BILKENT UNIVERSITY

CS491 - PROJECT SPECIFICATION DOCUMENT

Berkin Kağan Ateş	22102519
Burak Baştuğ	22102266
Umut Başar Demir	22102376
Berin Su İyici	22102342
Arda Öztürk	22103616

Table of Contents

1. Introduction

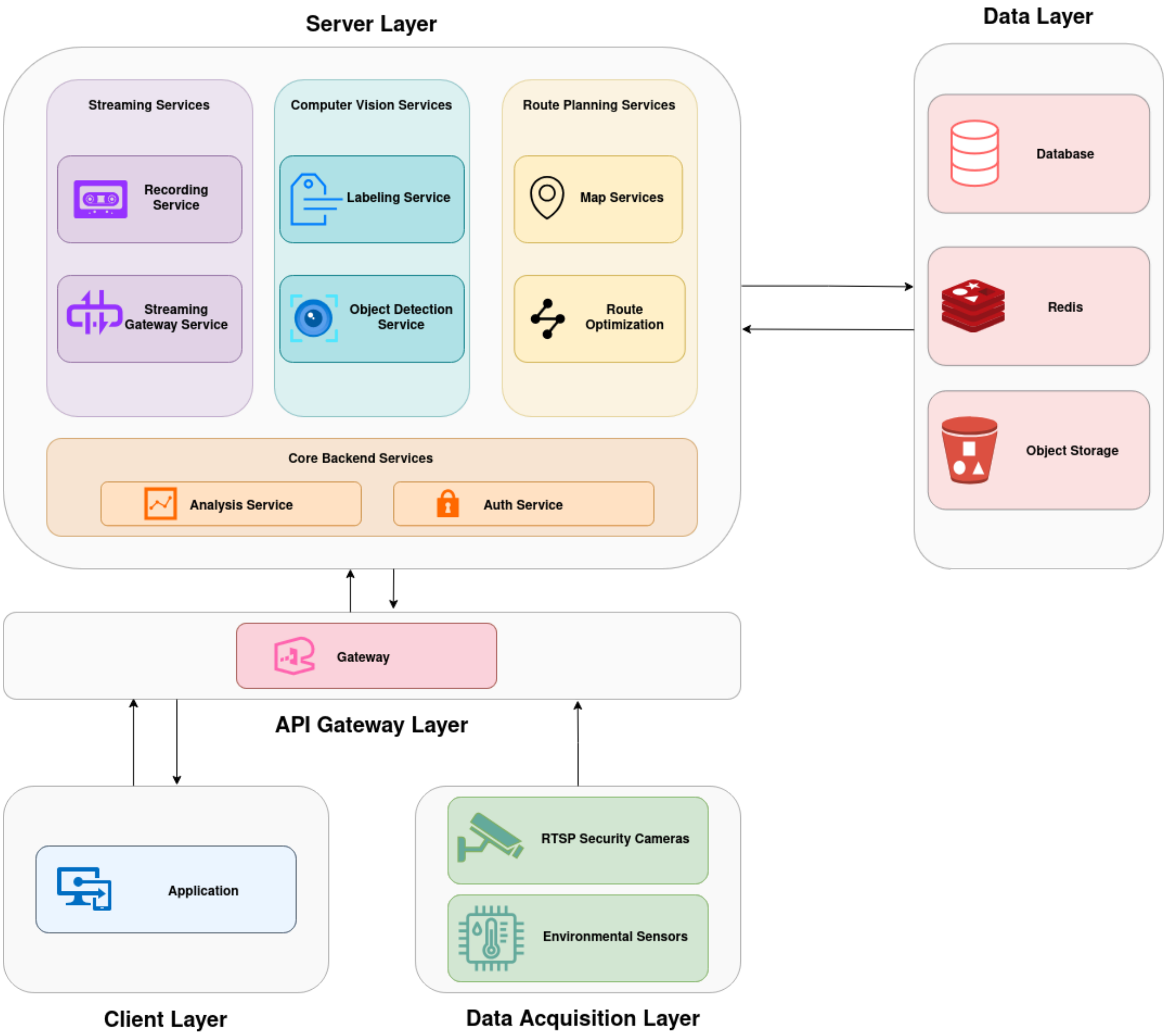
1.1 Description

The logistics industry is a dynamic and rapidly growing sector composed of various elements, including warehouses, trucks, loads, routes, time constraints, and cost considerations. Since this industry is inherently complex, managing all these elements may require assistance from technology. Based on this idea, TırGöz offers a user-friendly system for logistics industry companies by increasing the efficiency of logistics operations with the help of computer vision technology and route optimization algorithms.

TırGöz aims to develop a logistics monitoring and optimization platform that integrates computer vision and vehicle routing optimization to enhance efficiency in loading/unloading and transportation processes. The main functionalities of TırGöz can be divided into three modules: streaming, detection, and route optimization. The streaming module demonstrates the process of loading/unloading the truck in real-time, and the detection module utilizes computer vision to detect item counts and anomalies from this stream, such as misplaced or damaged items. The route optimization module utilizes the item count information in trucks/warehouses, along with other constraints such as time and cost, to calculate the optimal routes for the trucks in the system. This is how TırGöz aims to simplify and expedite logistics operations.

When the value and impact are considered, TırGöz proposes a system that provides a holistic view of logistics operations by merging perception and planning layers. TırGöz provides operational efficiency by reducing manual inspection time and optimizing manpower usage. TırGöz offers cost reduction by minimizing late deliveries and unnecessary travel through adaptive route planning. TırGöz offers work quality by preventing shipment mistakes with the generation of real-time alerts. TırGöz also considers the scalability by ensuring low latency and dynamic route optimization even with a large number of trucks and warehouses.

1.2 High-Level System Architecture & Components of Proposed Solution



1.2.1 Data Acquisition Layer

This project can be classified as an IoT-driven system, integrating hardware components commonly found in warehouse environments. Since the backend services rely on the data produced by these devices, establishing reliable communication protocols and software integrations is essential for ensuring stable and accurate data flow.

1.2.1.1 RTSP Security Cameras

Standard security cameras that provide RTSP streams are planned for use. Since these types of cameras are already available in many warehouses, we can avoid additional costs for clients by utilizing these existing cameras.

1.2.1.2 Environmental Sensors

These sensors will not be the main actors like security cameras. Their usage will be determined according to the clients we maintain contact with. Their biggest advantage will be determining which goods are transferred to the trucks and synchronizing the amount of goods in the warehouse and the trucks.

1.2.2 Client Layer

This layer will be responsible for presenting the data analysis component of our projects to the warehouse managers.

1.2.2.1 Application

It will be designed as a web application. In this way, the application can be accessed from anywhere that network constraints permit. It will provide a simple interface to the user and hit the most important points by processing the raw data gathered from environmental hardware using backend layers. UI/UX decisions will be refined collaboratively with our AI mentor to ensure usability and clarity.

1.2.3 API Gateway Layer

The entire project will be designed as a collection of various microservices, each running on Docker containers. To redirect all requests and communications to the services, a gateway layer will be established.

1.2.3.1 Gateway

We planned to use a reverse proxy. All requests will be directed to the related backend services. It also provides us with load balancing while managing multiple microservices, enabling us to avoid overloads on specific services. Moreover, using a gateway provides us with the opportunity to standardize our request and response formats, making the implementation more scalable and efficient. As a final advantageous point, we can monitor and log all communication between services to ease the testing phase load.

1.2.4 Server Layer

The server layer will be the primary layer that operates the core functionalities of our project, including displaying RTSP streams on a web-based application, gathering and processing data using computer vision tools, and applying route planning algorithms to optimize related logistics services.

1.2.4.1 Streaming Services

The streaming service will consist of two main components: the Recording Service and the Streaming Gateway Service. The Recording Service will be responsible for tracking stream chunks in case of anomalies and for general recording functionalities. It will store the recording data as video chunks, not in a pathway style, as is generally done in standard databases. Since it will have a feature of recording replay, keeping record data as chunks in an object storage will enable us to store and process visual media data efficiently. This storage type enables clients to replay records without downloading the entire file and also provides media data frame by frame to computer vision services.
The other part is the Streaming Gateway Service. Streaming media data in web-based applications has two main issues: data format and delay-related problems. Modern web browsers support a limited number of protocols, like WebRTC [1]. However, security cameras give RTSP streams. Therefore, the conversion of these protocols into supported protocols is mandatory. Keeping the delay to a minimum while performing these conversions is another issue. The Streaming Gateway Service will utilize peer-to-peer protocols to minimize delay and prepare video frames in a manner that enables computer vision services to utilize them.

1.2.4.2 Computer Vision Services

The Computer Vision Service is a core module responsible for detecting item counts, occupancy information, and operational anomalies in real-time across warehouses and truck interiors. Using real-time camera streams and periodic image snapshots, the service applies a CV pipeline to estimate the number, size, and spatial arrangement of items on shelves, pallets, or truck floors. Anomaly detection components, powered by autoencoders and temporal consistency checks, flag irregular events such as damaged packages, abnormal loading patterns, or sudden changes in warehouse zones. All detections are aggregated into structured occupancy metadata, including item counts, free space estimation, and anomaly severity scores, then streamed down to the route planner and the dashboard.

1.2.4.3 Route Planning Services

The Route Planning Service is responsible for generating optimal delivery and pickup routes in a logistics ecosystem. It integrates real-time occupancy data from trucks and warehouses, obtained through a computer-vision (CV) service, and combines it with live operational parameters (such as traffic, load weight, fuel constraints, and time windows) to continuously compute efficient and feasible routing plans.

1.2.4.4 Core Backend Services

Auth and Analysis Service will be our two core backend services. The Auth Service will manage all authentication processes, including role and permission assignments, token generation, and session management. On the other hand, the Analysis Service will process the data gathered and processed by other layers and visualize it to the client. It will handle some pre-calculations using caching mechanisms to avoid loaded processing operations. Its main goal will be to present the data to clients in a clear and straightforward manner. Together, these two services will constitute the logical core of the system, providing security, structured access control, and scalable data analysis across the entire platform.

1.2.5 Data Layer

This layer will be responsible for storing all kinds of data that can be used in other layers, such as users' credentials, cached calculated statistical operations, and streaming records.

1.2.5.1 Database

Our database will be in a tabular form using SQL. It will be used mainly for static data, which refers to data that remains unchanged over extended periods. ORM tools, such as SQLAlchemy [2], will be used to ease SQL operations during the implementation phase.

1.2.5.2 Redis

Redis [3] is a decent cache storage for the project. Caching will be an important approach for us due to the large scale of data caused by video frames, and fast access to this frequently used data becomes essential. Some metadata, such as the status of environmental sensors and cameras, gathered from streams and control data, will be stored. Storing this metadata in memory instead of disk will reduce the response latency and improve the robustness of communications between microservices.

1.2.5.3 Object Storage

We will use an object storage solution for storing unstructured data. This includes video frames, which will be stored in a round-robin fashion, e.g., keeping the last 2 minutes. Aside from current frames, it will also store specific recordings on request or on anomaly detection.

1.3 Constraints

1.3.1 Implementation Constraints

Technology and Platform Constraints: These constraints shape both the system architecture and the development process. First, the platform depends on RTSP-compatible security cameras, which restricts data transfer to devices capable of providing real-time video streams. This requirement influences hardware selection, ensuring it is properly selected to meet streaming standards. Additionally, the Computer Vision module must operate using real-time inference frameworks, such as YOLO, that support video decoding and efficient model execution to meet the low latency. On the server side, the backend architecture must be implemented using Docker, which supports containerized deployment. Additionally, the application is limited to running on cross-platform web technologies. This prevents the use of platform-specific UI frameworks.

Real-Time Processing Constraints: The system must adhere to these constraints, which impact pipeline design and algorithmic complexity. The Computer Vision module, which includes video decoding, object detection, and event generation, must operate within a latency of 1,000 milliseconds. Computationally heavy models or multi-stage processing pipelines are restricted if they exceed the allowable delay [4]. Furthermore, streaming must be real-time; the system shall not allow the video to pause or delay frames because operators need to see events instantly to react quickly. Additionally, the Route Optimization module is necessary to compute updated routes within a strict time budget when warehouse and truck occupancies change. Because vehicle routing is an NP-hard problem, exact optimization techniques, such as full linear or integer programming solvers, are generally impractical for real-time [6]. Instead, the design must prioritize methods that are heuristic or approximation-based to deliver near-optimal solutions within the required response time.

Integration Constraints: All system modules must communicate through standardized REST/HTTP interfaces to ensure consistent communication. The system is also constrained to store data within the planned storage components. The primary database stores structured information, including details about trucks, warehouses, item records, and system configurations. Redis functions as a memory cache for real-time system states, temporary computation results, and event queues. The object storage service is used to store large unstructured data, such as short video clips and images generated. This separation prevents the use of ad hoc and allows for predictable data access patterns. Furthermore, the Computer Vision module must conform to the predefined event logging schema so that other services consume the information consistently. This guarantees that events, alerts, and associated metadata remain traceable and integrated throughout the system.

Resource Constraints: This constraint requires the entire pipeline to be optimized for efficient memory and compute usage. The system must control the clip duration, resolution, and encoding parameters such as compression format, bitrate, and FPS so that it prevents excessive storage [5]. Furthermore, the route optimization module must be designed to account for bounded memory usage. The size of the search space and the associated data structures, such as distance matrices, cost tables, and constraint sets, increase exponentially as the system grows. Therefore, the system must employ memory- and time-efficient algorithms that meet industry standards.

Security and Compliance Constraints: All communication, including video streams and API traffic, is required to use secure protocols such as HTTPS/TLS to ensure encryption. Data retention rules defined by the GDPR and KVKK limit the duration for which video clips, event logs, and operational records can be stored; therefore, storage design and automated deletion mechanisms must be implemented. Furthermore, role-based authentication and authorization for each user (operators, security personnel, administrators) can utilize the data and functions permitted by their role.

Testing and Deployment Constraints: Since live cameras are not always accessible during development, the pipeline must support testing with recorded video streams, enabling repeatable evaluations. The testing phase prioritizes modular validation, small-scale experiments, and controlled simulation setups rather than full industrial deployment scenarios.

1.3.2 Economic Constraints

Several economic constraints limit the selection of technologies, hardware, and system architecture. Since the system is built as an academic project, it must operate without the infrastructure typically available in industry, including GPU servers, specialized networking hardware, and professional camera systems. As a result, all major components must operate on standard workstations and low-cost development servers. This constraint directly influences model selection, as large deep learning models or computationally expensive optimization solvers cannot be used due to the high training and inference costs. Additionally, minimizing operational expenditures is necessary for long-term maintainability. TırGöz must avoid using paid cloud services, third-party APIs with subscription fees, or commercial optimization software. Instead, the platform should be built using open-source libraries, free tiers of cloud platforms if needed, and efficient storage solutions. Video storage must be controlled through clip duration limits, compression, and retention mechanisms to prevent excessive storage expenses. Hardware acquisition is also economically constrained. The system must rely on affordable RTSP-compatible security cameras, which are already in use at warehouses.

1.3.3 Ethical Constraints

TırGöz must comply with ethical responsibilities related to privacy, transparency, and surveillance. Since the system analyzes sensitive operational data coming from video streams, all collection and processing of visual information must be limited to tasks related to logistics. The system must avoid storing personally identifiable information not required for operational purposes. Additionally, the Computer Vision module must not be used for employee surveillance beyond the scope of the project. Ethical constraints also require transparency in how detections and alerts are generated [7]. The system must clearly convey the purpose and limitations of automated decisions to prevent operators from being misled by incorrect predictions. The decisions must be easy to understand so that operators can see why a detection or recommendation was made, so that the system assists the operators' judgement but not to replace it.

1.4 Professional and Ethical Issues

The development of TırGöz involves professional and ethical considerations. Because the platform integrates real-time video analytics, automated decision-making, and sensitive operational data from logistics environments, it is essential to ensure that engineering practices adhere to professional standards, including accuracy, transparency, and accountability. This includes validating route optimization outputs and documenting all architectural decisions. The team must avoid introducing unnecessary risks and assumptions, as incorrect alerts or route decisions can cause delays, misshipments, or safety issues. Software engineering standards, such as modular design, version control discipline, and secure coding practice, are necessary to follow professional standards. Additionally, the system must respect the privacy and autonomy of warehouse personnel. The system must not be used for any form of personnel surveillance. The system must collect only the data necessary for logistics operations and avoid storing any personal information unless required. Users should also be informed about what the detection system can and cannot do. Another important issue is the risk of bias in automated detection [8]. Warehouse conditions can vary in terms of lighting, item appearance, and worker activity, so the detection models must be tested on diverse types of data. Otherwise, it may count items incorrectly, trigger unnecessary alerts. The team must evaluate model outputs with transparency and adjust training data whenever unintended bias is detected. Data retention policies (GDPR and KVKK) must be adhered to, and stored footage is automatically deleted after the defined retention period. Any integration with third-party libraries or external APIs must respect licensing requirements to avoid violating intellectual property rights.

1.5 Standards

The development is guided by industry and software engineering standards. These standards influence the system architecture, coding practices, data handling, and development methodology.

Software Engineering Standards: The project adheres to widely accepted engineering principles, including modular design, separation of concerns, and a maintainable code structure, in accordance with ISO/IEC 25010 software quality guidelines. Version control workflows are managed using Git to trace, branching strategies, and controlled integration of new features. For healthy team communication, Jira will be used.

Deployment and Containerization Standards: The system utilizes Docker containerization, adhering to industry conventions for reproducibility, portability, and environment isolation. This ensures that services can be deployed reliably across development, testing, and production environments.

AI and Machine Learning Standards: The AI models used in TırGöz must operate reliably in various warehouse conditions, so they are trained and tested on diverse data before deployment. The system avoids “black box” behavior by providing clear explanations of detections and alerts when possible. Each model is checked for accuracy and consistency to prevent false alarms and incorrect item counts. Any updates to the model are carefully reviewed to ensure they do not reduce performance or create unexpected behavior.

2. Design Requirements

2.1 Functional Requirements

2.1.1 Streaming

- TırGöz must demonstrate the live stream of the truck loading/unloading with minimal possible delay.
- TırGöz must allow users to record desired parts of the live streams.
- TırGöz must record the short clips of detected events to provide visual evidence for alerts.

2.1.2 Detection

- TırGöz must detect the location of the items in the warehouse.
- TırGöz must detect the physically damaged items during the loading/unloading.
- TırGöz must send alerts for the detection of physically damaged items.
- TırGöz must detect the count of the item loaded/unloaded.
- TırGöz must detect if the item is loaded into the wrong truck.
- TırGöz must send alerts for the detection of wrongly loaded items.
- Based on the detections, TırGöz must demonstrate the
    - warehouse occupancy percentage for every warehouse.
    - truck occupancy percentage for every truck.

    - item counts in the trucks and warehouses.
    - locations of the items in the warehouse.

2.1.3 Route Optimization

- TırGöz must demonstrate the truck and warehouse counts in the system.
- TırGöz must calculate and demonstrate the optimal routes for the trucks based on the desired cost and time constraints.
- TırGöz must dynamically recalculate the optimal routes based on the changes in the truck loads.

2.2 Non-Functional Requirements

2.2.1 Usability

TırGöz focuses on high-efficiency usage for logistics operations. The system can be used with minimal effort due to its simple and user-friendly interface. The core functionalities are intuitive and can be learned easily. For instance, clear and immediate vehicle location, optimized path, and real-time alert displays do not require the user to search or interpret complex data. Additionally, to enhance usability, anomaly detection alerts are accompanied by short event clips for easy visual verification, thereby minimizing the time required for manual inspection.

2.2.2 Reliability

TırGöz ensures high operational reliability. Since the Computer Vision and Route Optimization Services are isolated, a failure in one independent module does not affect the other independent modules. Additionally, TırGöz prioritizes rapid recovery from any failure to maintain the integrity of real-time events and data, particularly during the critical hours of the day. Additionally, the anomaly detection function aims to achieve a high accuracy rate through deterministic validation checks.

2.2.3 Performance

The primary goal of TırGöz is to ensure low-latency processing due to its real-time functional nature. To maintain real-time alerts, TırGöz processes live video streams and detects anomalies with an end-to-end latency of at most **1000** milliseconds. Also, the Route Optimization Module aims to recalculate the optimal route based on warehouse and truck occupancies within a desired time constraint for the standard operational load.

2.2.4 Supportability

TırGöz is designed to be straightforward, making it easier to understand bugs and problems, thereby reducing the time required to fix errors and install new features. This is why logs are used to record detailed notes in a consistent and easy-to-read format. Additionally, TırGöz aims to minimize full system restarts, thereby keeping the system operational during minor system changes. To further increase supportability, up-to-date technical documentation is maintained for the system in a complete and accurate manner.

2.2.5 Scalability

TırGöz must be able to handle the growth in the number of cameras, trucks, and more deliveries. For instance, the Computer Vision Service and Event Processing API should not exceed the expected real-time constraint, even when the number of cameras and streams is increased. The Route Optimization Service must perform calculations for both smaller and larger numbers of trucks and deliveries. Additionally, TırGöz must efficiently manage the rapidly growing volume of stored data, including short video clips of detected anomalies, events, and metrics.

3. Feasibility Discussions

3.1 Market & Competitive Analysis

We searched for some real life systems that combine computer vision systems with warehouse monitoring and route optimisation. There are commercial tools in the market used for similar systems to our TırGöz project. These systems are mainly focused on by large scale companies that acknowledge the benefits of them and have the resources to implement or purchase such systems, such companies include DHL, UPS and Amazon. Such examples that partly implement our idea support the feasibility of our project, showing the underlying technologies needed are commercially available.

For the CV side of the project, we found companies that use cameras and AI models to monitor warehouse operations including stocking and loading. Those examples include Zebra Technologies, Hikvision, and Irida Labs.

Zebra Technologies' SmartPack Trailer mounts 3D sensors and cameras that view trailer doors to measure things such as load density, the amount of packages loaded into the trailer [9,10]. It gives feedback to the end user with dashboards and shows them the statistics [10]. This is parallel to TırGöz project's aim to estimate truck occupancy and loading quality from video streams; however requires specialised equipment for the same purpose.

Hikvision's Smart Dock Management System uses AI integrated cameras to detect which loading docks are occupied by which vehicles and if that dock is in active use for loading or unloading. End users can reach the state of their docks based on these cameras [11]. This is similar to our module that aims on using CCTV cameras to log these events.

Irida Labs provide CV based solutions to count pallets, monitor the current stock, detect occupancy rates of the loading docks [12]. This company lists many partners and customers such as Intel, Axis which supports their vision on Industry 4.0 solutions [13, 14]. Their product demonstrates our CV module of the pipeline is possible in real environments, alongside the demand for it.

In addition to these companies, large logistics operators discuss using CV to estimate available volume, measure, packing quality and detect defects during loading. An example of this is DHL which reports on these analytics capabilities on improving load-efficiency and reducing company resource waste [15].

These examples show, the hardware components assumed by our project are realistic. Existing cameras combined with modern object detection models are capable of handling real time monitoring of loading operations, warehouse and truck occupancy rates and damage.

For the route optimisation side of TırGöz concerned, there are systems present on a large-scale that uses operational data, alongside traffic integration. Noteworthy examples include UPS and Amazon whose operations depend on such systems because of their large scale.

UPS' On-Road Integrated Optimisation and Navigation or shortly ORION plans daily delivery routes for more than 55 000 UPS drivers' vehicles and uses dynamic optimisation that updates driver routes based on traffic condition and request for change in pickup [16].

Reports state ORION reduced road time and hence fuel consumption [17]. This supports our plan to re-route in response to changing loads of warehouse and trucks rather than the practice of static route optimisation.

Amazon’s logistic infrastructure shows that companies use AI supported heuristics, taking warehouse capacity constraints, and real time monitoring of warehouse occupancy [18]. However these systems rely on high-tech installations and come with the reliance on the large-scale cloud enabled systems of Amazon.

As a result many companies use CV for monitoring and route optimisation. However, we could not find a widely used product that is similar to ours that responds to needs of small to middle scale companies with low budgets for such infrastructure. Tırgöz project differentiates by its aim to enable such companies to integrate AI based knowledge by utilising existing hardware to generate warehouse data such as item counting, free space, anomaly detection in loading, truck occupancy rates in near real time. While incorporating this knowledge to dynamically optimise routing in a single system. Other advances with specific hardware can be taken into account for further advancements or user-specific projects based on customer feedback, however our initial plan is providing an accessible end-to-end system.

3.2 Academic Analysis

Academic research also suggests the feasibility of TırGöz since modules are validated in logistics settings. These components include real-time computer vision, anomaly detection, occupancy estimation and dynamic route optimisation.

Computer Vision based deep learning models can detect and count items in real warehouse environments based on the research done by Villegas et al. [19]. The article states CV improves inventory accuracy and reduces the manual workload for inventory counts, effectively reducing the manual workload [19]. Also D. Patel et al, demonstrate deep learning approaches succeed in effectively automating warehouse inventory systems [20]. These results support the usage of real time camera feeds to measure the truck load and use effectiveness.

Damage and other anomaly detections are also demonstrated to work by academic sources. One such article by Herrera-Toranzo et al, shows this by utilising the YOLOv5 model to differ between damaged and undamaged goods with high accuracy rates, showing modern models can reliably detect physical damages [21]. This is furthermore shown by Ayoola et al, this article combines computer vision with sensor data for similar monitoring needs [22]. These studies support TırGöz modules aiming at generating damaged-item alerts, clips for anomaly timeframes and other operationally relevant events.

The route optimisation component aligns with academic research on VRP or in other words Vehicle Routing Problem. A review article done in 2014 states the 30 years of exploding research on this area. This area of problem has been traditionally studied with static methods that disregards real-world constraints that change in real-time such as delays in truck departure that can be caused by warehouse specific reasons [23]. Solutions regarding these kinds of constraints have become viable and increased attention with the hardware advances enabling AI driven systems. Modern studies such as Abdirad et al, support dynamic routing under capacity and timing constraints are computationally feasible with heuristics and metaheuristics based approaches [24]. These studies show evidence for feasibility of our route optimisation under the assumption the data provided through other modules are accurate.

4. Glossary

Term	Definition
RTSP	A networking protocol used for streaming video from security cameras.
CCTV	Also called Closed Circuit Television. A type of surveillance camera.
YOLO	A deep learning model for object detection
Ad hoc	A temporary solution that does not meet the standards of the system

Anomaly Detection	Detecting unusual events and abnormal patterns.
Occupancy Estimation	Process of approximating how full a space is, in our case warehouse and truck load.
Heuristic Algorithms	Fast and rule-based solutions that determine good solutions. Used in our case because route optimisation is computationally expensive.
Meta-Heuristic Algorithms	Problem agnostic frameworks that guide and improve heuristic rules.
Vehicle Routing Problem	An NP-hard optimisation in the field of logistics. The goal is to find optimal sets of routes for a fleet.
Dynamic Routing	Real-time adjustments of delivery routes based on new information such as load delays, traffic.

5. References

[1] *World Wide Web Consortium (W3C)*, “*WebRTC 1.0: Real-Time Communication Between Browsers*,” *W3C Recommendation*, Jan. 2021. [Online]. Available: https://www.w3.org/TR/webrtc/. Accessed: Nov. 21, 2025.

[2] *SQLAlchemy Project*, “*SQLAlchemy Documentation*.” [Online]. Available: https://www.sqlalchemy.org/. Accessed: Nov. 21, 2025.

[3] “Develop with Redis,” Redis Documentation, 2025. [Online]. Available: https://redis.io/docs/latest/develop/. Accessed: Nov. 21, 2025.

[4] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. *YOLOv4: Optimal Speed and Accuracy of Object Detection.*arXiv:2004.10934.

[5] Xu, M., et al. *A Survey on Video Compression and Streaming in Machine Vision Systems.* ACM Computing Surveys, 2022.

[6] Toth, P., & Vigo, D. *The Vehicle Routing Problem.* SIAM Monographs on Discrete Mathematics and Applications, 2001.

[7] Ribeiro, M. T., Singh, S., & Guestrin, C. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.* KDD 2016.

[8] Selbst, A. D., et al. *Fairness and Abstraction in Sociotechnical Systems.* FAT* 2019.

[9] Zebra Technologies, “TM2000 Trailer Monitoring Unit,” 2025. [Online]. Available: https://www.zebra.com/us/en/products/spec-sheets/smartpack/tm2000.html

[10] Zebra Technologies, *SPC-SPT Dashboard User Guide*, Zebra Technologies, Lincolnshire, IL, USA, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://www.zebra.cn/content/dam/support-dam/en/documentation/unrestricted/guide/software/SPC-SPT-Dashboard-User-Guide-en.pdf

[11] Hikvision, “Dock Management,” Hikvision Solutions by Function, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://www.hikvision.com/en/solutions/solutions-by-function/dock-management/

[12] Irida Labs, “Loading Dock Monitoring powered by Vision AI,” *PerCV.ai Industry 4.0 Suite*, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://iridalabs.com/automated-loading-dock-monitoring-using-vision-ai/

[13] Irida Labs, “Intel Partnership,” *Irida Labs Website*, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://iridalabs.com/intel-partnership/

[14] Axis Communications, “Irida Labs Vision AI solutions,” Axis partner page, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://www.axis.com/partner/irida-labs

[15] “AI-Driven Computer Vision: A DHL perspective on the impact of Computer Vision in logistics,” DHL. [Online]. Available: https://www.dhl.com/content/dam/dhl/global/csi/documents/pdf/DHL%20Trend%20Report%20AI-Driven%20Computer%20Vision%20-%20Full%20Report.pdf

[16] C. Holland, J. Levis, R. Nuggehalli, B. Santilli, and J. Winters, “UPS Optimizes Delivery Routes,” INFORMS Journal on Applied Analytics, vol. 47, no. 1, pp. 8–23, 2017, doi: 10.1287/inte.2016.0875.

[17] Team Ascend, “Boost logistics, slash costs: How UPS’s ORION system slashed delivery costs with route optimization,” Ascend Analytics, Jul. 2, 2025. [Online]. Available: https://www.ascendanalytics.co/post/how-upss-orion-system-slashed-delivery-costs-with-route-optimization

[18] Analitifi, “How Amazon Uses AI to Revolutionize Supply Chain & Warehouse Automation,” Analitifi, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://analitifi.com/amazon-logistic-ai-supply-chain-warehouse-automation/

[19] W. Villegas-Ch, A. Maldonado Navarro, and S. Sanchez-Viteri, “Optimization of inventory management through computer vision and machine learning technologies,” *Intelligent Systems with Applications*, vol. 24, p. 200438, 2024, doi: 10.1016/j.iswa.2024.200438.

[20] A. D. Patel and A. R. Chowdhury, “Vision-based object classification using deep learning for inventory tracking in automated warehouse environment,” in *Proc. 2020 20th Int. Conf. Control, Automation and Systems (ICCAS)*, Busan, Korea, Oct. 2020, pp. 1–6, doi: 10.23919/ICCAS50221.2020.9268394.

[21] P. Herrera-Toranzo, J. Castro-Rivera, and W. Ugarte, “Detection and verification of the status of products using YOLOv5,” in *Proc. 20th Int. Conf. Smart Business Technologies (ICSBT 2023)*, 2023, pp. 83–93, doi: 10.5220/0012123500003552.

[22] V. B. Ayoola, G. Osam-Nunoo, C. Umeaku, and B. O. Awotiwon, “IoT-driven smart warehouses with computer vision for enhancing inventory accuracy and reducing discrepancies in automated systems,” *IRE Journals*, vol. 8, no. 5, pp. 176–183, Nov. 2024.

[23] H. N. Psaraftis, M. Wen, and C. A. Kontovas, “Dynamic vehicle routing problems: Three decades and counting,” *Networks*, vol. 67, no. 1, pp. 3–31, 2016. doi: 10.1002/net.21628.

[24] M. Abdirad, K. Krishnan, and D. Gupta, “A two-stage metaheuristic algorithm for the dynamic vehicle routing problem in Industry 4.0 approach,” *Journal of Management Analytics*, 2020. doi: 10.1080/23270012.2020.1811166.