

D.4.2.2 DEFINITION OF AN ARTIFICIAL INTELLIGENCE (AI) TOOL TO ESTIMATE TRAFFIC FLOWS IN A SPECIFIC AREA

Document Control Sheet

Project Number:	10249002
Project Acronym	MIMOSA
Project Title	Maritime and Multimodal Sustainable Passenger transport solutions and services
Start Date	01/01/2020
End Date	30/06/2023
Duration	36 months

Related Activity:	A4.2. Piloting sustainable different transport modalities and e-services
Deliverable Name:	(D.4.2.2) Definition of an Artificial Intelligence (AI) tool to estimate traffic flows in a specific area.
Type of Deliverable	Study
Language	English
Work Package Title	Analysing and piloting new sustainable mobility solutions
Work Package Number	WP4
Work Package Leader	PP3 (Institute for Transport and Logistics Foundation)

Status	Final
Author(s)	PP3
Version	1
Deliverable Due Date	December 2022
Delivery Date	02/2023

Table of contents

1 Introduction and background of the Mimosa Artificial Intelligence (AI) activities	4
2 Artificial Intelligence: an overview of the different application in the framework of the Mimosa pilot	6
2.1. Introduction to the key concepts	6
2.2. Object Detection and Classification Models: an introduction	9
2.3. Object Detection and Classification Models: YOLO	11
2.4. Object Detection and Classification Models: Free Datasets	14
3 Methodological Aspects: The Mimosa tool's three main modules	16
3.1. Video Stream Processing Module	16
3.2. Short-Term Tracking Module	18
3.3. Statistics Generation Module	19
4 Mimosa Testing Activities at the Bologna Airport	20
4.1. First Recording Session	24
4.2. Second Recording Session	26
5 Problems and Potential Solutions of the MIMOSA AI solution	28
6 Conclusion and Recommendations	29
7 References	31
Annex 1. The Mimosa tool: "Fluxus-AI" App. Guide for the utilization	32

1 Introduction and background of the Mimosa Artificial Intelligence (AI) activities

This Mimosa pilot action is related to the definition of an Artificial Intelligence (AI) tool based on computer vision to estimate traffic flows in a specific area. The AI tool intends to monitor both vehicles (private and public) and person's flows. The collected data intends to provide an innovative technological tool able to provide in an easier and more cost-efficient way data for provide traffic data for mobility planning and management. The solution was designed in order to be replicable in other contexts with a particular attention on the Italy-Croatia area.

In particular this Mimosa pilot was born from the need to simplify and automatize the traffics flows data collection. In fact, the traffic flows data collection often required relatively high-cost technologies (traffic radars, dedicated sensors, etc.) or a high number of people spending a lot of times to collect the required data along the roads.

This Mimosa AI solution was developed as a potential alternative solution to the traditional traffic detector technologies. Compared for example to the inductive loop detectors technology, the Mimosa AI solution has the advantage to not require additional infrastructure works for the installation of these detectors in the roads. Moreover, AI allows to detect bicycle and pedestrian in a more accurate and complete way compared to inductive loop detectors. Moreover, as evidenced in the scientific literature, thanks to artificial intelligence it is possible to reduce the costs and the human resources needed to collect data on traffic flows. An alternative traffic detection technology to the AI solution developed in the Mimosa project is traffic radars technology. This is a very reliable solution but sometimes the costs for its implementation are quite high.

For all these reasons, this pilot was focused on developing an innovative technological solution based on Artificial Intelligence able to automatize traffic data collection. The aim is to provide the decision-makers with new tools for traffic monitoring and data-oriented decisions making on the topic of sustainable transport promotion.

For all these reasons, the purpose of this report is to document the progress and outcomes of the vehicular flow monitoring activities conducted by Fondazione ITL in collaboration with the external technical experts "GoatAI SRL", as part of the Interreg Italy-Croatia Mimosa project.

The analysis of traffic flows is an important aspect of transportation engineering, as it helps to understand the movement patterns of vehicles on roads and highways. This information can be used, for instance, to optimize traffic signals, improve road design, and reduce congestion. In recent years, artificial intelligence (AI) has emerged as a powerful tool for analysing vehicular flows, as it can process and analyse large amounts of data and images quickly and accurately. In particular in this Mimosa pilot a specific subset of the artificial intelligence was used: the computer vision technologies.

In this project, an AI system was used in order to analyse vehicular flows on a specific road and/or highway as in Figure 1. The goal of the project is to understand the patterns of vehicle movement and use this information to make recommendations for improving traffic flow. The results of this

analysis are presented in this report, which will include an overview of the AI system used, a description of the data collected, and an analysis of the vehicular flow patterns.

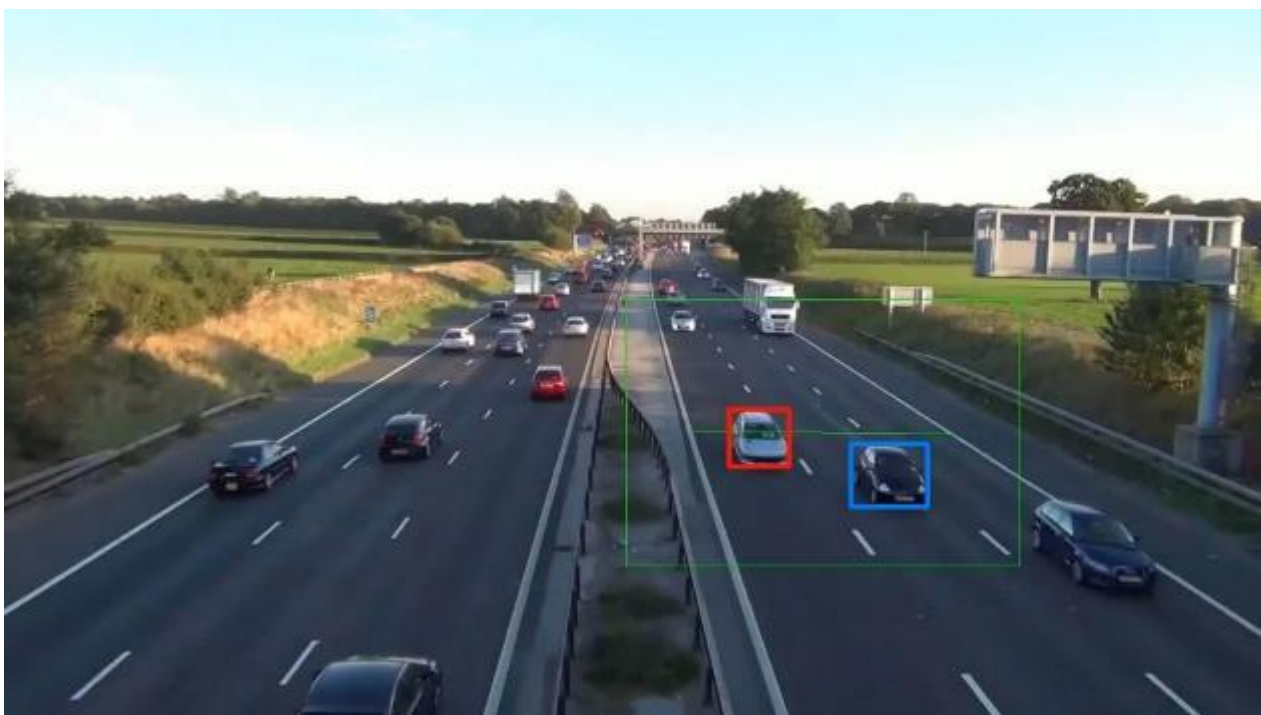


Figure 1. Qualitative results of the proposed traffic monitoring system. The solution is able to localize and track the transiting vehicles (Source: *GoatAI SRL*).

The main objectives of the system described in this Mimoso project report are:

- (a) accurately count the number of vehicles in a flow;
- (b) accurately classify the types of vehicles that go through an identified road;
- (c) extract and analyse aggregate statistics related to vehicular flow.

The other key goal of this pilot action is to develop an open-source tool able to be used also in other territorial contexts. For this purpose, the AI tool was developed starting only from open-source algorithms and models. The choice of open-source technology undoubtedly has remarkably value per se. However, it could bring to some limitations by reducing the set of potential/candidate resources to be used. In the present pilot it implied reducing the number of vehicles' categories monitored. Nevertheless, also considering the needs to be addressed, this proved to be acceptable since the monitored categories are enough to have a clear analysis of the traffic flows characteristics.

A beta version of this AI tool for traffic monitoring is available on an online repository on GitHub (<https://github.com/ITLBologna/Fluxus-AI>).

2 Artificial Intelligence: an overview of the different application in the framework of the Mimosa pilot

In this section, it is provided the basis for understanding the artificial intelligence (AI) solutions adopted for the implementation of the vehicular flows analysis system. In particular, a specific focus is dedicated to the object detection and classification models and the data required for their training and testing.

2.1. Introduction to the key concepts

The emergence of artificial intelligence (AI) has played a key part in ushering in the Fourth Industrial Revolution. According to the World Economic Forum, “it is disrupting almost every industry in every country.” Considering the complexity of this topic, this report starts from a fast definition of the key definitions and concepts related to the Mimosa AI solution.

Artificial intelligence (AI)

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. As defined by Deloitte, “in general terms, AI refers to a broad field of science encompassing not only computer science but also psychology, philosophy, linguistics and other areas. AI is concerned with getting computers to do tasks that would normally require human intelligence. Having said that, there are many points of view on AI and many definitions exist”¹. Artificial Intelligence works with large amounts of data that are first combined with fast, iterative processing and smart algorithms that allow the system to learn from the patterns within the data. This way, the system would be able to deliver accurate or close to accurate outputs.

¹ <https://www2.deloitte.com/se/sv/pages/technology/articles/part1-artificial-intelligence-defined.html>

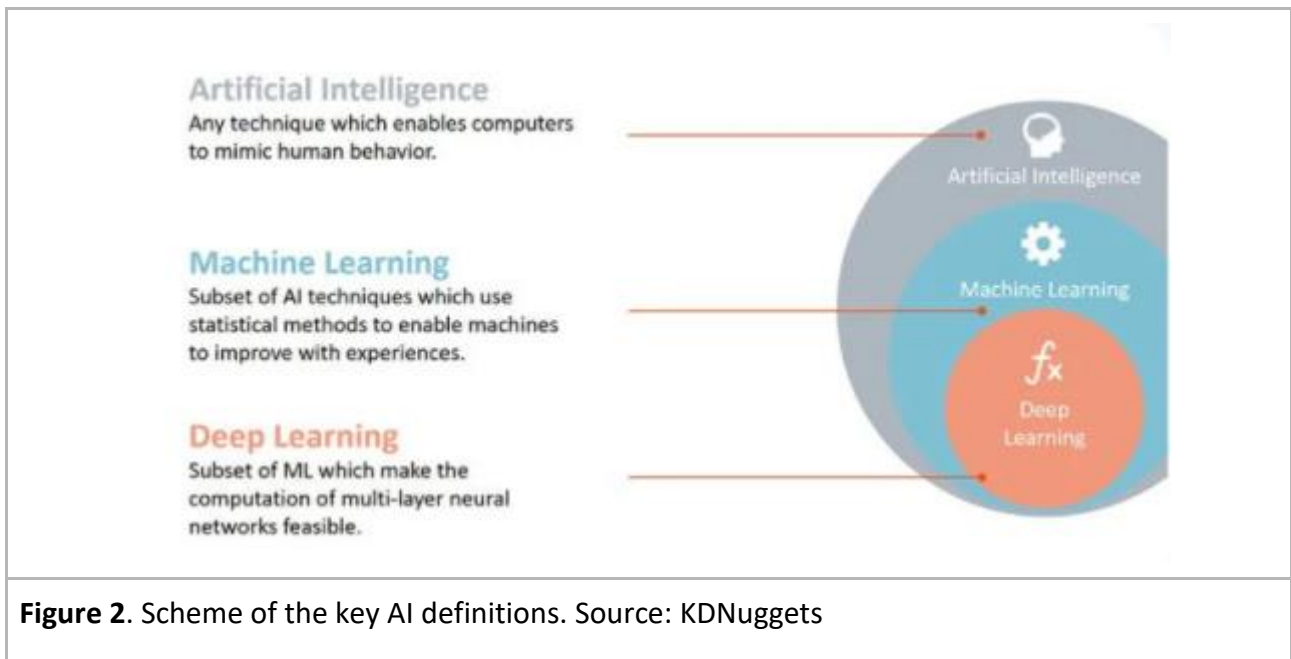


Figure 2. Scheme of the key AI definitions. Source: KDNuggets

Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them. Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch.

Deep Learning

Deep learning is a subset of machine learning that uses artificial neural networks to mimic the learning process of the human brain. Deep Learning gets its name from the fact that we add more "Layers" to learn from the data. Where machine learning algorithms generally need human correction when they get something wrong, deep learning algorithms can improve their outcomes through repetition, without human intervention. A machine learning algorithm can learn from relatively small sets of data, but a deep learning algorithm requires big data sets that might include diverse and unstructured data.

Machine learning	Deep learning
A subset of AI	A subset of machine learning
Can train on smaller data sets	Requires large amounts of data
Requires more human intervention to correct and learn	Learns on its own from environment and past mistakes
Shorter training and lower accuracy	Longer training and higher accuracy
Makes simple, linear correlations	Makes non-linear, complex correlations
Can train on a CPU (central processing unit)	Needs a specialized GPU (graphics processing unit) to train

Source: <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>

Artificial Neural Networks

The Artificial Neural Networks were developed by getting inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. The artificial neural networks are a subset of machine learning and are at the heart of deep learning algorithms. Artificial neural networks are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Artificial Neural Networks are one of the most important tools in Machine Learning to find patterns within the data, which are far too complex for a human to figure out and teach the machine to recognize. The Artificial Neural Networks “learns” through a “training process” carried out by providing a massive amount of input. This training consists of processing examples, each of which contains a known "input" and "result," forming probability-weighted associations between the two, which are stored within the data structure of the net itself. This difference is the error. The network then adjusts its weighted associations according to a learning rule and using this error value. Successive adjustments will cause the neural network to produce output that is increasingly similar to the target output. After a sufficient number of these adjustments, the training can be terminated. This is known as supervised learning.

Convolutional neural network (CNN)

A CNN is a particular typology of artificial neural network that “helps a machine learning or deep learning model “look” by breaking images down into pixels that are given tags or labels. It uses the labels to perform convolutions (a particular kind of mathematical operation on two functions to produce a third function) and makes predictions about what it is “seeing.” The neural network runs convolutions and checks the accuracy of its predictions in a series of iterations until the predictions start to come true. It is then recognizing or seeing images in a way similar to humans”².

Computer vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognize images. Two essential technologies are used to accomplish this: a type of machine learning called deep learning and a convolutional neural network (CNN).

2.2. Object Detection and Classification Models: an introduction

Object detection models are typically trained using deep convolutional neural networks (CNNs³) as defined in the previous paragraph. To train a CNN for object detection, a large dataset of labelled images is required. These images must be annotated with bounding boxes⁴ around the objects of interest and corresponding classification labels. The CNN is then trained to predict the presence and location of these objects in new images. This process is known as “supervised learning”, as the model is given labelled examples to learn from.

Once trained, object detection models can be used to analyse new images or videos and detect the presence and location of objects in real time. An object detection and classification model typically outputs a list of bounding boxes around the detected objects, along with a classification label for each box indicating the type of object it contains.

Modern object detectors can generally be divided into two macro-categories: *two-stage detectors* and *one-stage detectors*. Figure 3 (a) shows the basic architecture of two-stage detectors, while Figure 2 (b) illustrates the basic architecture of one-stage detectors.

² <https://www.ibm.com/topics/computer-vision>

⁴ The bounding box is a rectangular box that contains an object or a set of points. When used in digital image processing, the bounding box refers to the border's coordinates that enclose an image.

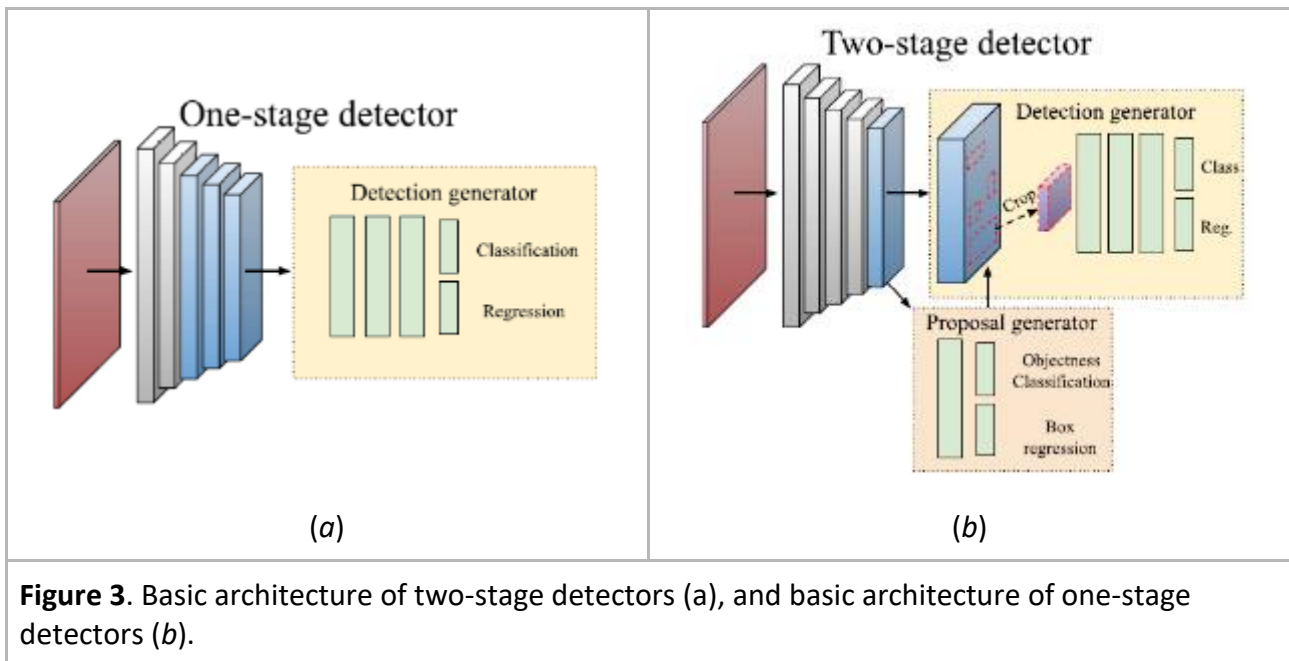


Figure 3. Basic architecture of two-stage detectors (a), and basic architecture of one-stage detectors (b).

One-stage detectors, such as **You Only Look Once (YOLO)** [1], use a single convolutional neural network (CNN) to directly predict the bounding boxes and class probabilities for objects in an image. In contrast, two-stage detectors, such as **Faster R-CNN** [2], use a region proposal network (RPN) in addition to a separate classification network to first generate a set of candidate bounding boxes (also known as region proposals), and then use a separate CNN to classify each proposal and refine the bounding box coordinates.

One-stage detectors are generally faster and more efficient than two-stage detectors, which makes them more suitable for real-time applications. This is because one-stage detectors do not require the additional step of generating candidate bounding boxes, which can be computationally expensive. In addition, one-stage detectors have a simpler architecture, which makes them easier to implement and train.

However, two-stage detectors tend to have higher accuracy compared to one-stage detectors, particularly on more challenging datasets. This is because the additional step of generating region proposals allows the detector to consider a larger number of potential object locations, which can be beneficial for detecting objects that are small or have complex shapes.

In summary, one-stage detectors are faster and more efficient than two-stage detectors, making them well-suited for real-time applications, but two-stage detectors tend to have higher accuracy. The choice between a one-stage or two-stage detector will depend on the specific requirements of the application, including the desired accuracy, speed, and complexity.

For the specific case of the Mimososa project, detection and classification were limited to a small subset of objects and sporadic errors do not significantly affect the final aggregate statistics derived from the analysis of several hours of footage. For these reasons, after careful

consideration, we have determined that it would be more appropriate to adopt a one-stage detector as the first step for the vehicular flow analysis system. In particular, it is chosen to take advantage of one of the latest and best-performing variants of the famous open-source YOLO model, namely **YOLOx** [3].

2.3. Object Detection and Classification Models: YOLO

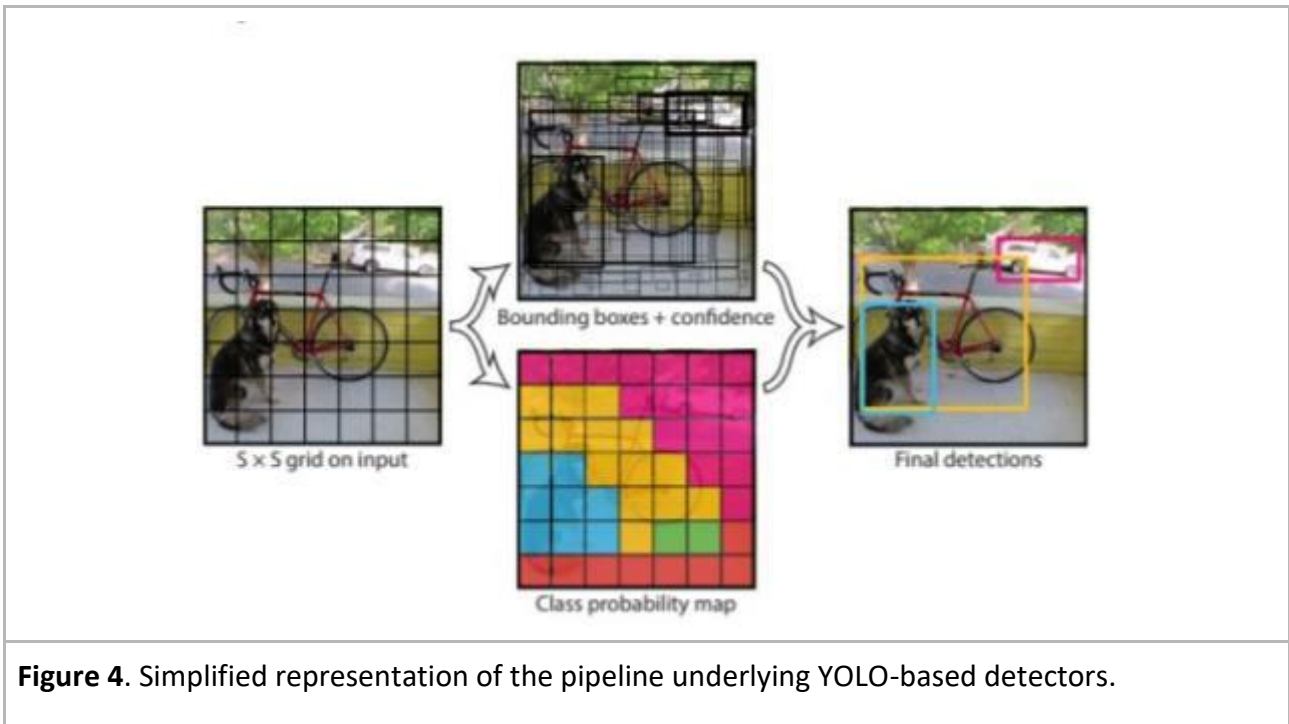
In this section, it is briefly discussed the history and evolution of the YOLO (You Only Look Once) object detection and classification model, culminating in the version used as the foundation for the Mimosa project pilot, in order to more effectively understand the various design choices.

YOLO (You Only Look Once), presented in 2015, is one of the most famous open-source and real-time object detection and classification system. It is designed to be fast and efficient, able to process images in real time and accurately identify and localize objects in the image.

One of the key features of YOLO is that it treats object detection as a regression problem rather than a classification problem. Rather than presenting the network with a set of candidates and asking it to classify each candidate as an object or background, YOLO directly predicts the image pixels as objects and their bounding box attributes.

To do this, YOLO divides the input image into a grid of cells, and each cell is responsible for detecting objects within its region. Each cell can predict a pre-defined number of bounding boxes, and each prediction consists of $(5 + C)$ elements:

- the centre of the bounding box (x and y),
- the dimensions of the box (w and h)
- a confidence score for that bounding box that indicates how confident the model is that it actually encloses an object
- a sequence of C unit-sum values, where C is the total number of considered classes; the i -th value of that sequence represents the confidence in the model to associate the bounding box with the i -th class.



YOLO is trained using a multitask loss function that combines the loss of all predicted components. Non-maximum suppression (NMS) is used to remove class-specific multiple detections. While YOLO was very successful at the time of its release and outperformed its contemporaries in both accuracy and speed, it did have some limitations. It struggled with localization accuracy for small or clustered objects, and it was limited in the number of objects it could detect per cell. These issues were addressed in later versions of YOLO, such as YOLOv2, YOLOv3, and YOLOx. YOLOv2, also known as YOLO9000, was released in 2016 and made several improvements to the original YOLO algorithm. One of the main improvements in YOLOv2 was the use of anchor boxes, which helped the model to better detect objects of different shapes and sizes. YOLOv2 also introduced batch normalization, which improved the model's training stability and its ability to generalize on new data.

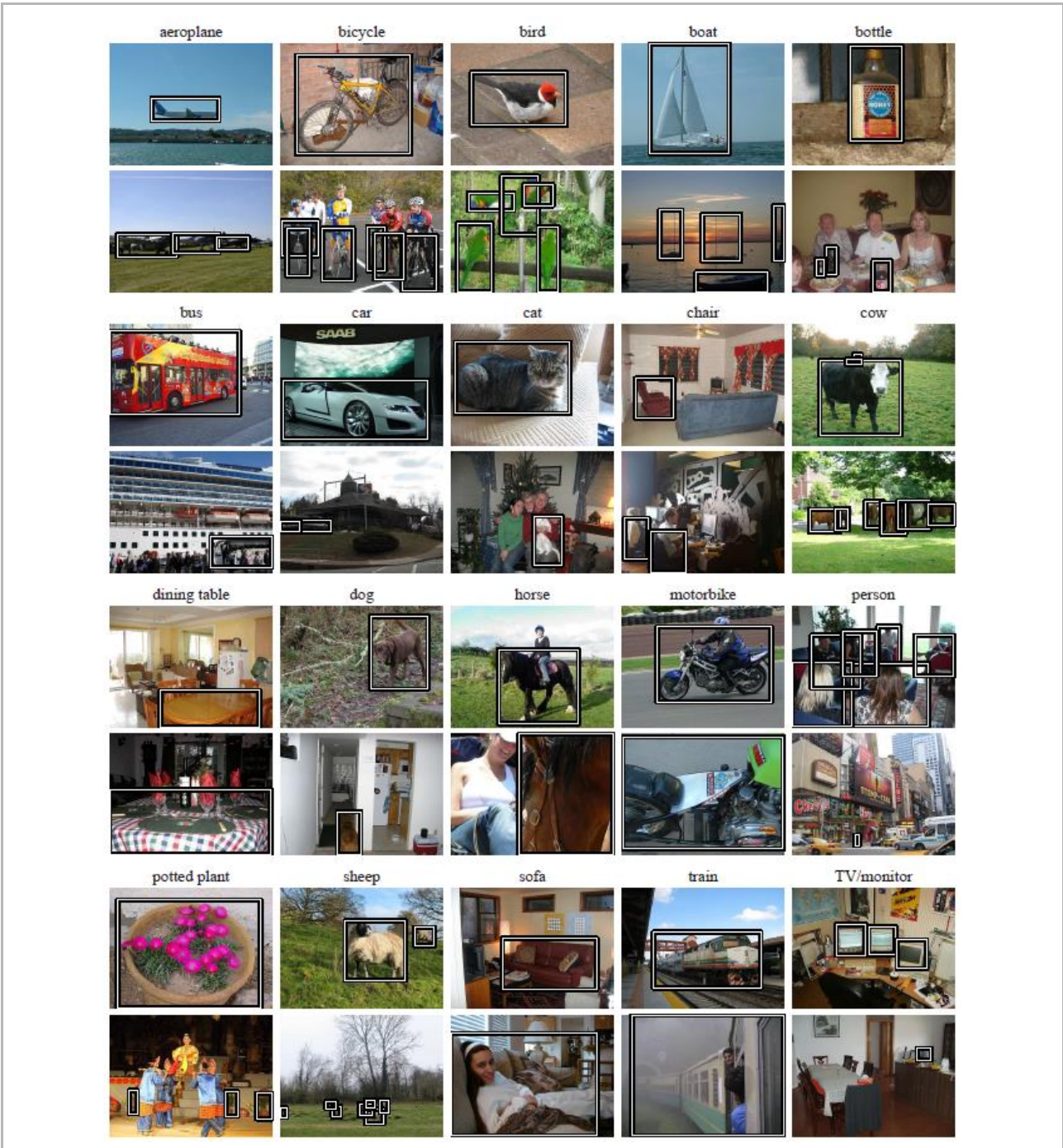


Figure 5. Annotated image sample from the Pascal VOC dataset.

YOLOv3, released in 2018, made further improvements to the original YOLO algorithm. One of the main improvements in YOLOv3 was the use of a new CNN architecture called Darknet-53, which was designed to be more efficient and accurate than the backbone architecture used in the original YOLO and YOLOv2.

Finally, YOLOx is an extension of YOLOv3 that was released in 2021. It was designed to improve the speed and accuracy of YOLOv3, in which the main improvement is represented by the use of feature pyramid networks (FPN); it also restores the anchor-free detection paradigm of the first YOLO versions. Because of the state-of-the-art performance of this model in terms of both accuracy and processing speed, we considered it to be the best choice for this project.

2.4. Object Detection and Classification Models: Free Datasets

This section presents an overview of the 2 most commonly used free datasets for training and testing object detection and classification models: **Pascal VOC** and **COCO**.

Pascal VOC (Visual Object Classes) [4], in its final version, was released in 2012 as a benchmark on which to test participants in the famous challenge of the same name (VOC Challenge). The dataset consists of 9,963 images from a variety of sources, including web searches and photographs taken in natural settings, and each of them is annotated with the location and class of the objects depicted in the image; the total number of annotated objects is 24,640. It contains a total of 20 object classes belonging to 4 macro-categories: people, animals, vehicles and indoor objects. The types of vehicles in the dataset, unfortunately, do not match the needs of the Mimosa project, as it was possible to find only the categories related to *airplane*, *bicycle*, *boat*, *bus*, *car*, *motorcycle*, and *train*.

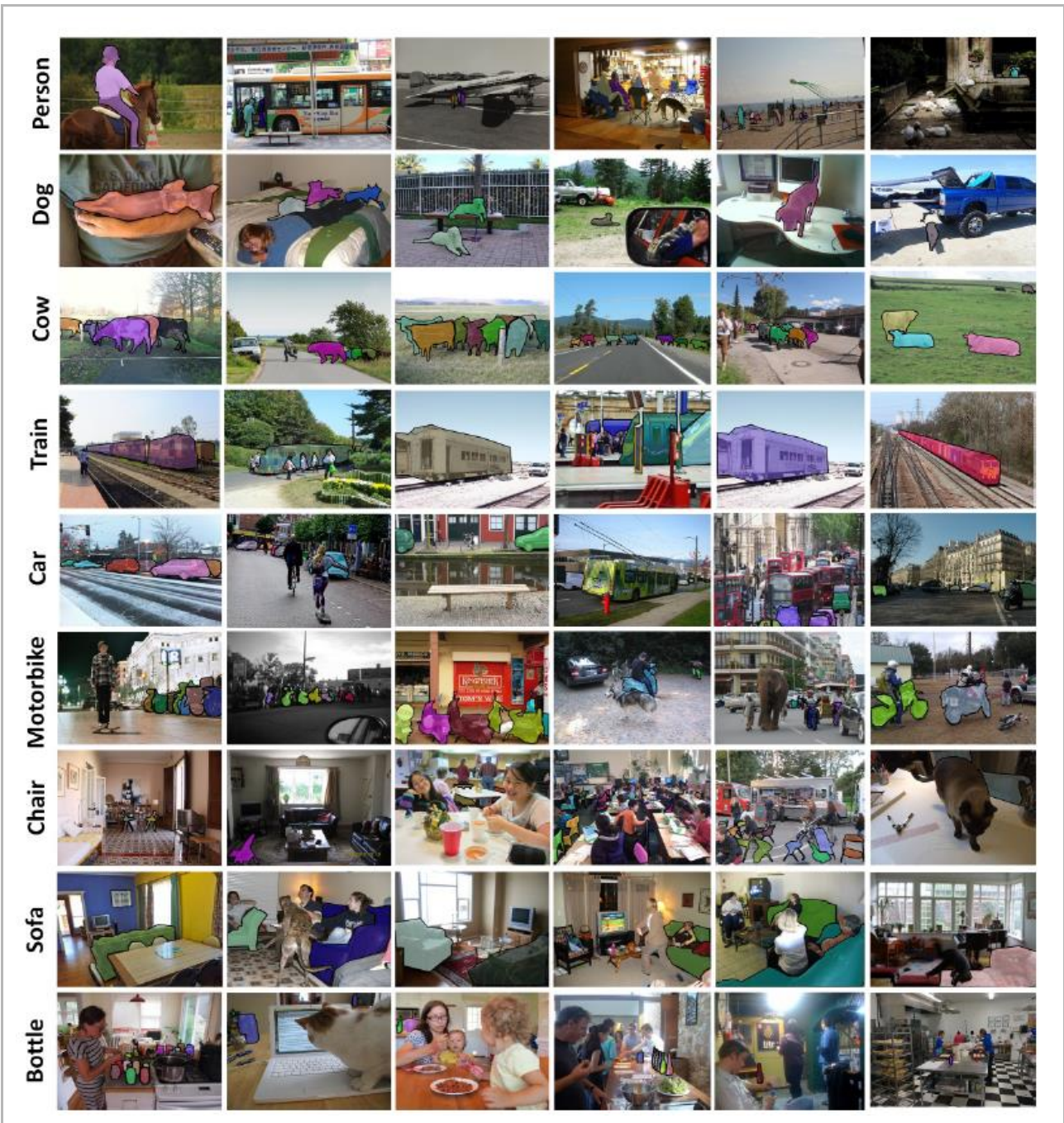


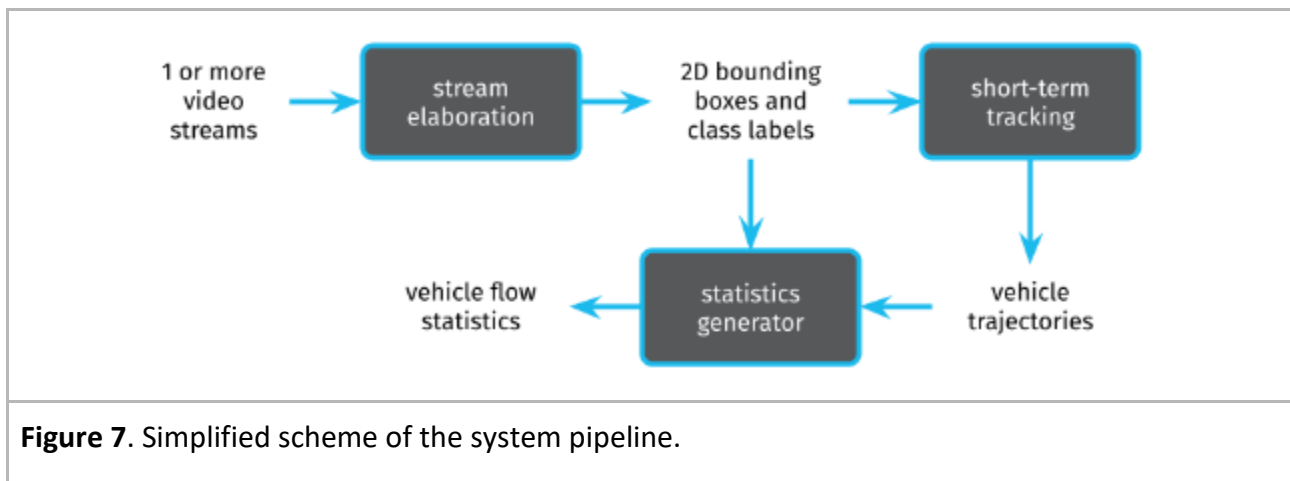
Figure 6. Annotated image sample from the COCO dataset.

Two years later, Pascal VOC was surpassed by what now is probably the most important datasets when it comes to object detection and classification, i.e. *Microsoft Common Objects in COntext*

(COCO) [5], which is also a milestone for other tasks such as segmentation and captioning. It contains over 200,000 images and more than 1.5 million object instances in their natural contexts. This dataset considers 80 different object categories, including animals, people and a variety of vehicle types. The COCO dataset has played a significant role in advancing the field of object detection. It has been used in many research papers and has inspired the development of various object detection algorithms. It has also been used in numerous academic and industry challenges in recent years. For all these reasons, and because it contains almost all the classes of interest for the project, it was selected this dataset for training the model used in the Mimosa project. It was also used the COCO test set to numerically quantify the performance of the model.

3 Methodological Aspects: The Mimosa tool’s three main modules

The vehicular flows analysis system developed in the Mimosa project consists of three main modules: (I) the video stream processing module, (II) the short-term tracking module, and (III) the flow statistics generation module. In this section, it is provided a detailed description of each of these modules, highlighting their inputs and outputs. In this regard, refer to Figure 7 for an overview of the complete pipeline.



3.1. Video Stream Processing Module

The video processing module is a system designed to allow for the offline analysis of one or more video streams. This analysis was conducted by utilizing an object detection and classification

model, specifically the YOLOx model. The purpose of this model is to identify and classify objects within each frame of the collected video stream.

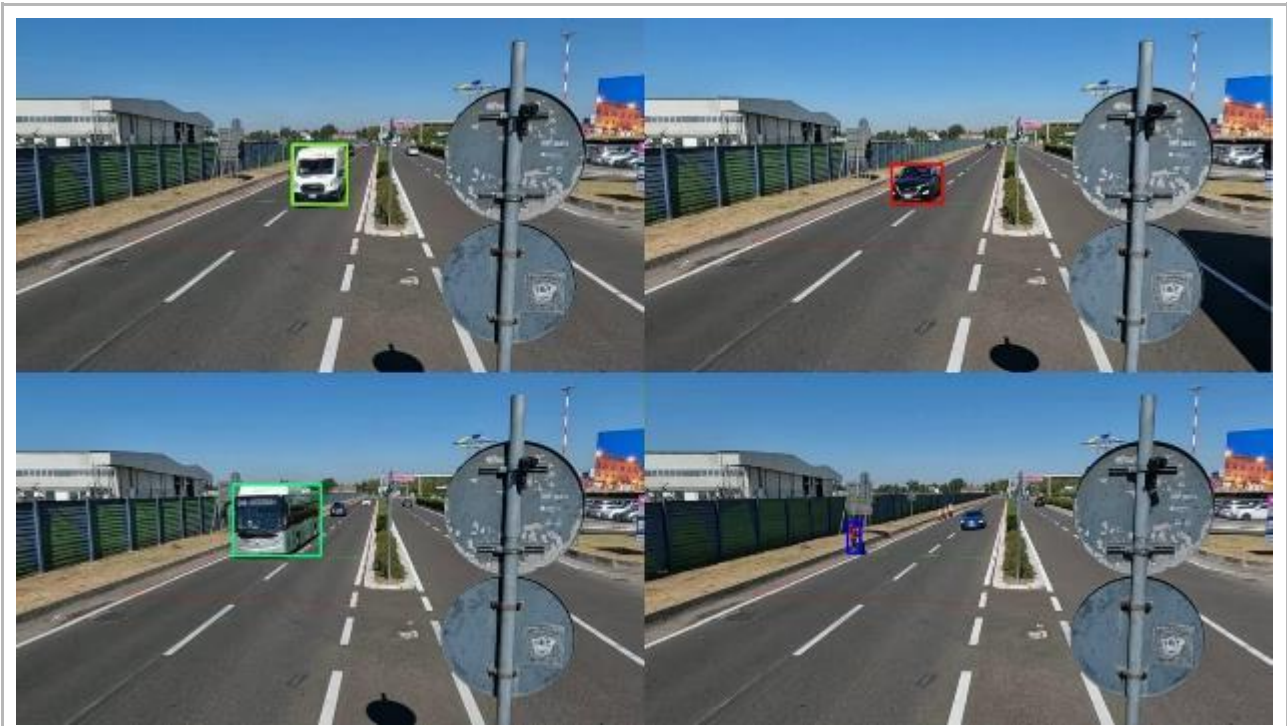


Figure 8. Visual representation of the output of the video stream processing module during the Bologna testing phase. The colours of the bounding boxes encode the class.

To begin the process, each frame of the video stream is normalized and transferred to the graphics processing unit (GPU) for further processing. Once on the GPU, the frame is passed as input to the YOLOx model. The model analyses the input image for all object categories covered by the COCO Dataset (on which the Mimosa model in use was trained). Note that the number of classes covered by the dataset (80 classes) is much larger than those of interest to the Mimosa project, which is limited to the following 7 classes: **(I) light commercial vehicles, (II) heavy commercial vehicles, (III) passenger cars, (IV) motorcycles, (V) buses, (VI) bicycles, and (VII) pedestrians.**

Upon receiving the input frame, the YOLOx model outputs a list of N 2D bounding boxes, where N is the total number of detected objects. Each bounding box represents a rectangular region that encloses a detected object and it is modelled with a tuple of 4 elements $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ where (x_{\min}, y_{\min}) are the pixel coordinates of the top-left corner of that rectangle, and (x_{\max}, y_{\max}) are the coordinates of its bottom-right corner.

For each detection, the model also outputs a unit-sum array of real values within the range of [0, 1]. The array is made of 80 elements, where the i -th element represents the model's confidence in associating the i -th class with the input image. To uniquely identify the class of a given bounding box, we consider the element with the highest confidence value. Any bounding boxes that do not correspond to the object classes of interest for this project are then discarded.

3.2. Short-Term Tracking Module

The short-term tracing module is a crucial component of the Mimosa AI system that enables us to analyse the movement and flow of vehicular traffic. It operates by taking the detection output from the video processing module, which independently analyses each frame of the video stream, and linking them together over time to create time traces. This is necessary in order to extract meaningful statistics about vehicular flows and understand the patterns and trends of traffic within our system.

To understand how this module works, it is first necessary to specify what is meant by the term *tracking* in the context of computer vision. In this context, *tracking* refers to the process of estimating the movement of an object or multiple objects over time in a video sequence. It involves constructing a trajectory, or path, of the object's movement through the video frame, maintaining its identity for the whole video (long-term tracking) or for a short period of time (short-term tracking).

The best-known tracking algorithms exploit association techniques based on motion prediction, achieved by exploiting various techniques and tools well-known in Computer Vision such as the Kalman filter or optical flow. In situations where long-term tracking is necessary, these algorithms may also incorporate appearance-based elements.

For the Mimosa project, we decided to use the **SORT** [6] (**Simple Online and Real-time Tracking**) tracking algorithm. It was presented in 2016 and uses a combination of object detection and data association to track objects in a video sequence. The main idea behind SORT is to use a simple and efficient data association algorithm to track objects in real time. It is designed to be used in scenarios where processing speed plays an important role and where the accuracy in the short term is more important than overall consistency over the entire video.

SORT is designed to work on the output of an object detection algorithm. It then associates the detected objects with tracks using a combination of the Hungarian algorithm and Kalman filtering. The Hungarian algorithm is used to minimize the cost of assigning detected objects to tracks, while Kalman filtering is used to smooth out the trajectory of the tracks over time.

Thanks to the SORT algorithm, the system was then able to assign a unique identifier to each of the detections produced by processing individual frames of the input video, forming short tracks that represent the movement of vehicles or pedestrians within the camera's field of view.

3.3. Statistics Generation Module

Having processed the entire video through the two modules previously described, the Mimosa system will finally proceed to the analysis of vehicular and pedestrian flows taking into consideration the classification of the detected vehicles, as well as their coordinates and trajectories. In particular, the system is able to detect:

- count
- direction of travel
- classification of vehicular type:
 - car
 - bicycle
 - motorcycle
 - pedestrian
 - light commercial vehicles
 - heavy commercial vehicles
- frequency of passage
- average speed

Using the statistical generation module, each of the above statistics can be aggregated by vehicle/pedestrian class, and/or motion direction, and/or road lane depending on the user's input query.

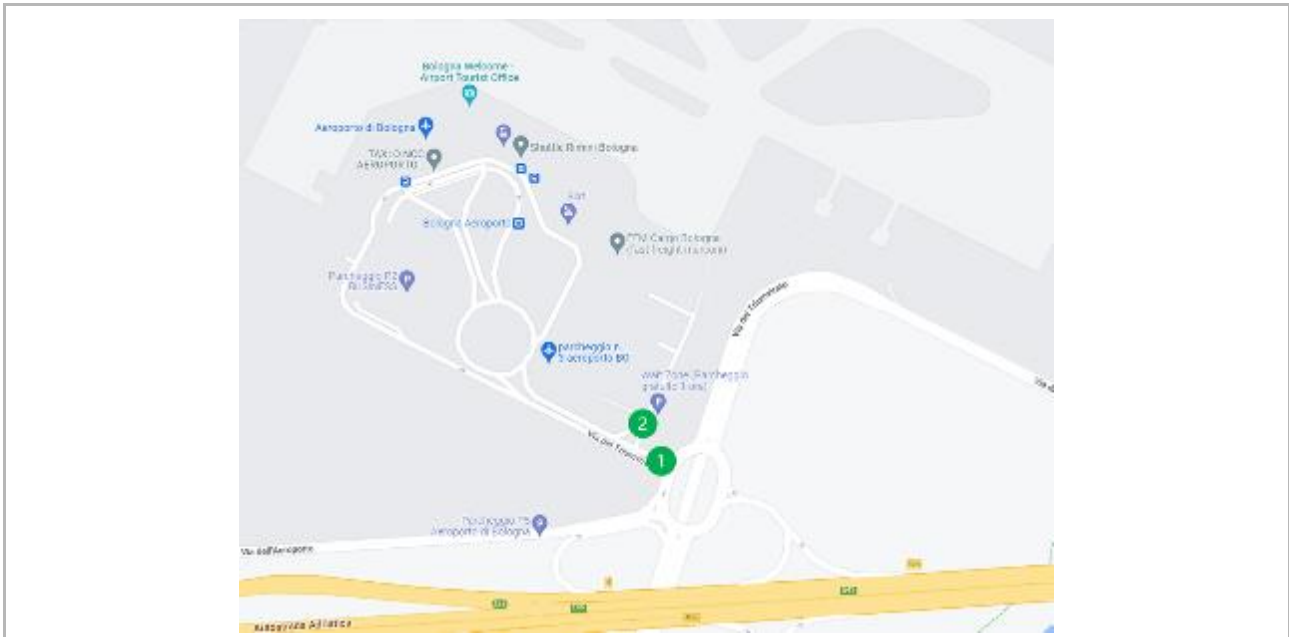


Figure 9. Map highlighting the two locations of interest.

4 Mimosa Testing Activities at the Bologna Airport

The Mimosa testing activities have been carried out at the Bologna Airport. In particular, considering the road accesses to the Bologna airport and in coordination with the Bologna airport key stakeholders, two monitoring locations have been identified:

- Airport main road entrance (point number 1 in Figure 9);
- Airport cargo entrance (point number 2 in Figure 9).




A total of 3 GoPro Hero 10 cameras have been installed in proximity of the predefined locations. More specifically, 2 cameras have been utilized for the main entrance (one for each direction of travel) and 1 for the cargo entrance.

To facilitate installation, the cameras were attached to road sign poles using magnetic supports, as depicted in Figure 10. Each recording session has been supervised by operators in order to prevent thefts of the installed equipment. In fact, the cameras could have easily been removed by passengers as they have to be accessible by the Mimosa operators in order to check battery status and internal SD memory status.




The position and inclination of the camera have been carefully made to maximize lane visibility. As a rule of thumb, a vehicle traveling on the monitored lane should be visible at least at 30 meters from the camera location. It is recommended to place the camera at least 3 meters above the ground. The field of view of each installed camera is reported in Figure 11.



Figure 10. Picture showing the GoPro Hero 10 camera attached to a pole with a magnetic support.

<p>CAM 1</p>	
<p>CAM 2</p>	
<p>CAM 3</p>	
<p>Figure 11. Video frames captured by each installed camera in the project area.</p>	

Below are some quantitative results of the detection and tracking algorithm, referring to CAM 2. As can be seen, the algorithm "hooks" the vehicle as it passes over the flow segment (highlighted in green).

<p>Frame 1</p>	
<p>Frame 2</p>	
<p>Frame 3</p>	
<p>Figure 12. Example of detection performed by Mimosa system.</p>	

During the Mimosa pilot test, two data acquisition sessions were conducted in accordance with the Emilia-Romagna Region and the Bologna Airport key stakeholders. In the following subsections, the statistical results collected using the proposed algorithm are presented.

4.1. First Recording Session

The first recording session took place on **date 03/08/2022**. It started at 09:48 am and ended at 11:08 am. A total of 01h:20m of registration from CAM 1 and CAM 2 has been collected. Recordings from CAM 3 were not successful due to overheating of the equipment, which prevented the correct recording of the video.

From the preliminary data processing, a total of **871** incoming vehicles and people and a total of **937** outgoing vehicles and people were tracked and counted. Table 1 provides results divided by category. Figure 13 and Figure 14 additionally show traffic curves with time steps of 10 minutes.

Type	Entering (CAM 1)	Exiting (CAM 2)
Cars	824	733
Bicycles	4	2
Motorcycles	7	11
Pedestrians	33	48
Light Commercial Vehicles	49	59
Heavy Commercial Vehicles	20	18
Total	937	871

Table 1. Statistic results obtained from the first recording session.

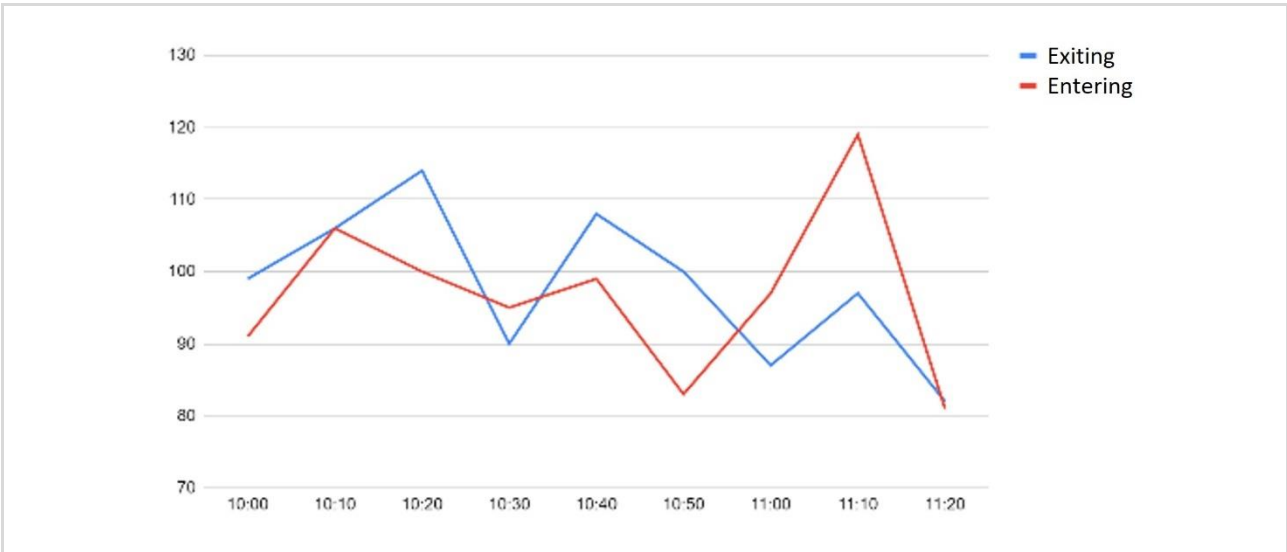


Figure 13. Total traffic curve.

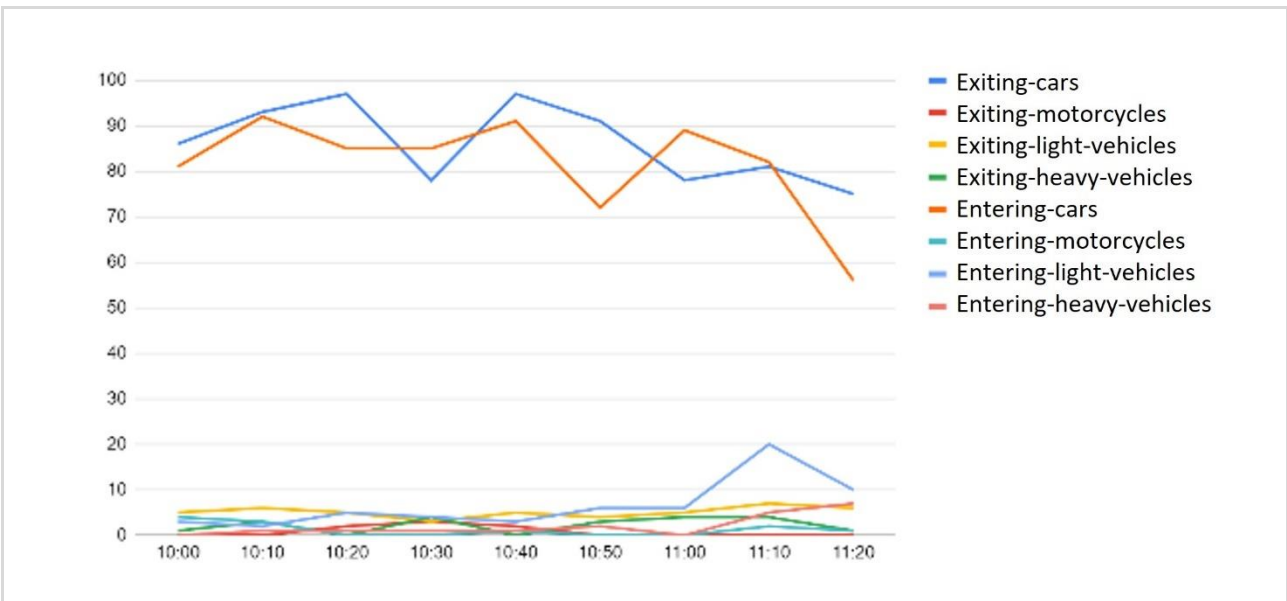


Figure 14. Per-category traffic curve.

4.2. Second Recording Session

The second recording session took place on date **01/11/2022**. It started at 9:30 am and ended at 11:30 am. A total of 120 minutes of registration from CAM 1, CAM 2, and CAM 3 has been collected.

Table 2 provides results divided by category for each of the 3 cameras (CAM 1, CAM 2, CAM 3). Figure 15 and Figure 16 additionally show traffic curves with time steps of 15 minutes for CAM 1 (airport entrance) and CAM 2 (airport exit).

Type	Entering (CAM 1)	Exiting (CAM 2)	Passing (CAM 3)
Cars	874	1055	305
Bicycles	3	0	2
Motorcycles	7	6	1
Pedestrians	11	13	13
Light Commercial Vehicles	146	37	35
Heavy Commercial Vehicles	42	23	8
Total	1083	1134	364

Table 2. Statistic results obtained from the second recording session.

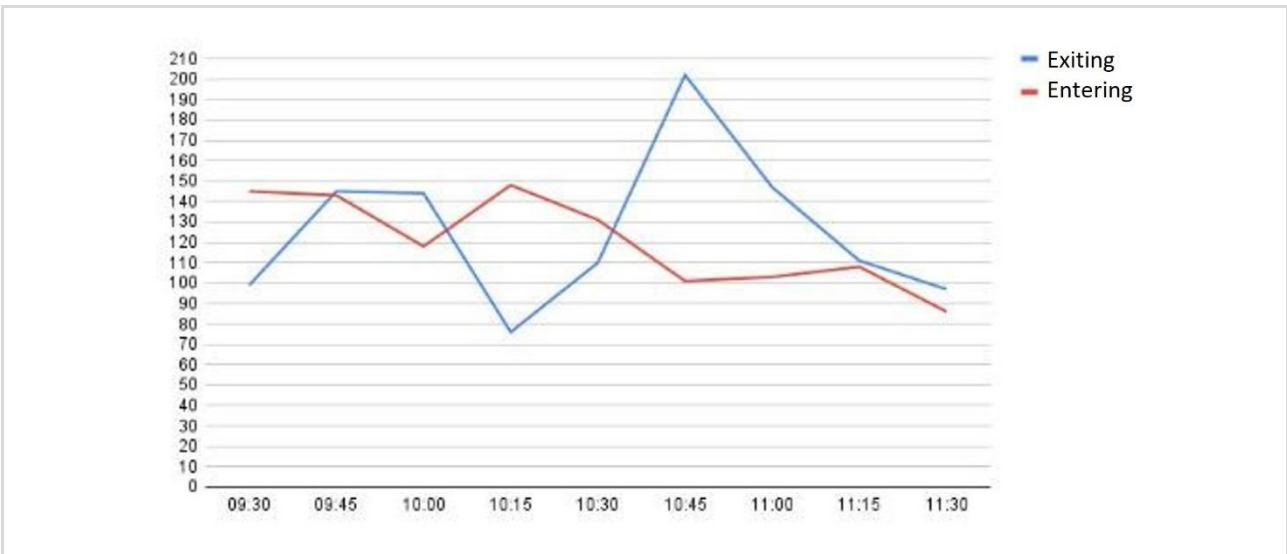


Figure 15. Total traffic curve.

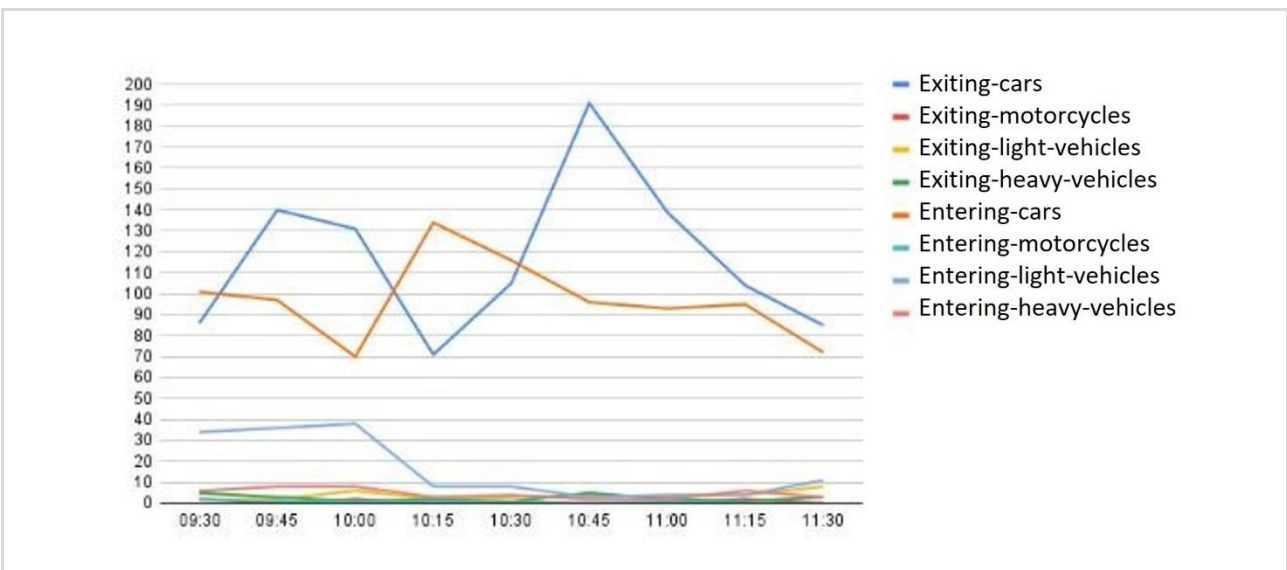


Figure 16. Per-category traffic curve.

The MIMOSA AI solution reliability was tested in an empirical way by selecting some frames of the registered video and checking if the number and typology of the detected vehicles correspond to

the video frame. This human check conducted by Mimosa project staff allowed to detect a complete correspondence among the vehicles detected by humans and by the developed AI solution. Based on this empirical check conducted on videos of both the testing day it was possible to verify the reliability of the Mimosa Ai solution.

More accurate analysis on AI performance require an annotation phase where the recorded videos are manually labelled by human operators. In this way it is possible to compare the performance of humans with the performance of the AI algorithm. However, it is important to note that this procedure is time consuming and error prone. A better solution requires the use of reference benchmarks currently available in literature. The more challenging benchmark for object recognition currently available is the COCO dataset. Compared to all the others opensource algorithms, YOLOx has the best performance in terms of Average Precision while maintaining a good trade of between accuracy and speed.

In relation to the video collection procedures, the general conditions for high confidence of the whole system are: good lighting conditions, absence of occluding objects, good camera positioning facing the incoming traffic (at least at 3 meters above ground).

5 Problems and Potential Solutions of the MIMOSA AI solution

The main problems in the artificial intelligence application on traffic flow analysis are:

- Recorded videos can be considered sensitive under the EU's data protection law.
- Overheating of the equipment during hot seasons.
- The precariousness of the installed cameras.
- Short recording period due to action camera memory size and battery life.
- Bad camera positioning.

The main problems in the utilization of an opensource artificial intelligence technology for traffic analysis are:

- Limited number of vehicle categories available (this limit was strictly related to the utilization of an opensource technology. In the market there are dedicated proprietary softwares where the artificial intelligence technology is trained for the recognition of an high number of vehicles' typologies);
- Requires dedicated and expensive hardware.

Potential technological and organization solutions are:

- Privacy can be handled by displaying dedicated video surveillance warning signs that report GDPR General Data Protection Regulation (EU) 2016/679 with guidelines 3/2019 on the processing of personal data through video devices by the European Data Protection Board ([LINK](#)). The warning signs must be positioned in such a way as to be clearly visible before the interested party can enter the area. Along with the warning sign, a GDPR Second Layer Information document must be accessible from the internet, possibly via a QR code ([LINK](#)).
- Overheating and precariousness of the equipment could be solved by installing dedicated video cameras on-site or using preinstalled cameras for outdoor applications.
- The short recording period could be solved using action cameras with increased battery life and by expanding the memory with SD cards of at least 256 GB.
- Camera positioning should be made according to the following specifications:
 - Cameras must face the incoming traffic (monitored vehicles move towards the camera).
 - A vehicle traveling on the monitored lane must be visible at least 30 meters from the camera location.
 - The camera should be placed at least 3 meters above the ground and less than 6 meters
- Limitations about the open-source nature of the tool can be solved by purchasing ad hoc software from a consultancy software house with Artificial Intelligence and Computer Vision knowledge.

6 Conclusion and Recommendations

During this action, Fondazione ITL in collaboration with the external technical experts GoatAI SRL, developed a tool for automatic traffic flow estimation based on Artificial Intelligence and Computer Vision able to (i) count the number of vehicles and people, (ii) classify the types of vehicles and (iii) extract aggregate statistics. The aim of this project was to provide the decision-makers with a new tool for traffic monitoring and data-oriented decisions making on the topic of sustainable transport promotion. The analysis of vehicular flows is an important aspect of transportation engineering, as it helps to understand the movement patterns of vehicles on roads and highways. This information can be used, for instance, to optimize traffic signals, improve road design, and reduce congestion.

The proposed software has been tested at the Bologna Airport on two separate days. During the tests, a total of 02h:10m has been recorded from three different cameras. The collected videos have been elaborated by the proposed tool. Videos have been promptly deleted after computation to be compliant with privacy regulations.

The data extracted during experimentation is intended to provide insights into urban mobility planning in a cost-efficient way. The solution has been designed to be replicable in multiple contexts where a camera can be easily placed in an elevated position above traffic flows. The software has been open-sourced under MIT license and can be downloaded on GitHub (<https://github.com/ITLBologna/Fluxus-AI>).

In order to replicate the tests in other contexts, it is necessary to formulate a series of recommendations:

- Cameras should be positioned to face the incoming traffic (monitored vehicles should move towards the camera).
- A vehicle traveling on the monitored lane should be visible at least 30 meters from the camera location.
- The camera should be placed at least 3 meters above the ground. The defined position should not exceed 6 meters of height.
- For long recordings that should last for more than 3 hours, we advise to employ preinstalled cameras, as the battery life of action cameras is usually in the range of 100-200 minutes.

The AI solution reliability has been measured on the most challenging reference benchmark namely COCO dataset. The good performance obtained by the used AI solution show good generalization capability and demonstrates that the system can be employed in different contexts.

7 References

- [1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
- [2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Proceedings of the IEEE International Conference on Computer Vision (2015).
- [3] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun. YOLOX: Exceeding YOLO Series in 2021. arXiv (2021)
- [4] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision (2010).
- [5] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. Microsoft coco: Common objects in context. European Conference on Computer Vision (2014)
- [6] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft. Simple Online and Realtime Tracking. IEEE International Conference on Image Processing (2016)

Annex 1. The Mimosa tool: “Fluxus-AI” App. Guide for the utilization

In this section, we present the Fluxus-AI application, which serves as a convenient and user-friendly interface for the integration of the various modules of the vehicle flow analysis system. The Fluxus-AI” App is the Mimosa user-friendly interface for the utilization of all the mentioned modules.

To use the Mimosa Fluxus-AI application, you first need to properly prepare the videos to be processed. A recording session may consist of several video files, which must be grouped together and placed in an appropriate directory. Consider the example in Figure 17, in which we have two video files representing 2 consecutive recording sessions of the same area, namely **video_01.MP4** and **video_02.MP4**, both inside a directory named **fluxus_ai_demo**. To ensure the system operates correctly, it is recommended that each video be given a descriptive name using a common prefix (“**video**” in our example) followed by a numerical suffix to indicate the time ordering (“**_01**” and “**_02**” in our example)”. This will help to clearly identify and organize the videos within the system. Once you have populated the directory with the videos you want to analyze, you can open the Mimosa application and select this directory for processing. At this point, the calibration procedure will be automatically started. Specifically, the system will present the user with a frame of the area to be analyzed and request to select the flow line (see Figure 18), which is the imaginary line on the road that will be used to extract statistics. The statistics will always refer to the time when a vehicle crosses the flow line. Then, the user must delimit the analysis area to a specific portion of the image by selecting an appropriate polygon containing the flow line. By excluding areas that are irrelevant to the analysis, you can speed up video processing and makes it more efficient.

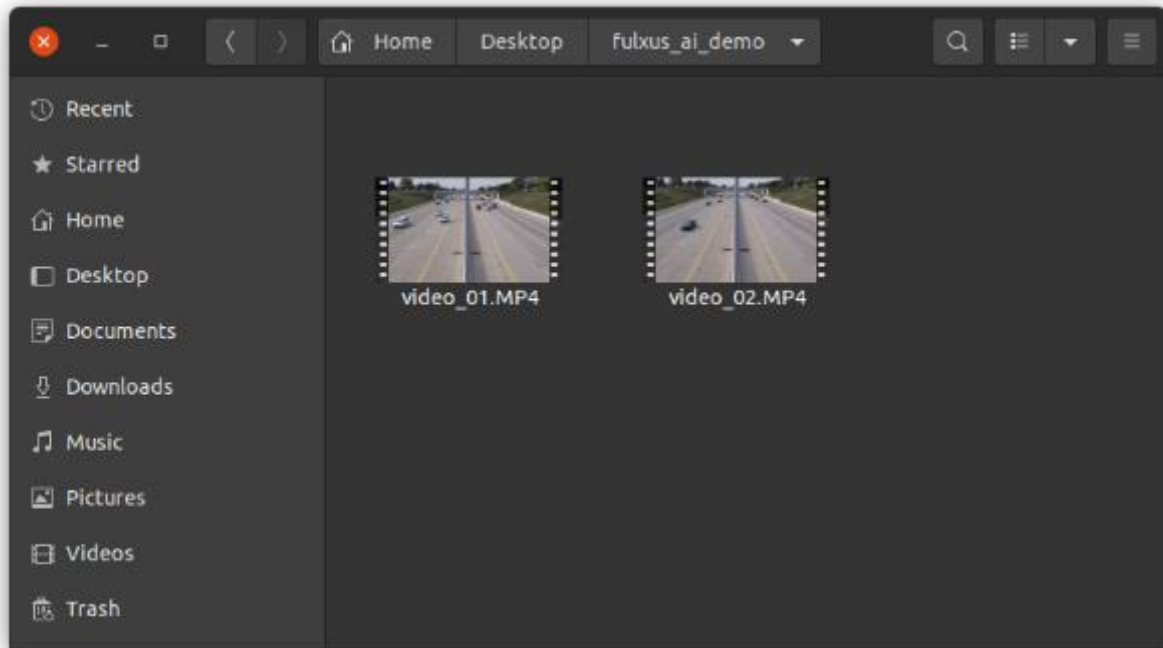


Figure 17. Fluxus-AI app - first step: data preparation.

The system will now begin processing the videos using the methods outlined in previous sections. The results of this processing will be saved in the previously designated directory and will consist of three files: **full_video.trk**, **full_video.dat**, and **data.csv**. The first two files are binary files used exclusively by the Fluxus-AI application and contain the detections and labels extracted with YOLOx and the tracks obtained through the short-term tracking module. The third file, **data.csv**, is a standard CSV file that can be opened and analysed with any application that supports this format, such as Microsoft Excel.

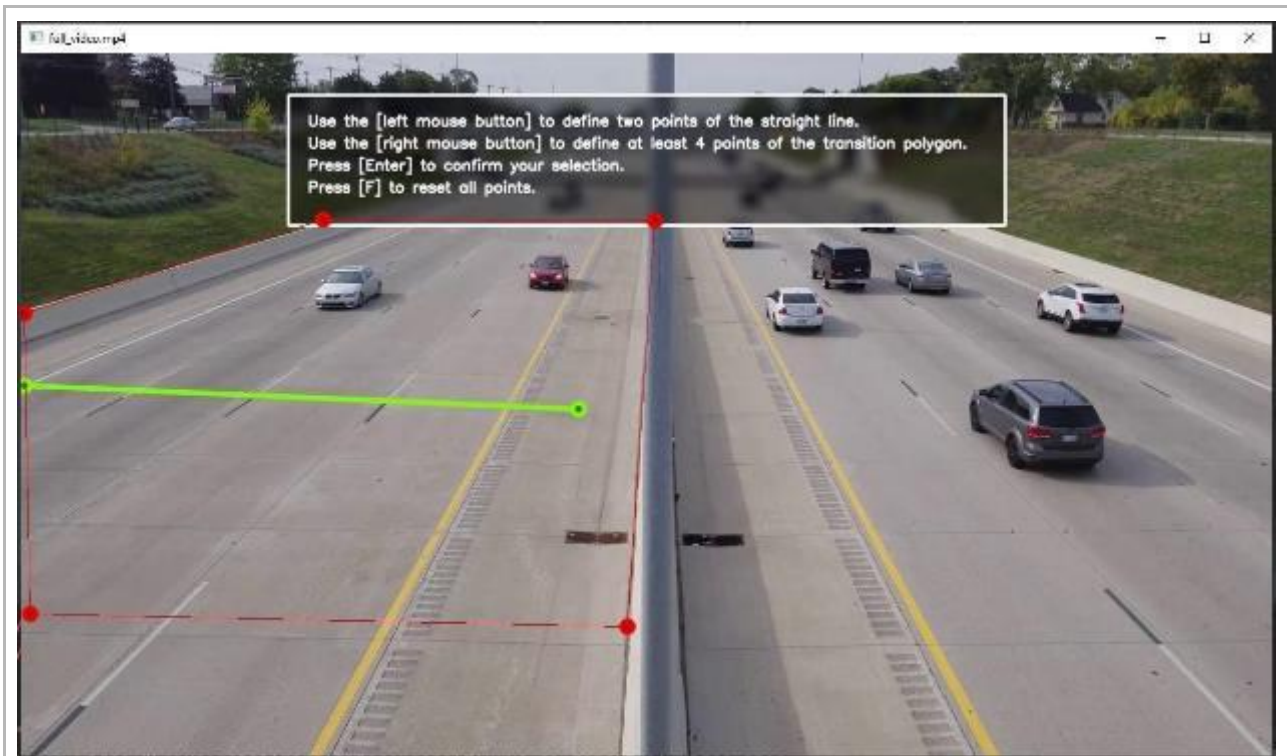
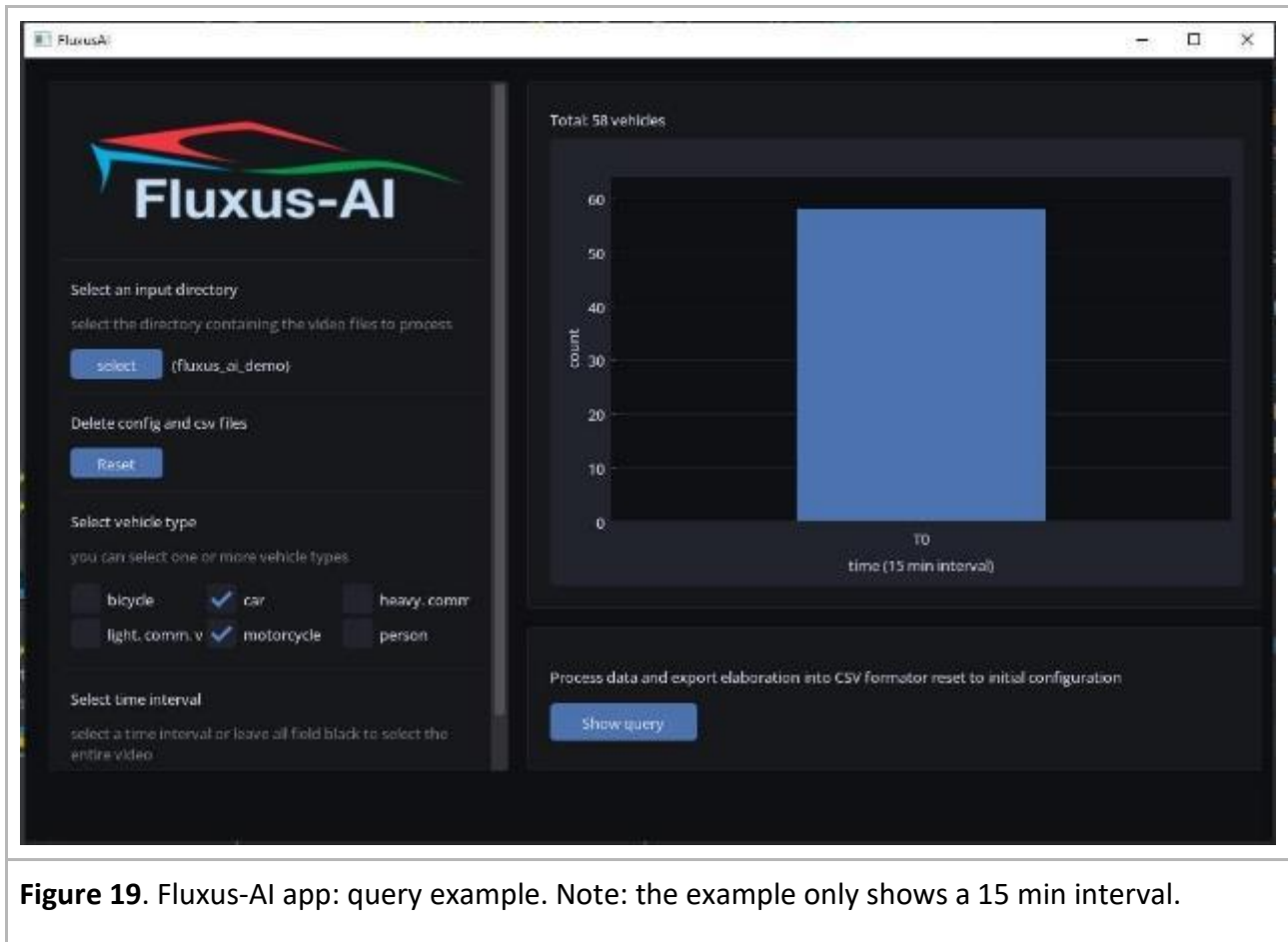


Figure 18. Fluxus-AI app: flow line selection interface.

Note that, if you select a directory that already contains the [data.csv](#) file, you do not need to repeat either the streamline selection step or the video processing step.

The Fluxus-AI application is able to display the aggregate statistics based on the queries specified by the user, who can filter the data using numerous parameters including vehicle classes and time intervals. In this regard, refer to Figure 19, in which an example of a query is shown, in which the requests to view the vehicular flow in the first 15 minutes of video, considering only vehicles belonging to the class "car" and the class "motorcycle."



Fluxus-AI Installation guide

In this section, we will provide additional details about software installation. A beta version of Fluxus AI is available on an online repository on GitHub (<https://github.com/ITLBologna/Fluxus-AI>). The repository contains the source code of the application previously described. To install the software please follow the following instructions:

- Install Anaconda (<https://www.anaconda.com/products/distribution>) as the Python interpreter
- Install Git (<https://gitforwindows.org/>) as the version control software
- Open the Anaconda Prompt command line and create new "mimosa" environment: ***conda create -n mimosa python=3.9***
- Activate the environment: ***source activate mimosa***
- Clone this repository: ***git clone https://github.com/berserkrambo/mimosa.git***
- Change dir to root project: ***cd mimosa***

- Install the requirements: **`pip install -r requirements.txt`**
- Install PyTorch: **`pip install torch torchvision --extra-index-url https://download.pytorch.org/whl/cu116`**
- Run the app: **`python entry_point.py`**
- You can also run the application by double-clicking FluxusAI.bat from the project root folder.

The instruction on how to use the software has been reported in the previous section and are available on the GitHub repository.