Method for Automated Monitoring of Moving Objects using Computer Vision Models[*]

Olena Yashyna[1,*,†], Tetiana Boiarska[2,*,†], Eduard Khomiak[3,*,†], Olena Havrylenko[4,*,†] and Pavlo Hrynevych[5*,†]

National Aerospace University "Kharkiv Aviation Institute", Vadyma Manka St., 17, 61070 Kharkiv, Ukraine

Abstract

The article is devoted to the research and development of methods for monitoring moving objects using computer vision and machine learning. Computer vision and artificial intelligence technologies are increasingly used in road traffic monitoring. At the same time, the implementation of computer vision and machine learning technologies encounters a number of problems associated with insufficient video image quality, unstable weather conditions and high requirements for computing power. This paper proposes a comprehensive method for applying the Viola-Jones algorithm, convolutional neural network and Kalman filter for detecting and tracking moving objects under various lighting conditions.

Keywords

computer vision, moving objects detection, YOLOv5, Viola-Jones algorithm, Kalman filter

1. Introduction

Monitoring of moving objects is becoming especially important in the context of rapid technological development and population growth in cities. The introduction of technologies for monitoring moving objects contributes to the creation of "smart cities", where automated video surveillance systems, traffic management and other intelligent solutions can significantly improve the quality of life. The development of software for monitoring moving objects has significant potential in the areas of road safety. In addition, such systems are able to optimize traffic management, reducing congestion and improving road safety.

At the same time, the automation of traffic monitoring has a number of problems: unstable lighting conditions, dispersion and degradation of the characteristics of video surveillance equipment, difficult weather conditions, etc. In addition, distributed computing systems are often built on budget equipment and have limited computing resources. Thus, the relevance of the work lies in the need to develop new solutions in the field of monitoring moving objects that meet modern safety and efficiency requirements.

The aim of this article is automation of the process of moving objects monitoring using methods and models of computer vision and machine learning.

2. The problem of moving objects monitoring

Monitoring of moving objects is one of the most important topics in modern research, covering a wide range of applications in various areas of life. With the development of computer vision, machine learning and artificial intelligence technologies, monitoring capabilities have been

[*] *ProfIT AI'25: 5th International Workshop of IT-professionals on Artificial Intelligence, October 15–17, 2025, Liverpool, UK*

[*] Corresponding author.

[†] These authors contributed equally.

o.yashina@khai.edu (O. Yashyna); t.m.boiarska@student.khai.edu (T. Boiarska); eakhomiak@gmail.com (E. Khomiak); o.havrylenko@khai.edu (O. Havrylenko); p.s.hrynevych@khai.edu (P. Hrynevych)

0000-0003-2459-1151 (O. Yashyna); 0009-0004-9026-5076 (T. Boiarska); 0000-0002-2579-2986 (E. Khomiak); 0000-0001-5227-9742 (O. Havrylenko); 0009-0003-7870-3216 (P. Hrynevych)

significantly expanded. Today, monitoring of moving objects is used in security systems, transportation systems, as well as in smart cities to improve the quality of life of the population.

Moving objects are physical objects that change their position in space. They can be cars, pedestrians, animals, or any other moving objects. It is important to note that for effective monitoring, it is necessary to be able not only to detect these objects, but also to identify them and track their movement.

Recent advances in big data, artificial intelligence, and the Internet of Things have opened up enormous opportunities and potential to solve traffic management problems. Cameras, WSN, and VANET technologies are common data sources used in smart cities. By using them at intersections, we can collect various traffic data in real time. AI-based approaches play a promising role in minimizing the challenges of effective traffic management [1].

Some of the available methods use lidar, radar, and computer vision. Since a camera is cheaper than radar or lidar, machine vision-based vehicle detection and classification systems have become more popular than lidar or radar-based detection systems. Despite the significant increase in computing power, vehicle detection and classification is not an easy task. The problem lies in the dynamic road environment. The road condition is unpredictable. There may be many artificial infrastructure objects on it, including pedestrians, which complicates the task. In addition, there are background and lighting changes, occlusion, and vehicle heterogeneity. Dense traffic also leads to object overlaps [2].

In general, moving object monitoring involves the integration of various technologies and methods to ensure the accuracy and efficiency of detection, recognition, and tracking.

3. Analysis of computer vision technologies

The main technologies used for monitoring are convolutional neural networks (CNN) and machine learning algorithms. Convolutional neural networks are a powerful tool for image processing because they are able to automatically detect and learn from object features, making them ideal for pattern recognition tasks. This approach allows the model to learn from large data sets, which significantly improves recognition accuracy.

The CNN has superior features for autonomous learning and expression, and feature extraction from original input data can be realized by means of training CNN models that match practical applications. Due to the rapid progress in deep learning technology, the structure of CNN is becoming more and more complex and diverse.

With the development of network architectures, neural network models tend to be deeper, wider, and more complex. Although this evolution can facilitate the networks to capture better feature representations, there is no guarantee that it can operate efficiently in all cases. Models still suffer from disadvantages such as the fact that the networks are more likely to fall into overfitting, and instead of decreasing, the error rate of the training set increases as the networks become deeper and more complex.

As computer vision tasks become increasingly complex, there is a pressing need for CNN models and algorithms that offer higher performance and efficiency [3].

Machine learning algorithms also play a key role in computer vision. They can be used to classify objects based on their characteristics, as well as to predict the behavior of objects over time. The most common machine learning algorithms include support vector machines (SVMs), decision trees, and random forests [4, 5].

Images captured by traffic cameras are often of poor quality due to noise and unstable lighting. They are pre-processed to improve their quality and facilitate object detection. Common pre-processing methods include the following. Normalization adjusts the pixel intensity values of an image to a specific range to ensure consistent input for neural networks. Noise reduction aims to smooth out the image to remove random variations in intensity. Gaussian Blur is a fundamental image pre-processing technique that employs a Gaussian kernel to smooth images, significantly reducing high-frequency noise. This process is crucial for preparing images for more detailed analysis by neural networks. Median Filtering is a non-linear process used in image processing to reduce noise. It replaces each pixel's value with the median value of the intensities found in its immediate neighborhood. Due to its nature, Median Filtering is particularly effective in preserving edges while removing noise, making it a preferred choice for pre-processing in various applications. Histogram Equalization is a fundamental technique in image processing to improve an image's contrast by expanding its intensity distribution. This method is especially valuable in preparing images for neural network analysis by enhancing features that may otherwise be obscured in low-contrast images. Contrast Stretching is a vital image pre-processing technique that linearly adjusts the histogram of image pixels, enhancing the contrast by spreading out the most frequent intensity values [6].

Other approaches to computer vision include traditional image processing techniques such as filtering, edge detection, and segmentation. These techniques can be useful for preprocessing data before feeding it to machine learning algorithms or neural networks. For example, an edge detector can help to highlight important object contours in an image, which will improve the results of subsequent analysis [7, 8].

Thus, the analysis of computer vision technologies demonstrates a wide range of available methods for monitoring moving objects. The choice of a particular technology depends on the requirements for accuracy, data processing speed, and available resources. Each approach has its own strengths and weaknesses, so it is important to carefully evaluate them in the context of specific monitoring tasks.

4. Methods and technologies for moving objects detection

The proposed method is based on modern computer vision and machine learning technologies, in particular Viola-Jones algorithms, convolutional neural networks, and Kalman filters.

4.1. Viola-Jones algorithm for primary detection

The Viola-Jones algorithm, used for initial object detection, is a powerful tool in the field of computer vision. This algorithm is based on the use of Haar features and AdaBoost cascade classification. The key idea behind this method is to highlight contrasting areas of an image, which allows for a gradual reduction in the number of areas analyzed. This, in turn, significantly improves the efficiency of the algorithm, as it focuses only on the most promising areas of the image where the probability of an object being present is highest [8, 9].

The Viola-Jones algorithm uses a series of simple classifiers that work in a cascaded fashion. First, a simple classifier is used to quickly filter out areas that do not contain objects. Only those areas that pass this initial filter are further processed by more complex classifiers. The algorithm is able to quickly process large amounts of data and effectively detect objects in real time.

For each region in the image, the Haar feature value is calculated, which is defined as the difference between the sums of pixel intensities in the light and dark parts:

$$f(x)=\sum_{white}^{x} I-\sum_{black}^{y} I, \tag{1}$$

where *f(x)* is the result of calculating the Haar feature value for a certain area of the image; $\sum_{white}^{x} I$ is the sum of the pixel intensities in the light parts of the template; $\sum_{black}^{y} I$ is the sum of the pixel intensities in the dark parts of the template; I is the intensity of each pixel, i.e. its brightness value, which usually varies from 0 (black) to 255 (white) in grayscale.

The formula calculates the difference between the sum of the intensities in the light and dark parts of an image for a given area. This provides a measure of the contrast in that area.

To quickly calculate Haar features, an integral image is used. The integral image at each point (x, y) contains the sum of the pixel values above that location:

$$P(x,y)=\sum_{i=0}^{x}\cdot\sum_{j=0}^{y}i(x,y). \tag{2}$$

where *P(x,y)* is the value of the integral image at the point; *i(x,y)* is the value of a pixel in the original image in coordinates; $\sum_{i=0}^{x}\cdot$ is the sum of the pixel values over all values; $\sum_{j=0}^{y}\cdot$ is the sum of the pixel values over all values.

This means that the value of the integral image at any point (x, y) is equal to the sum of all pixel values in the rectangle that starts at the upper left corner (0,0) and ends at the point (x,y). This allows you to calculate the sum of pixels in any rectangle in an image in a fixed number of operations, significantly speeding up the process of calculating Haar features.

Each object passes through a cascade of classifiers, which discard areas that do not match the desired features. At each stage, a powerful classifier trained using the AdaBoost method (adaptive boosting) is used.

The Viola-Jones algorithm allows for rapid detection of regions of interest that may contain objects, facilitating subsequent CNN classification. This optimizes the processing, reducing the computational burden on the system [10].

Haar Cascade algorithm was used to identify an object as a vehicle and count the number of passing vehicles on a particular road using traffic vedios as an input [11]. AdaBoost classifier based on Haar features, which reduces the computational load and provides very good detection accuracy. The proposed method provides an acceptable and satisfactory result for vehicle counting and road condition classification in terms of accuracy, robustness and execution time [2].

4.2. Convolutional neural networks for object detection

Convolutional Neural Networks (CNN) are a specific type of artificial neural network that was developed to work efficiently with images. The main feature of CNNs is their ability to use convolutional layers to extract important features of objects in images. This can include a variety of features such as contours, textures, and shapes, which are critical for successful object recognition.

One of the key advantages of using convolutional neural networks is their ability to process data in two dimensions. This means that they can analyze not only individual pixels, but also take into account spatial information between them. Thus, CNNs can successfully cope with the tasks of object detection and classification in complex scenes.

A CNN algorithm typically consists of several stages: first, convolution occurs, then subsampling (pooling), and then a fully connected layer for classification. Convolution allows you to detect local patterns in images, while subsampling reduces the dimensionality of the data and helps avoid overtraining the model. Finally, the fully connected layer is responsible for making the final decision about the class of the object.

The main element of CNN is the Convolutional Layer, which consists of filters (convolution kernels) that pass through the image and calculate values based on local pixels. Filters are used to extract certain features (contours, textures) in the image. The convolution formula (3) for two images *A* and *B* of size *N*×*M* looks like this:

$$(A*B)_{ij}=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}A_{m,n}B_{(i-m),(j-n)}, \tag{3}$$

where *A* is the input image and *B* is the filter.

The Pooling Layer is responsible for reducing the dimensionality of spatial data, which reduces computational costs and allows for object detection regardless of their location. The most popular method is Max Pooling, which selects the maximum value from a block of pixels.

The most commonly used activation function is ReLU (Rectified Linear Unit) $f(x)=max(0,x)$.

It provides nonlinearity in the model, allowing for the detection of complex patterns.

One drawback of the ReLU is that it does not activate for non-positive inputs, causing the deactivation of several neurons during training, which can be viewed again as a vanishing gradient problem for negative values. This problem is solved with the introduction of the Leaky rectified linear unit (Leaky ReLU, LeLU) [2], which activates slightly for negative values.

One can encounter a number of other variations of ReLU in the literature. One basic variation of the ReLU is the Parametric Rectified Linear Unit (PReLU), which has a learnable parameter, α, controlling the leakage of the negative values, presented in equation (6). In other words, PReLU is a Leaky ReLU; however, the slope of the curve for negative values of x is learnt through adaptation instead of being set at a predetermined value [12]:

$$y=\begin{cases}\alpha x\, if\; x<0\\ x\, if\; x\geq 0\end{cases}. \tag{4}$$

Fully Connected Layer is the final layer of the network that combines all the features extracted by the convolutional layers and creates an object classification (e.g., car, pedestrian).

The vehicle detection methods that are widely used can be divided into two categories. One is the R-CNN series of operation methods based on the two stages of the candidate region. It consists of two parts, namely, generating candidate regions and detecting them.

The one-stage method does not require candidate boxes to be generated in advance. Instead, candidate boxes are predicted and classified directly at various locations in the image. One of the most representative examples of one-step approaches is YOLO.

From the rapid development of the YOLO series algorithm in recent years, it can be seen that both academia and industry have great interest and expectations for this algorithm. Many scholars' research progress in object detection focuses on the improved optimization algorithm based on the YOLO algorithm. They put forward some difficult problems and research trends in the field of object detection.

YOLOv5 is the one of most advanced detection network of the YOLO object detection algorithm. Based on the YOLO v3 and YOLO v4 algorithms, the arithmetic set innovation was carried out to improve the detection speed. YOLO v5 borrowed the idea of anchor boxes to improve the speed of the R-CNN algorithm, and manually selected anchor boxes were abandoned [13].

The YOLO architecture introduced the end-to-end, differentiable approach to object detection by unifying the tasks of bounding box regression and object classification into a single neural network. Fundamentally, the YOLO network comprises three core components. The backbone, a convolutional neural network, is responsible for encoding image information into feature maps at varying scales. These feature maps are then processed by the neck, a series of layers designed to integrate and refine feature representations. Finally, the head module generates predictions for object bounding boxes and class labels based on the processed features [14].

The *backbone* consists of multiple convolutional layers that downsample the input image and extract features at different scales. Yolov5 uses cross-stage partial networks (CSP) and spatial pyramid pooling (SPP) as the main building blocks to extract important features from input images. SPP is useful for identifying the same object at different sizes and scales, which is important for the correct generalization of the model regarding object scaling.

The *neck* fuses multi-scale features from the backbone in both top-down and bottom-up fashion. This generates feature maps at three different resolutions to detect objects of different sizes. The feature pyramid architectures of the feature pyramid network (FPN) and path aggregation network (PANet) are used in constructing the neck network. Powerful semantic features are distributed throughout the FPN structure, starting from the upper feature maps and moving down to the lower feature maps. At the same time, the PANet structure is responsible for ensuring that reliable localization features are transferred from the lower feature maps to the upper ones. PANet is used as the "neck" in Yolo v5, which enables the generation of a pyramid of features.

Finally, the *head* uses these combined feature maps to predict class probabilities, bounding box coordinates, and object confidence. The head consists of convolutional layers that generate the final detection results [15, 16].

YOLOv5 is a single-stage object detection model with several versions: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. YOLOv5s is the fast and small model with a parameter of 7.0 M and a weight of 13.7 M. In YOLOv5, the versions of the network architecture are the same, and the size of the network structure is controlled by two parameters: depth factor (depth_multiple) and width factor (width_multiple). For example, the C3 operation in YOLOv5s is only done once, while the depth of YOLOv5l is three times that of v5s, so three C3 operations will be performed [17].

YOLOv5 builds upon the robust foundation laid by its predecessors, offering enhanced speed, accuracy, and efficiency. The model's architectural innovations, including the refined CSP-Darknet53 backbone and the PANet neck, contribute to superior performance metrics, making it a formidable tool for a wide range of applications. The flexibility provided by YOLOv5's various model sizes (YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x) allows for tailored deployment across different hardware environments and use-case scenarios [18].

The CSP module of the original YOLOv5s used the Leaky ReLU function and the Mish function as activation functions. An Intersection over Union (IoU) is typically utilized to calculate the location relationship between the predicted and actual boxes in target detection:

$$IoU = \frac{|M \cap N|}{|M \cup N|}. \tag{5}$$

where M is the area of the predicted bounding box and N is the area of the ground truth bounding box [19].

Also, from the developers' perspective, it is essential to point out that YOLOv5 is available in the PyTorch framework using Jupyter notebooks or Google Colab (Google Colaboratory) tools [20].

4.3. Kalman filter for object tracking

The Kalman filter for object tracking is an important element in monitoring systems used to track moving objects. This optimal linear estimation algorithm is designed to predict the position of objects based on previous observations. Its wide application in real-time object tracking is due to its ability to effectively handle noisy data and predict the position of objects even in cases of partial information loss.

The Kalman filter consists of two main stages: a prediction stage and a correction stage. In the first stage, the filter predicts where the object should be based on its previous coordinates and velocity. This allows the system to have an idea of the object's possible location, even if its position data has not been received in time. In the second stage, when new data becomes available, the filter corrects its previous prediction based on current observations.

In our case, the Kalman filter is used to predict the position of moving objects, which allows us to reduce tracking errors. This is especially important in conditions where objects can quickly change their speed or direction of movement. Due to its ability to adapt to changes in motion, the Kalman filter ensures tracking stability even when the object briefly disappears from the camera's field of view.

The model is described by state variables that contain position, velocity, and other parameters that change over time. For example, for a two-dimensional object moving, the state variables might include the X and Y positions, as well as the velocity:

$$x_k = F x_{k-1} + B u_k + w_k, \quad (6)$$

where x_k is the state at time k; F is the state transformation matrix describes how the state changes from time $k-1$ to k; B is the control matrix; u_k is the control vector; w_k is the process noise.

The Predict phase updates the current state and error matrix based on the previous state and the transformation matrix. The Update phase adjusts the predicted state based on new data obtained from the observation (in this case, from a neural network that captures the position of the object in the frame) [21, 22].

The Kalman filter allows predicting the position of the object even in the event of a short-term loss of visibility. In addition, the filter works effectively with noisy data, which is important for monitoring in urban environments [23]. The result of adjusting Kalman filter function to create a vehicle counting tool, can be considered as an economical alternative for obtaining traffic data [24].

5. A complex method of monitoring moving objects

The traditional CNN model is designed for object detection tasks in natural scenes and there are several main problems with using previous models directly to perform object detection on images of traffic scenes. Firstly, traffic scene images are often captured by cameras set up at various intersections and the different camera angles lead to large variations in target size, which can easily lead to missed and false detections. Secondly, due to hardware specifications and lighting conditions, the captured images may have low resolution and blurrier objects. Thirdly, the large coverage area of the camera results in images containing a large number of complex backgrounds, resulting in extremely small sized objects that are difficult to detect. These problems result in the traditional YOLO model performing poorly in traffic scene images and cannot be directly applied to object detection tasks in traffic scenes [25].

Therefore, we propose a three-stage video image processing method that includes the sequential application of the Viola-Jones algorithm, the CNN, and the Kalman filter (Figure 1).

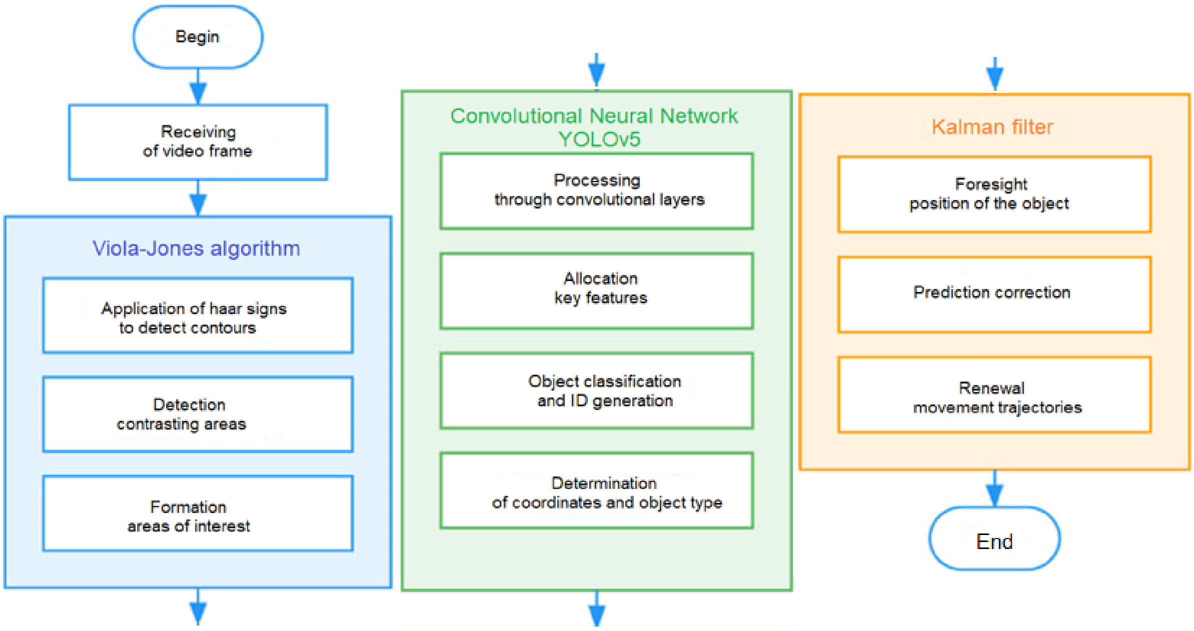
Begin
Receiving of video frame
Viola-Jones algorithm
Application of haar signs to detect contours
Detection contrasting areas
Formation areas of interest
Convolutional Neural Network YOLOv5
Processing through convolutional layers
Allocation key features
Object classification and ID generation
Determination of coordinates and object type
Kalman filter
Foresight position of the object
Prediction correction
Renewal movement trajectories
End

Figure 1: Proposed traffic monitoring method

The Viola-Jones algorithm at the initial stage of the system receives a video stream, which is then divided into separate frames. Each of these frames is subject to processing, to which the Viola-Jones algorithm is applied for rapid detection of areas where objects may potentially be located. This algorithm performs an important function by determining the contours and basic shapes of objects in the frame. This provides initial data filtering, which significantly reduces the number of areas that require further processing.

After the Viola-Jones algorithm detects regions of interest, this data is fed to a network YOLOv5 for accurate object identification. The YOLOv5 extracts important features such as contours and textures, and classifies the detected objects, such as vehicles or pedestrians. This process allows the system to not only detect the presence of an object, but also to determine its type.

Once objects are identified, their positions are passed to the Kalman filter, which starts tracking each of these objects in real time. The Kalman filter predicts the next position of the object based on previous data about its movement. When new coordinates are available, it adjusts its prediction, allowing tracking to remain stable even when the object's speed changes or it briefly disappears from the frame.

At the final stage, the system collects and stores data about each object. This may include parameters such as the trajectory of movement, speed and direction. The collected data can be used for further analysis. For example, they can be used to create traffic reports or identify potential violations of traffic rules. Thanks to this systematization of information, it becomes possible not only to monitor the situation on the roads, but also to take measures to improve it.

By combining the Viola-Jones algorithm and YOLOv5, the system is able to detect objects with high accuracy, reducing the number of false positives. The Kalman filter provides reliable prediction and correction of object positions even under conditions of variable lighting, high object speed, or temporary loss of visibility.

In summary, the conceptual model integrates the above methods to achieve the main goal of ensuring reliable detection, identification and tracking of objects in real time.

Each of these models plays an important role. Viola-Jones algorithm provides speed for primary processing, CNN provides high recognition accuracy, and Kalman filter guarantees stable tracking.

6. Testing of proposed method

A Python program was developed to test the proposed algorithm. The source code is available on GitHub: https://github.com/khai-edu/Traking_2025.

A computer with a budget NVIDIA GTX 1650 graphics card was used for testing. Testing the system covers several key aspects.

1. Impact of image quality – testing the system in low light or bright conditions. It is envisaged that filters will be used to increase image contrast and brightness.
2. Object detection using YOLOv5 – evaluating the ability of a deep learning model to detect objects in frames with different lighting conditions and visual obstacles such as rain or snow.
3. Tracking objects using the Kalman filter – assessing the accuracy of object coordinate correction, which allows increasing tracking stability even in the presence of errors in the detected coordinates.

All of these tests are an important part of analyzing the effectiveness of the developed system, as they allow you to assess its ability to work in real-world conditions and identify possible weaknesses that need improvement.

Object labels:

- green rectangles are the result of YOLOv5, where each detected vehicle receives the coordinates of the rectangle and a Kalman Filter (KF) label;
- blue rectangles are objects detected by the Viola-Jones algorithm. They can serve as preliminary regions for YOLO validation;
- red dots represent the coordinates of the center of the object's predicted position using the Kalman filter.

Test 1: Normal conditions. The image (Figure 2) shows how the program analyzes and processes a video stream under normal conditions.

YOLOv5 recognition looks quite accurate – green frames cover objects well (cars, buses, etc.), which allows for high-quality identification of objects even in crowded conditions. All detected vehicles are in the area of interest, which indicates the correct setting of the detection area. Viola-Jones algorithm. highlights objects with wider blue frames, demonstrating the effectiveness of coarse detection.

The Kalman filter predicts the trajectory of each object. The red dots are within the green frames, which indicates accurate prediction. Even at an intersection with heavy traffic, the program does not lose track of objects. Tracking works stably for several objects at the same time. Despite their proximity to each other, there is no false merging of tracks.

The program demonstrates high stability to conditions of a large number of objects in the frame. Detection does not depend on the location of cars: recognition works for vehicles on the road and in the parking lot. The system takes into account movement and change of coordinates, allowing long-term monitoring of objects.

However, testing reveals some problems. In crowded areas, objects overlap each other. This can reduce tracking accuracy. You can also notice a few false positives.

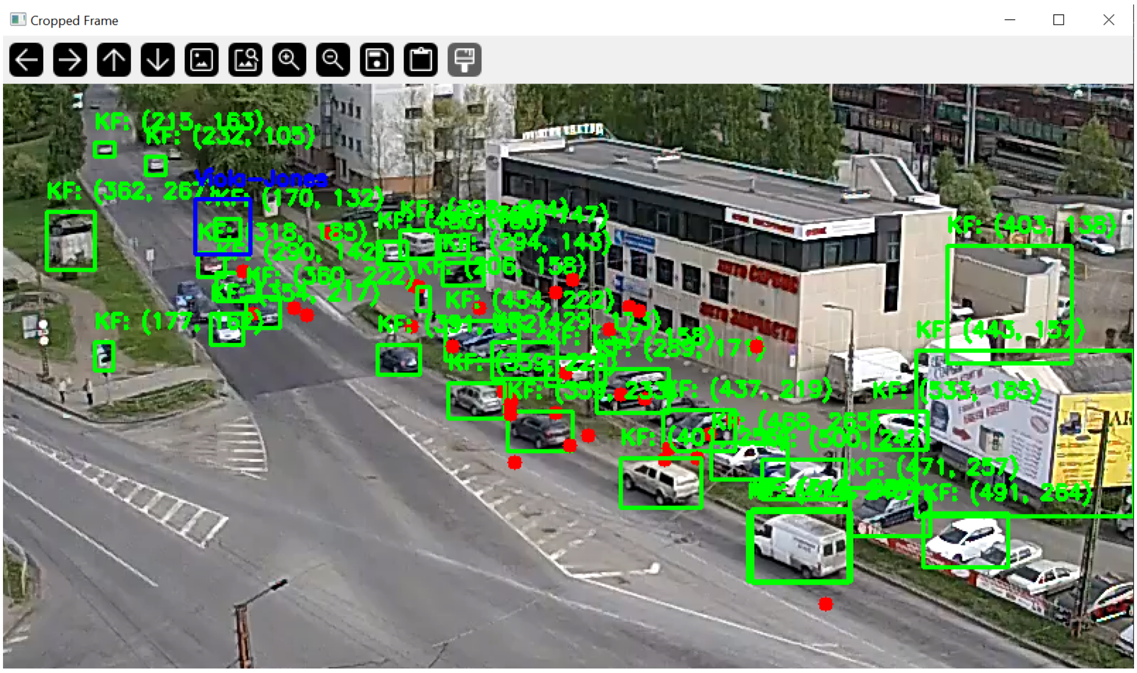

Figure 2: Example of the developed program in normal mode

Test 2: Difficult weather conditions. The next video stream (Figere 3) to test is a road traffic scene with bad weather conditions. The frame shows several Viola-Jones hits. The object identified as a road sign (blue frame) is identified, but the frame is less accurate compared to YOLOv5. The Viola-Jones algorithm often generates false detections or allocates overly general areas and unnecessary frames for objects (e.g., empty areas with no visible cars).

YOLOv5 works well, the green frames accurately cover the cars, which confirms the robustness of the model to noise.

The coordinate marks (KF: (x, y)) indicate the predicted centers of the objects. This indicates the stable operation of the Kalman Filter.

The red dots show the predicted centers of the objects. They are located close to the center of the cars, which indicates the effective operation of the tracker.

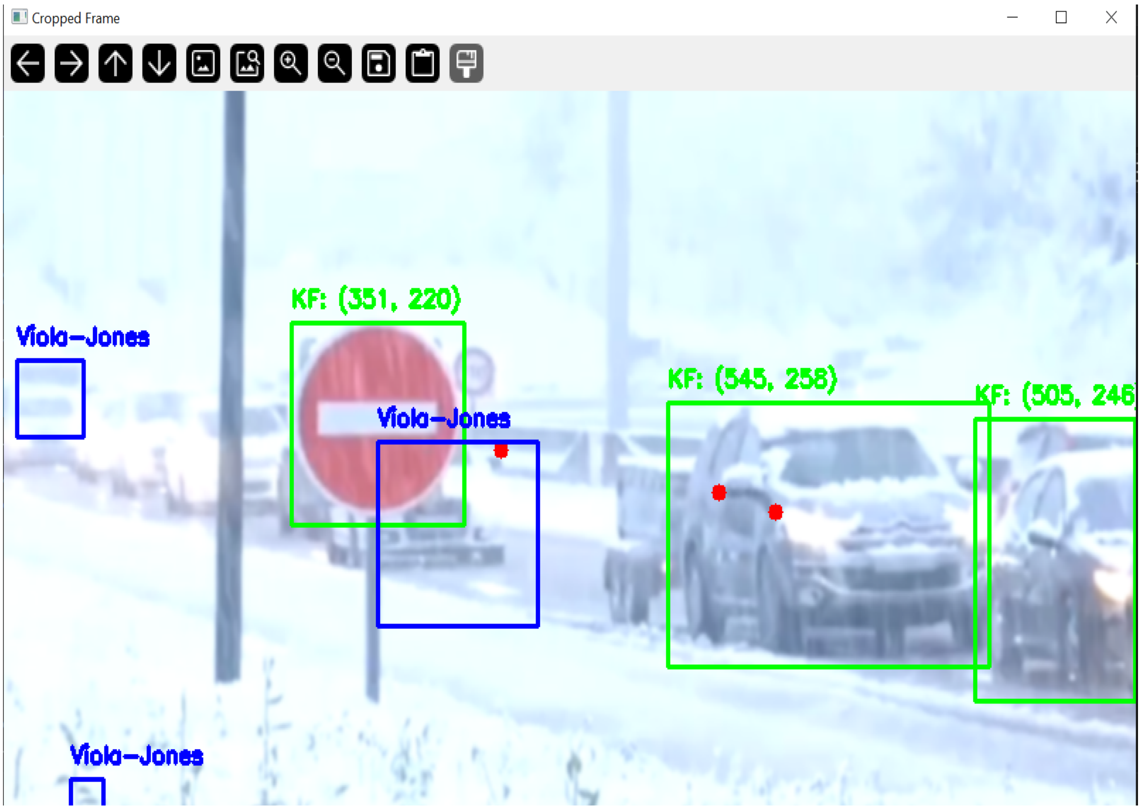

Figure 3: Example of the developed program working in bad weather conditions

Although the lighting in the frame is quite bright due to the snowy environment, the frames have been pre-processed to reduce the impact of high brightness and ensure stable operation of the algorithms. However, white cars are recognized less accurately.

After analyzing this video stream, it can be noted that the algorithms demonstrate stable operation in difficult conditions (snow, bright light), but there are false positives. To adapt the algorithm to specific conditions, you can replace or strengthen the Viola-Jones model by training a custom model. You can also reduce the weight of Viola-Jones in the final integration if accuracy is critical. For even greater stability, you can refine the YOLOv5 model, for example, add more training examples for snowy road conditions.

Test 3: Low light. In Figure 4, with low light, the quality enhancement function `convertScaleAbs` was applied, which increased the contrast and brightness. The parameter `alpha = 2.1` significantly increases the contrast, making the objects in the image more visible. This is especially noticeable on the bright areas of the road and cars. The parameter `beta = 50` also helps to brighten the dark areas. The cars are detected quite accurately. It can be seen that image enhancement improves the performance of YOLO, as the objects have a clearer outline.

Viola-Jones finds objects in the frame, but significantly less than YOLO due to its high sensitivity to noise. The Kalman filter motion predictions coincide with the centers of the cars, demonstrating correct object tracking.

For more accurate detection, a YOLO model trained on low-light data can be used.

Test 4: Heavy traffic. In the frame on Figure 5 several cars are visible on the road.

The detection works quite accurately, even for objects with different sizes, distances and angles of inclination. The green rectangles cover almost all the cars on the road.

The Viola-Jones algorithm only finds a small fraction of objects, indicating its limited effectiveness in complex scenes. This confirms its role as a coarse filter to optimize the performance of YOLO.

The Kalman Filter (KF) coordinates correctly predict the average position of the object, which is useful for tracking moving targets in a video stream.

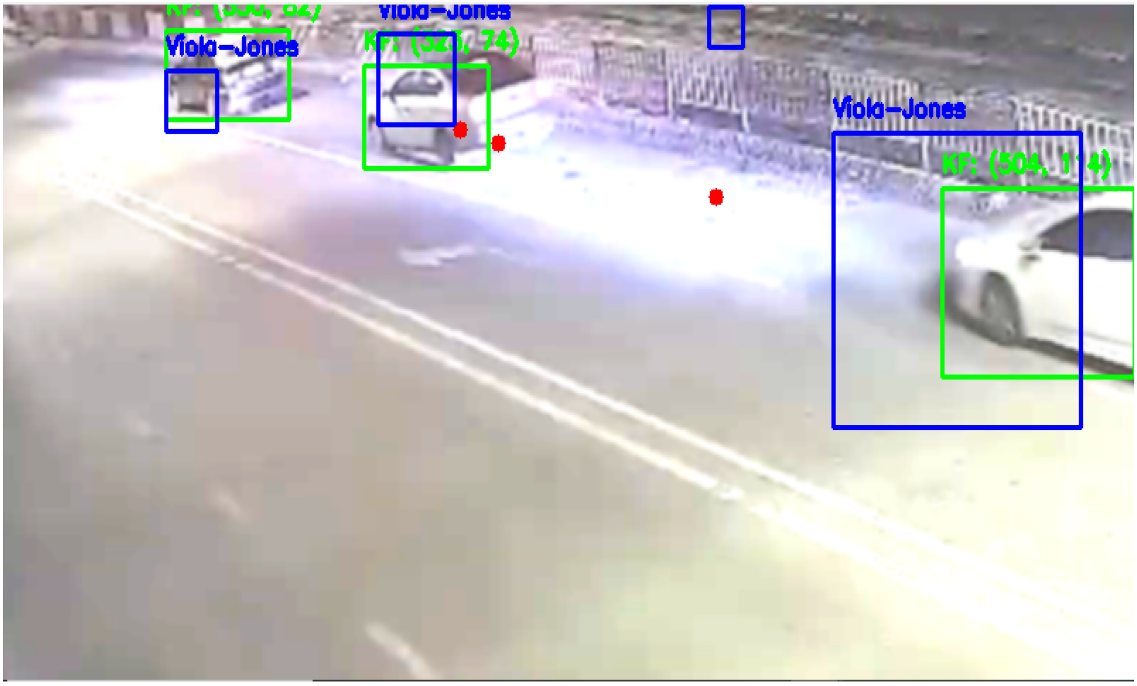

Figure 4: Testing the application in low visibility conditions

To optimize the performance of the Viola-Jones algorithm, it is necessary to use pre-processing (e.g., background clipping) to improve detection in scenes with low levels of detail. To avoid duplication and false positives, it is possible to combine detections from YOLO and Viola-Jones, for example, by comparing the coordinates of rectangles and keeping only those that match both algorithms. To minimize the deviation of red dots, it is possible to adjust the parameters of the Kalman filter (in particular, `processNoiseCov` and `measurementNoiseCov`). To avoid overlapping multiple detections on the same object, it is necessary to implement logic to remove duplicates (e.g. checking whether rectangles intersect).

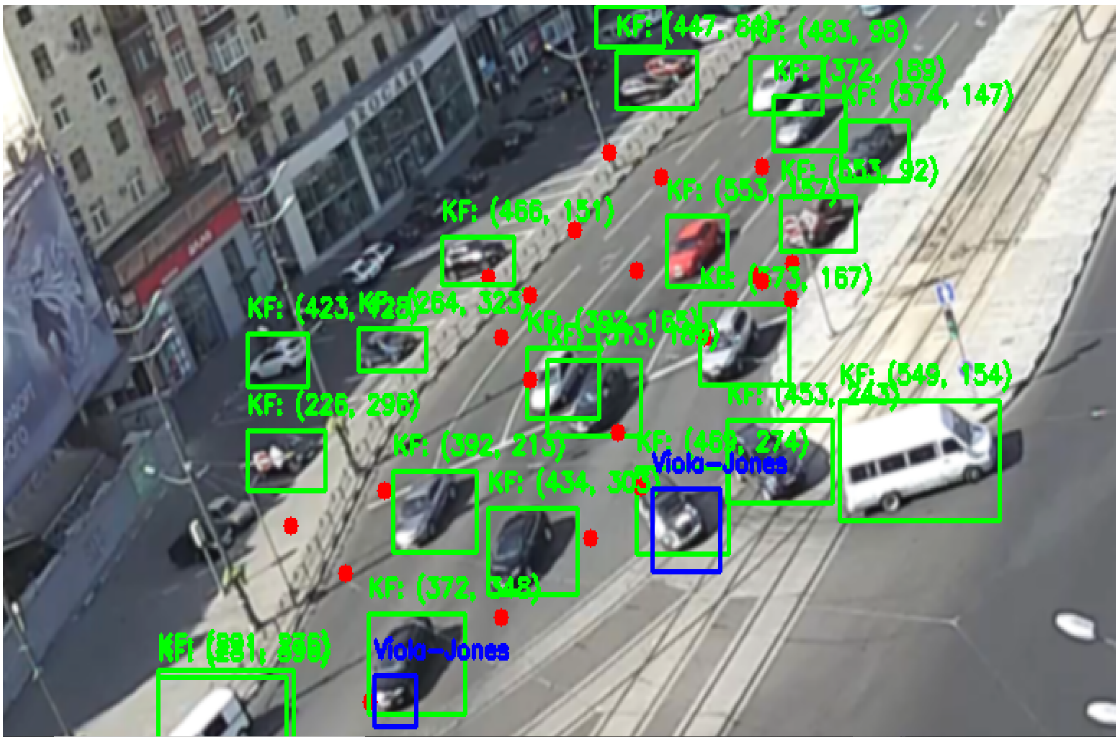

Figure 5: Testing the application under heavy traffic

Future experiments. For a more accurate assessment the proposed method integrating the Viola-Jones algorithm, YOLOv5, and Kalman filter will be benchmarked on a publicly available dataset, such as the KITTI Vision Benchmark Suite, designed for traffic scene analysis and containing annotated data for object detection and tracking. The testing process will encompass the following steps (Figure 6):

1. Preprocessing the dataset images (normalization, contrast enhancement, noise reduction) to mitigate variations in lighting and weather conditions.
2. Applying the Viola-Jones algorithm for initial detection of regions of interest containing potential objects.
3. Utilizing YOLOv5 for precise classification and identification of objects (e.g., vehicles, pedestrians) within these regions.
4. Employing the Kalman filter for real-time object tracking, including prediction and correction of object positions.
5. Evaluating performance using standard metrics: Mean Average Precision (mAP) for detection accuracy, Multiple Object Tracking Accuracy (MOTA), and Multiple Object Tracking Precision (MOTP) for tracking stability.

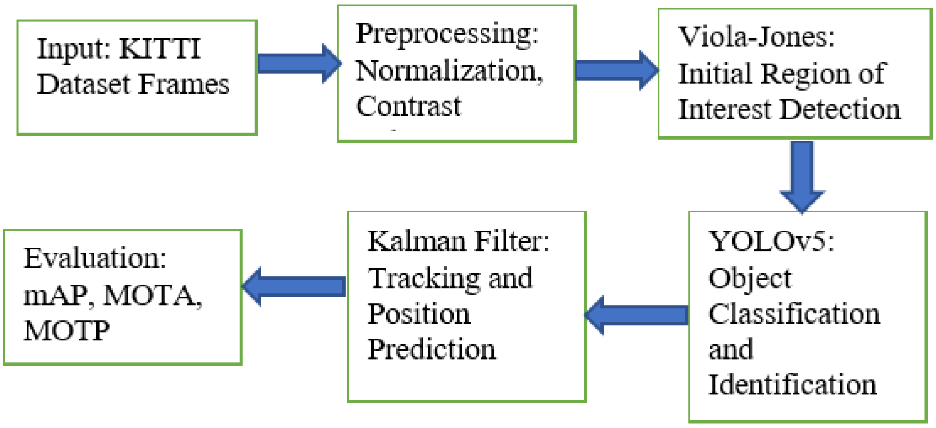
Input: KITTI Dataset Frames
Preprocessing: Normalization, Contrast
Viola-Jones: Initial Region of Interest Detection
YOLOv5: Object Classification and Identification
Kalman Filter: Tracking and Position Prediction
Evaluation: mAP, MOTA, MOTP

Figure 6: Testing scenario

The method will be tested across diverse scenarios (e.g., normal conditions, low light, adverse weather) to assess robustness. Results will be compared against state-of-the-art methods, such as YOLOv8, to quantify improvements in accuracy, speed, and computational efficiency on a budget equipment.

7. Conclusions and perspectives

By consistently using these methods, the system achieves a balance between speed, accuracy, and stability. The Viola-Jones algorithm provides fast initial detection of objects in the frame, the CNN performs detailed classification, and the Kalman filter guarantees continuous and smooth tracking. The integration of these components allows the monitoring system to operate in real time, efficiently processing the video stream and maintaining accuracy in various environmental conditions.

The Kalman filter is fast, but working with a large number of objects in real time also puts a strain on the processor.

The YOLOv5 family of models provides sufficiently high performance on limited computing resources. Even budget GPUs can provide a significant speed increase when using the YOLOv5s model. Using more powerful CNN models such as YOLOv8 or YOLOv11 will help improve detection results, but requires more powerful hardware, especially for real-time operation. It is also possible to use a server or device with higher power to process the video stream, and then transmit the results to the client device. The integration of the proposed solutions will significantly increase the efficiency of the system without losing the quality of detection and tracking.

Image pre-processing, the use of filters, and contrast enhancement can improve monitoring results in difficult weather conditions and poor lighting. However, developing recommendations for variable lighting conditions requires further research.

The proposed method successfully coped with the recognition of moving objects in videos of varying quality, including in low light and difficult weather conditions. Testing the system in various conditions demonstrated its high functionality and ability to adapt to environmental changes. Further research could be aimed at integrating even more flexible adaptive algorithms and improving the models to improve performance and work in extremely challenging conditions.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Asma Ait Ouallane, Ayoub Bahnasse, Assia Bakali, Mohamed Talea. Overview of Road Traffic Management Solutions based on IoT and AI. Procedia Computer Science, Volume 198, 2022, pp 518-523, ISSN 1877-0509. doi: 10.1016/j.procs.2021.12.279.

[2] Bhuiyan, T. A. U. H., Mrinmoy Das, and Md Shamim Reza Sajib. "Computer vision based traffic monitoring and analyzing from on-road videos." Global journal of computer science and technology (2019). doi : 10.17406/GJCST.

[3] Zhao, X., Wang, L., Zhang, Y. et al. A review of convolutional neural networks in computer vision. Artif Intell Rev 57, 99 (2024). doi: 10.1007/s10462-024-10721-6.

[4] Khan, Asharul Islam, and Salim Al-Habsi. "Machine learning in computer vision." Procedia Computer Science 167 (2020): 1444-1451. doi:10.1016/j.procs.2020.03.355.

[5] Nachuan Ma, Jiahe Fan, Wenshuo Wang, Jin Wu, Yu Jiang, Lihua Xie, Rui Fan, Computer vision for road imaging and pothole detection: a state-of-the-art review of systems and algorithms, Transportation Safety and Environment, Volume 4, Issue 4, December 2022, tdac026, doi: 10.1093/tse/tdac026.

[6] M. Raisul Islam et al., "Deep Learning and Computer Vision Techniques for Enhanced Quality Control in Manufacturing Processes," in IEEE Access, vol. 12, pp. 121449-121479, 2024, doi: 10.1109/ACCESS.2024.3453664.

[7] Abbas, Aymen Fadhil, et al. "A comprehensive review of vehicle detection using computer vision." TELKOMNIKA (Telecommunication Computing Electronics and Control) 19.3 (2021): 838-850. doi:10.12928/TELKOMNIKA.V19I3.12880.

[8] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012. doi: 10.1109/CVPR.2012.6248074.

[9] Viola, Paul, and Michael Jones. "Fast and robust classification using asymmetric adaboost and a detector cascade." Advances in neural information processing systems 14 (2001). https://www.researchgate.net/publication/2539888_Fast_and_Robust_Classification_using_Asymmetric_AdaBoost_and_a_Detector_Cascade.

[10] Romdhane, Nadra Ben, Hazar Mliki, and Mohamed Hammami. "An improved traffic signs recognition and tracking method for driver assistance system." 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). IEEE, 2016. doi: 10.1109/ICIS.2016.7550772.

[11] Pavani, K., and P. Sriramya. "Novel vehicle detection in real time road traffic density using haar cascade comparing with KNN algorithm based on accuracy and time mean speed." Revista Geintec-Gestao Inovacao E Tecnologias 11.2 (2021): 897-910. doi:10.47059/revistageintec.v11i2.1723.

[12] Maniatopoulos, Andreas, and Nikolaos Mitianoudis. 2021. "Learnable Leaky ReLU (LeLeLU): An Alternative Accuracy-Optimized Activation Function" Information 12, no. 12: 513. doi: 10.3390/info12120513.

[13] Zhang, Yu, Zhongyin Guo, Jianqing Wu, Yuan Tian, Haotian Tang, and Xinming Guo. 2022. "Real-Time Vehicle Detection Based on Improved YOLO v5" Sustainability 14, no. 19: 12274. doi: 10.3390/su141912274.

[14] J. E. Gallagher and E. J. Oughton, "Surveying You Only Look Once (YOLO) Multispectral Object Detection Advancements, Applications, and Challenges," in IEEE Access, vol. 13, pp. 7366-7395, 2025, doi: 10.1109/ACCESS.2025.3526458.

[15] Dewi, Christine, Rung-Ching Chen, Yong-Cun Zhuang, and Henoch Juli Christanto. 2022. "Yolov5 Series Algorithm for Road Marking Sign Identification" Big Data and Cognitive Computing 6, no. 4: 149. doi: 10.3390/bdcc6040149.

[16] Khanam, Rahima, and Muhammad Hussain. "What is YOLOv5: A deep look into the internal features of the popular object detector." arXiv preprint arXiv:2407.20892 (2024). doi:10.48550/arXiv.2407.20892.

[17] Guo, G., Zhang, Z. Road damage detection algorithm for improved YOLOv5. Sci Rep 12, 15523 (2022). doi: 10.1038/s41598-022-19674-8.

[18] Jaiswal, Sandeep Kumar, and Rohit Agrawal. "A comprehensive review of YOLOv5: advances in real-time object detection." Int. J. Innov. Res. Comput. Sci. Technol 12.3 (2024): 75-80. doi: 10.55524/ijircst.2024.12.3.12.

[19] Shao, Lei, Han Wu, Chao Li, and Ji Li. 2023. "A Vehicle Recognition Model Based on Improved YOLOv5" Electronics 12, no. 6: 1323. doi: 10.3390/electronics12061323.

[20] Horvat, Marko, Ljudevit Jelečević, and Gordan Gledec. "A comparative study of YOLOv5 models performance for image localization and classification." Central European Conference on Information and Intelligent Systems. Faculty of Organization and Informatics Varazdin, 2022. https://archive.ceciis.foi.hr/public/conferences/2022/Proceedings/IS/IS4.pdf.

[21] Puttemans S. OpenCV Tutorials. Object Detection. Cascade Classifier Training. https://docs.opencv.org/3.3.0/dc/d88/tutorial_traincascade.html.

[22] Selvaraj Vasantha Kumar, Traffic Flow Prediction using Kalman Filtering Technique, Procedia Engineering, Volume 187, 2017, Pages 582-587, ISSN 1877-7058, doi: 10.1016/j.proeng.2017.04.417.

[23] Ali, Shuja, et al. "Vehicle detection and tracking in UAV imagery via YOLOv3 and Kalman filter." Computers, Materials & Continua. 2023. doi: 10.32604/cmc.2023.038114.

[24] Espejel-García, Daphne, et al. "An Alternative Vehicle Counting Tool Using the Kalman Filter within MATLAB." Civil Engineering Journal 3.11 (2017). doi: 10.28991/cej-030935.

[25] Li, Ang, Shijie Sun, Zhaoyang Zhang, Mingtao Feng, Chengzhong Wu, and Wang Li. 2023. "A Multi-Scale Traffic Object Detection Algorithm for Road Scenes Based on Improved YOLOv5" Electronics 12, no. 4: 878. doi: 10.3390/electronics12040878.