TEKNOFEST

AI in Transportation Competition

Preliminary Design Report

Team name: SEE AI

Atılım University – Undergraduate project, supported by the ATAK programme

English version of the original Turkish submission. The content has been translated; figures that contained non-English text were redrawn.

List of Figures

Figure 1: History of the YOLO algorithms

Figure 2: Performance of the YOLO algorithms on the MS COCO dataset

Figure 3: Flowchart

Figure 4: Photograph generated by the DALL·E application

Figure 5: Photograph generated by the DALL·E application

Figures 6–9: Sample photographs with different characteristics from our dataset

List of Tables

Table 1: Project timeline

Table 2: Results obtained with the YOLO models

Abbreviations

ATAK – Atılım University Student Research Career Support Programme

SAHI – Slicing Aided Hyper Inference

YOLO – You Only Look Once

GELAN – Generalized Efficient Layer Aggregation Network

PGI – Programmable Gradient Information

GPS – Global Positioning System

SLAM – Simultaneous Localization and Mapping

1. Team Structure (5 points)

SEE AI	
Member 1	**Captain**
• Dataset creation • Data analysis and organization • Data labelling • Algorithm research and analysis	• Algorithm research and analysis • Model testing and development • Data analysis and organization • Data labelling

2. Project Status Assessment (10 points)

After deciding to take part in the competition, we quickly researched algorithms and open-source datasets. We applied to the ATAK support programme offered at our university to help students prepare for Teknofest and similar competitions. Within the scope of the support programme, a drone and other third-party services (Google Colab, Google Drive, Roboflow) were purchased in order to produce an original dataset that meets the competition requirements. We researched different artificial intelligence algorithms and carried out a series of procedures to train various models and test their results. Among the compared models, the one with the highest metrics and performance was identified as the YOLOv8 model. However, the new YOLOv9 algorithm, released in February, was trained on the same datasets and compared with the YOLOv8 model. In addition to the datasets we already had, the dataset that we had started to create ourselves was expanded. The continuously updated datasets were used in model training, and the differences between the models began to be identified.

For the second task, research started on algorithms that perform position estimation and detect displacement. For position estimation without a GPS system, information began to be collected on the Kalman filter and visual odometry algorithms.

3. Algorithms and System Architecture (25 points)

3.1 Datasets (10 points)

- The dataset of the teams that placed first and second in the 2022 AI in Transportation Competition
- Our own dataset
- Datasets from previous years of the competition, shared by the competition organizers
- A dataset generated with artificial intelligence

We use the labelled dataset of the teams that placed first and second in the 2022 Teknofest AI in Transportation Competition, which we consider a major advantage. In total, this dataset contains nearly 25,000 labels of aerial-view people, vehicles and designated landing areas. We use it because its content is highly varied and its video angle and height match the competition format. In addition, we are building our own dataset on our university campus. With it we obtain videos and images that are very similar to the sample videos provided by the competition team. We also have a large quantity and variety of data because the competition team shared the datasets used in previous years. Finally, we have nearly 500 images generated with various AI applications, as described in detail in the Originality section.

3.2 Algorithms (10 points)

YOLO

YOLO is an object detection algorithm designed to detect objects in a single stage. YOLOv9 is the latest version of the YOLO algorithms, which were first introduced in 2015. This version was introduced by Chien-Yao Wang and his team in the first months of 2024. It introduces innovative methods such as Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN) to address problems of information loss and computational efficiency effectively.

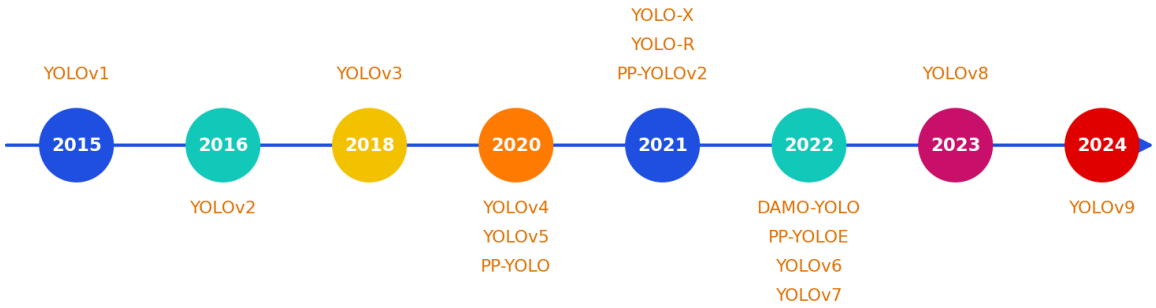


Figure 1: History of the YOLO algorithms

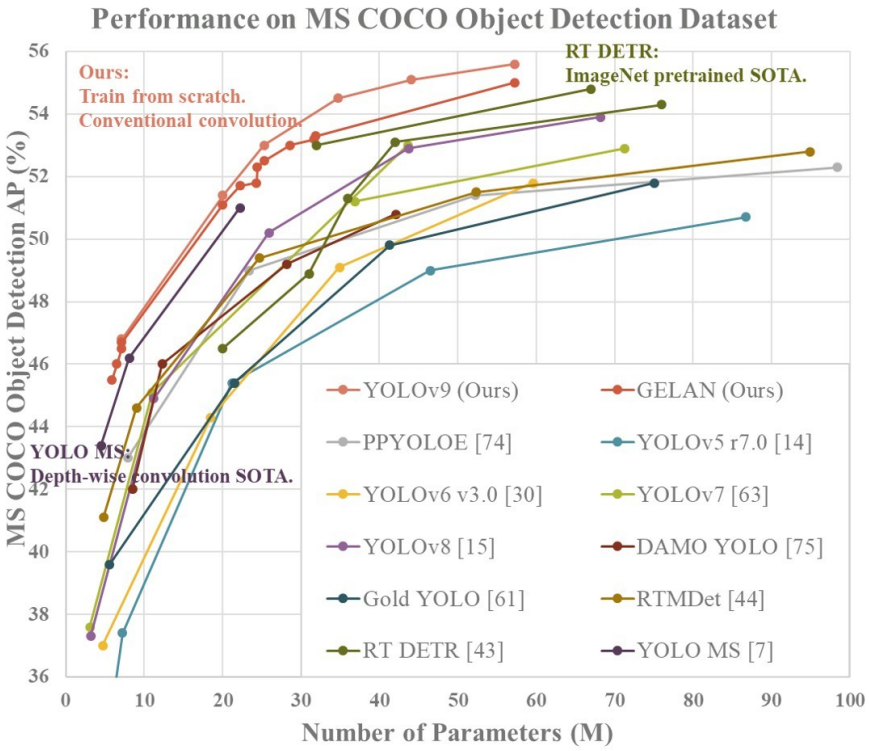


Figure 2: Performance of the YOLO algorithms on the MS COCO dataset (source: Wang et al., 2024 [2])

SAHI

To increase our success rate in detecting small objects, we decided to use the SAHI algorithm. SAHI splits a large or high-resolution image into smaller pieces (slices), runs object detection on these slices, and then merges the slices to reconstruct the object locations on the original image. SAHI is an innovative library designed to optimize object detection algorithms for large-scale, high-resolution images. Its main function is to divide images into manageable slices, run detection on each slice and combine the results. SAHI is compatible with a range of object detection models, including the YOLO series, which gives flexibility while allowing computing resources to be used efficiently. Our reasons for using SAHI are:

- **Easy integration:** SAHI integrates easily with YOLO models, so slicing and detection can be started without many code changes.
- **Resource efficiency:** by splitting large images into smaller pieces, SAHI optimizes memory use and allows high-quality detection on hardware with limited resources.
- **High accuracy:** SAHI preserves detection accuracy by using intelligent algorithms to merge overlapping detection boxes during the stitching step.

Visual odometry

Visual odometry is a technique that uses camera images to estimate the motion of a camera and to build a three-dimensional map of its surroundings. This technology is critical for robots, autonomous vehicles, and augmented and virtual reality applications that need to track their position and interact with their environment. Its main steps are:

- **Feature extraction:** distinctive features such as corners, edges and texture patterns are extracted from each image.
- **Motion detection:** features in consecutive images are compared and the differences are determined. These differences may indicate movement of objects, movement of the camera, or changes in the environment.
- **Motion estimation:** using the geometric relationships between matched features, how much and in which direction the camera has moved is estimated.
- **Position tracking:** the position and orientation of the drone are estimated using the motion of the camera and the three-dimensional map of the environment.

3.3 Flowchart (5 points)

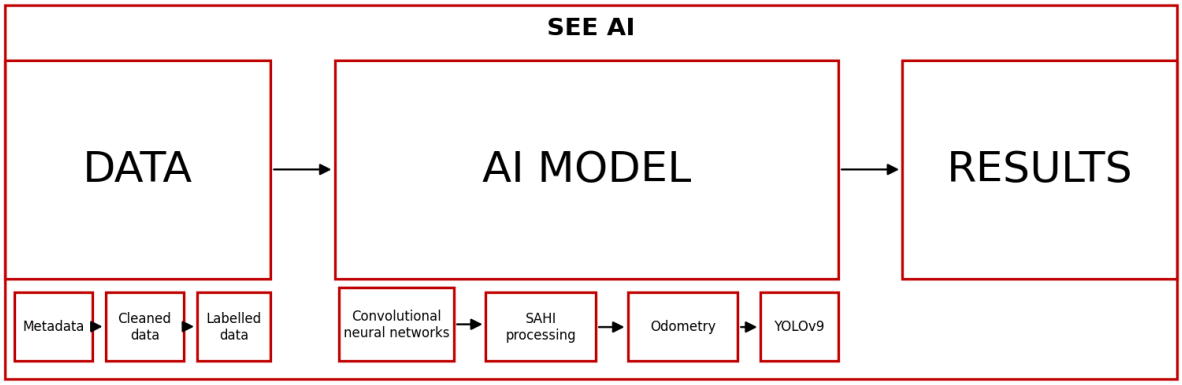


Figure 3: Flowchart

4. Originality (20 points)

As a team, we spent time exploring various AI tools and applications. As a result, we found some AI applications that can produce photographs and videos in the required format. In our trials, we found OpenAI's DALL·E application very successful. By giving it the inputs we wanted, we had it generate data for us, and in this way we obtained a dataset of 500 photographs. Figure 4 and Figure 5 were produced with this application.



Figure 4: Photograph generated by the DALL·E application



Figure 5: Photograph generated by the DALL·E application

By shooting on our university campus, we created our own dataset. It is very similar to the datasets shared by the competition team. Using some applications, we adjusted the datasets to add contrast, brightness and richness. Because the SAHI algorithm is adapted specifically for YOLOv8 and earlier models, it had not yet been adapted to YOLOv9. With algorithms that we designed ourselves, we integrated SAHI into YOLOv9.

For the second task, we carried out extensive research to find photographs and videos that contain location information. We found that there were not many suitable datasets with these features. Using the Google Earth application, we created synthetic videos together with location information.

5. Project Timeline (10 points)

Table 1 shows the work done and planned, by month.

	Requirements	Feb	Mar	Apr	May	Jun	Jul	Aug
1	Dataset creation	●	●	●				
2	Data labelling		●	●	●			
3	Algorithm research and development	●	●	●	●	●	●	
4	Model validation and testing			●	●	●	●	
5	Model improvement			●	●	●	●	●

Table 1: Project timeline

6. Results and Review (25 points)

The results showed that the YOLOv9 algorithm performed better than the YOLOv8 algorithm. The integration of the SAHI algorithm also improved detection performance significantly.

The YOLOv8 and YOLOv9 algorithms were first tested separately without any additional algorithm, so that the baseline performance of each algorithm was evaluated without SAHI. The tests were then repeated after adding the SAHI algorithm to the YOLOv8 and YOLOv9 models. With these tests, we evaluated how SAHI affects the performance of each model.

Model / Variables	Data	FLOPs	mAP [.5:.95]
YOLOv8	5000	247.3	55%

Model / Variables	Data	FLOPs	mAP [.5:.95]
YOLOv8 + SAHI	5000	259.7	64.5%
YOLOv9	5000	326.1	69.4%
YOLOv9 + SAHI	5000	356.4	73.2%

Table 2: Results obtained with the YOLO models

For model training, the data was divided into three groups:

- **Training (60%):** the main dataset used to optimize the model parameters.
- **Test (30%):** the dataset used to evaluate the performance of the model after training.
- **Validation (10%):** the dataset used to optimize the hyperparameters of the model.

This three-way split helps prevent overfitting and tests the generalization ability of the model.

In addition, model training was carried out with datasets containing different weather conditions. In this way, the system was enabled to detect objects correctly in all kinds of weather. Some photographs that exemplify the tested weather conditions are shown below:



Figure 6



Figure 7

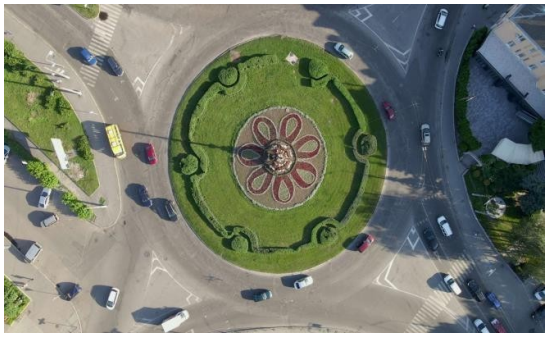


Figure 8

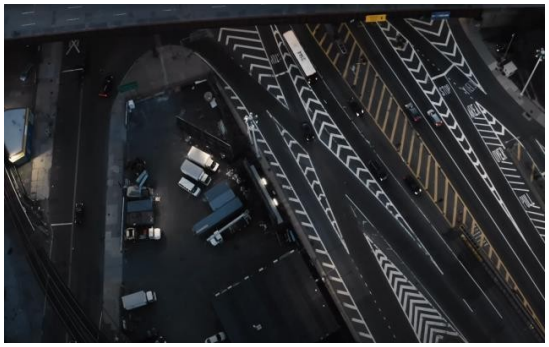


Figure 9

We are still working on the visual odometry technique that we will use in the second task. We continue to carry out detailed research on visual odometry, SLAM and Kalman filters, and we are working on integrating these methods with one another.

Problems and Errors

Some problems and errors were encountered during the development of the system:

- Imbalanced distribution of objects in the dataset
- Long model training times
- Complexity of integrating the SAHI algorithm

- Human shadows or small bushes being detected as people

These problems were solved using different techniques. For example, the imbalanced distribution of objects in the dataset was balanced by supplementing it with our own dataset. The long training time was addressed by using more powerful hardware and parallel processing techniques. The complexity of integrating SAHI was overcome by studying the detailed documentation and applying our coding knowledge. The problem of human shadows or small bushes being detected as people was reduced to some extent by using the SAHI algorithm.

Lessons Learned

In this project we gained experience in:

- Using the YOLOv8 and YOLOv9 algorithms for object detection in drone images.
- Using the SAHI algorithm to improve object detection performance.
- Overcoming the difficulties of processing large datasets and training complex models.

7. References (5 points)

[1] F. C. Akyon, S. O. Altinuc and A. Temizel, "Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection," 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 966–970, doi:10.1109/ICIP46576.2022.9897990.

[2] C.-Y. Wang, I.-H. Yeh and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," arXiv:2402.13616, 2024. https://doi.org/10.48550/arXiv.2402.13616

[3] 2022 competition datasets, obtained from https://github.com/sezer-muhammed/Teknofest-Ulasimda-Yapay-Zeka-Veri-Seti

[4] Ultralytics, YOLOv8. https://github.com/ultralytics/ultralytics

[5] J. Wei and A. Yilmaz, "A Visual Odometry Pipeline for Real-Time UAS Geopositioning," Drones, 2023, 7, 569. https://doi.org/10.3390/drones7090569

[6] M. O. A. Aqel, M. H. Marhaban, M. I. Saripan and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," DOI 10.1186/s40064-016-3573-7.