472e+05
No. Observations:	19973	**AIC:**	2.095e+05
Df Residuals:	19955	**BIC:**	2.096e+05
Df Model:	17	**Max. lags (m):**	750
Covariance Type:	HAC		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-1823.9422	387.977	-4.701	**0.000***	-2584.409	-1063.476
a_1_c_0	7.4094	3.913	1.893	0.058	-0.261	15.080
a_1_c_1	15.3200	5.820	2.632	**0.008***	3.913	26.727
a_1_c_3	7.6626	4.575	1.675	0.094	-1.305	16.630
a_1_c_2	-8.4943	5.356	-1.586	0.113	-18.993	2.004
a_2_c_0	6.6421	3.186	2.085	**0.037***	0.397	12.887
a_2_c_1	1.7357	2.633	0.659	0.510	-3.425	6.896
a_2_c_3	-2.5306	2.519	-1.005	0.315	-7.467	2.406
a_2_c_2	-5.5149	4.515	-1.221	0.222	-14.365	3.335
a_0_c_0	3.1876	4.408	0.723	0.470	-5.453	11.828
a_0_c_1	9.4812	3.764	2.519	**0.012***	2.104	16.859
a_0_c_3	1.8572	3.862	0.481	0.631	-5.713	9.427
a_0_c_2	-8.7555	5.523	-1.585	0.113	-19.582	2.071
temperature	-3.8372	1.473	-2.605	**0.009***	-6.724	-0.950
relativeHumidity	0.2502	0.341	0.734	0.463	-0.418	0.918
pressure	2.8719	0.371	7.739	**0.000***	2.145	3.599
sin_time_of_day	2.3384	4.196	0.557	0.577	-5.886	10.563
cos_time_of_day	3.6062	3.149	1.145	0.252	-2.567	9.779

Omnibus:	2551.893	**Durbin-Watson:**	0.055
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4019.333
Skew:	0.905	**Prob(JB):**	0.00
Kurtosis:	4.246	**Cond. No.**	1.01e+05

Note: * p-value significant at 95% confidence level

Table 12. PM1 Regression Results

Dep. Variable:	PM1	**R-squared:**	0.102
Model:	OLS	**Adj. R-squared:**	0.102
Method:	Least Squares	**F-statistic:**	3.396
Date:	Thu, 19 Jun 2025	**Prob (F-statistic):**	2.53e-06
Time:	13:31:54	**Log-Likelihood:**	-85425.
No. Observations:	19973	**AIC:**	1.709e+05
Df Residuals:	19955	**BIC:**	1.710e+05
Df Model:	17	**Max. lags (m):**	425
Covariance Type:	HAC		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-263.2162	195.428	-1.347	0.178	-646.271	119.839
a_1_c_0	-0.0452	1.441	-0.031	0.975	-2.869	2.779
a_1_c_1	2.1435	1.643	1.305	0.192	-1.077	5.364
a_1_c_3	6.0193	1.868	3.222	**0.001***	2.358	9.681
a_1_c_2	-6.2899	1.940	-3.242	**0.001***	-10.092	-2.488
a_2_c_0	0.0183	0.908	0.020	0.984	-1.762	1.798
a_2_c_1	1.2942	1.426	0.908	0.364	-1.501	4.089
a_2_c_3	1.6919	1.345	1.258	0.208	-0.944	4.328
a_2_c_2	-6.1976	1.856	-3.338	**0.001***	-9.836	-2.559
a_0_c_0	-1.4145	1.470	-0.963	0.336	-4.295	1.466
a_0_c_1	-0.7023	1.423	-0.494	0.622	-3.491	2.086
a_0_c_3	3.1106	1.822	1.707	0.088	-0.460	6.682
a_0_c_2	-5.0238	1.820	-2.761	**0.006***	-8.591	-1.457
temperature	-0.3386	0.416	-0.814	0.416	-1.154	0.477
relativeHumidity	0.0991	0.120	0.825	0.409	-0.136	0.335
pressure	0.2754	0.188	1.467	0.142	-0.093	0.643
sin_time_of_day	1.8811	1.274	1.476	0.140	-0.617	4.379
cos_time_of_day	-0.1435	1.377	-0.104	0.917	-2.843	2.556

Omnibus:	1967.327	**Durbin-Watson:**	0.019
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2602.142
Skew:	0.877	**Prob(JB):**	0.00
Kurtosis:	3.226	**Cond. No.**	1.01e+05

Note: * p-value significant at 95% confidence level

Table 13. PM2P5 Regression Results

Dep. Variable:	PM2P5	**R-squared:**	0.121
Model:	OLS	**Adj. R-squared:**	0.121
Method:	Least Squares	**F-statistic:**	3.306
Date:	Thu, 19 Jun 2025	**Prob (F-statistic):**	4.49e-06
Time:	13:34:44	**Log-Likelihood:**	-88245.
No. Observations:	19973	**AIC:**	1.765e+05
Df Residuals:	19955	**BIC:**	1.767e+05
Df Model:	17	**Max. lags (m):**	425
Covariance Type:	HAC		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-297.3664	223.286	-1.332	0.183	-735.026	140.293
a_1_c_0	-0.5369	1.627	-0.330	0.741	-3.726	2.652
a_1_c_1	2.7909	1.961	1.423	0.155	-1.052	6.634
a_1_c_3	8.1241	2.479	3.277	**0.001***	3.265	12.983
a_1_c_2	-6.9711	2.258	-3.088	**0.002***	-11.397	-2.546
a_2_c_0	-0.2377	0.998	-0.238	0.812	-2.195	1.719
a_2_c_1	1.4779	1.613	0.916	0.359	-1.683	4.639
a_2_c_3	2.0568	1.518	1.355	0.175	-0.919	5.032
a_2_c_2	-7.1524	2.114	-3.383	**0.001***	-11.297	-3.008
a_0_c_0	-1.7283	1.600	-1.080	0.280	-4.865	1.409
a_0_c_1	-0.4251	1.674	-0.254	0.799	-3.705	2.855
a_0_c_3	4.4185	2.301	1.920	0.055	-0.092	8.929
a_0_c_2	-5.4824	1.990	-2.756	**0.006***	-9.382	-1.583
temperature	-0.5198	0.496	-1.048	0.295	-1.492	0.453
relativeHumidity	0.1560	0.136	1.147	0.251	-0.111	0.423
pressure	0.3093	0.215	1.442	0.149	-0.111	0.730
sin_time_of_day	1.8053	1.496	1.207	0.228	-1.127	4.737
cos_time_of_day	-0.4930	1.565	-0.315	0.753	-3.561	2.575

Omnibus:	1866.334	**Durbin-Watson:**	0.021
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2445.476
Skew:	0.856	**Prob(JB):**	0.00
Kurtosis:	3.084	**Cond. No.**	1.01e+05

Note: * p-value significant at 95% confidence level

Table 14. PM10 Regression Results

Dep. Variable:	PM10	**R-squared:**	0.091
Model:	OLS	**Adj. R-squared:**	0.090
Method:	Least Squares	**F-statistic:**	3.356
Date:	Thu, 19 Jun 2025	**Prob (F-statistic):**	3.26e-06
Time:	13:37:53	**Log-Likelihood:**	-88312.
No. Observations:	19973	**AIC:**	1.767e+05
Df Residuals:	19955	**BIC:**	1.768e+05
Df Model:	17	**Max. lags (m):**	500
Covariance Type:	HAC		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-253.6808	235.257	-1.078	0.281	-714.805	207.443
a_1_c_0	-0.2270	1.689	-0.134	0.893	-3.537	3.083
a_1_c_1	3.3164	2.068	1.604	0.109	-0.736	7.369
a_1_c_3	6.3333	2.533	2.500	**0.012***	1.369	11.298
a_1_c_2	-6.9112	2.999	-2.305	**0.021***	-12.789	-1.033
a_2_c_0	0.9177	1.419	0.647	0.518	-1.863	3.698
a_2_c_1	2.5069	1.733	1.446	0.148	-0.890	5.904
a_2_c_3	1.3379	1.477	0.906	0.365	-1.558	4.234
a_2_c_2	-7.6565	2.962	-2.584	**0.010***	-13.463	-1.850
a_0_c_0	-0.9193	1.995	-0.461	0.645	-4.830	2.992
a_0_c_1	0.7850	1.748	0.449	0.653	-2.642	4.212
a_0_c_3	3.5512	2.138	1.661	0.097	-0.640	7.743
a_0_c_2	-5.8369	2.630	-2.220	**0.026***	-10.991	-0.683
temperature	-0.2491	0.605	-0.411	0.681	-1.436	0.938
relativeHumidity	-0.3273	0.185	-1.770	0.077	-0.690	0.035
pressure	0.3093	0.225	1.374	0.170	-0.132	0.751
sin_time_of_day	2.0389	1.531	1.332	0.183	-0.963	5.040
cos_time_of_day	0.8364	1.727	0.484	0.628	-2.549	4.222

Omnibus:	1000.379	**Durbin-Watson:**	0.022
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	973.134
Skew:	0.494	**Prob(JB):**	4.86e-212
Kurtosis:	2.560	**Cond. No.**	1.01e+05

Note: * p-value significant at 95% confidence level