

Article

From Layout to Data: AI-Driven Route Matrix Generation for Logistics Optimization

Ádám Francuz and Tamás Bányai * 

Institute of Logistics, University of Miskolc, 3515 Miskolc, Hungary; adam.francuz@student.uni-miskolc.hu

* Correspondence: tamas.banyai@uni-miskolc.hu

Abstract

This study proposes an end-to-end mathematical framework to automatically transform warehouse layout images into optimization-ready route matrices. The objective is to convert visual spatial information into a discrete, graph-based representation suitable for combinatorial route optimization. The problem is formulated as a mapping from continuous image space to a structured grid representation, integrating image segmentation, graph construction, and Traveling Salesman Problem (TSP)-based routing. Synthetic warehouse layouts were generated to create labeled training data, and a U-Net convolutional neural network was trained to perform multi-class segmentation of warehouse elements. The predicted grid representation was then converted into a graph structure, where feasible cells define vertices and adjacency defines edges. Shortest path distances were computed using Breadth-First Search, and the resulting distance matrix was used to solve a TSP instance. The segmentation model achieved approximately 98% training accuracy and 95–97% validation accuracy. The generated route matrices enabled successful construction of feasible and optimal round-trip routes in all tested scenarios. The proposed framework demonstrates that warehouse layouts can be automatically transformed into discrete mathematical representations suitable for logistics optimization, reducing manual preprocessing and enabling scalable integration into digital logistics systems.

Keywords: route planning; layout optimization; digitization; image processing; deep learning; U-net; convolutional neural networks

MSC: 68T02



Academic Editor: Shih-Wei Lin

Received: 27 January 2026

Revised: 16 February 2026

Accepted: 5 March 2026

Published: 7 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The digital evolution and transformation of recent years can be observed in most industries, including logistics and supply chain management (SCM). The possibilities offered by artificial intelligence (AI) can transform many new logistics processes in order to reduce the time required for processes, thereby releasing significant resources and saving considerable costs. These advantages can be utilized by the literature as new and innovative research opportunities and by industrial practice as opportunities for financial savings.

The application of artificial intelligence in the logistics industry is extremely diversified, as due to the complexity of the processes, the efficiency of each sub-process can be further developed and increased by utilizing the possibilities of tools and digitization. This makes the topic of logistics optimization multifaceted, as there are numerous models or algorithms available to solve a given problem. The use of AI capabilities in logistics applications can significantly improve decision-making processes, optimize resource utilization, and

minimize environmental impacts [1]. This research examines the integration of artificial intelligence into the Traveling Salesman Problem (TSP), so it is important to consider the special properties of TSP in theoretical considerations: the starting and ending points must be the same, and each defined point must be visited once. According to these properties, the Traveling Salesman Problem is an NP-hard combinatorial optimization problem [2], as the number of possible routes increases factorially ($n!$), so the exact, globally optimal solution can only be calculated within a realistic time frame in small-scale cases. To solve this problem, several metaheuristic algorithms have been developed that use a global search strategy with local improvement, thereby providing acceptable, near-optimal solutions. The most widely used algorithms are Simulated Annealing (SA), Genetic Algorithms (GAs), Ant Colony Optimization (ACO), and Tabu Search (TS) [3].

Effective implementation of warehouse layout planning is critical, as companies generally focus on maintaining productivity and competitiveness. Warehouse activities are well known for their impact on the entire supply chain, which also confirms the relevance of layout planning [4]. Although warehousing is a key component of SCM, warehousing research remains an understudied area in supply chain research and accounts for only a small fraction of all research conducted in this field [5]. When designing warehouse layouts, numerous parameters must be taken into account, which means that the design process can be interpreted as a complex decision-making problem that cannot be solved using a single objective function and can only be resolved through compromise [6]. However, most existing approaches assume that the required structured input data (such as distance matrices, adjacency graphs, or discrete route representations) are already available. The transformation of visual warehouse layouts (e.g., floor plans or CAD-based images) into optimization-ready graph or matrix representations typically requires manual interpretation, preprocessing, or rule-based modeling. Consequently, although route optimization itself is well studied, the automatic conversion of layout images into structured route matrices remains largely unaddressed in the literature. This gap limits scalability in dynamic environments, where layout modifications would require repeated manual data reconstruction.

In the case of an existing warehouse with a layout, route planning is already achievable, because the necessary data is available: route matrices, distance values, starting points and destinations, obstacles, etc. Numerous strategies and objective functions can be used for route planning and control [7].

This scientific research examines the conversion of warehouse layout images into optimized routes using artificial intelligence. The article describes the functioning of the framework, its advantages, limitations, and possibilities for further development. Through a systematic review of the literature, we analyze the results and connections of published research, then explain the specific goal and presumed advantages of the research. The article presents the steps involved in generating the layout images used in the research, their processing and conversion into a path matrix, and finally describes the optimization algorithm. The hypothesis of the study is to use modern technologies to create a framework that reduces the time required for optimization and the need for human interaction, resulting in a product that can be integrated into a real industrial system and has practical benefits in addition to its value in the literature. This paper also discusses limitations and assumptions, which may inspire further research and new ideas.

2. Materials and Methods

The purpose of this section is to review the literature related to the research topic and to present the research objectives and hypotheses. In the first subsection, we present the published literature, which we examine according to keywords and perform several

vehicle routing and routing algorithms. Location problems are a particularly important topic in current scientific research, as AI automates these problems and reduces the amount of human interaction. The published research highlights that the quality of the distance matrices used as input is crucial, as inaccuracies can distort the overall optimization [8]. The effectiveness of algorithms depends heavily on the quality of the data, as well as its structure and preprocessing [9]. The green cluster represents logistics optimization and cost focus, as the main keywords are cost reduction, sensitivity analysis, and decision-making. These studies clearly show that the goal of all logistics optimization is cost reduction, as companies increase the efficiency of their processes in order to reduce their costs in some way. The theoretical assumption can be justified, and logistics optimization directly contributes to cost reduction and service quality improvement, especially when decision-making is supported by integrated analytical models [10]. These models are based on structured and readable input data for the purpose of various simulations and optimization [11]. The previous statement well supports the research hypothesis that an AI-supported optimization framework can be created using existing input data. The yellow cluster includes the keywords supply chain and information management. In these scientific approaches, the warehouse is an element of the entire system, not an independent object. Most articles emphasize that data and information control are key to warehouse efficiency, making the warehouse not only a physical space but also a data source [12]. This research is based on the mentioned data source, as the route matrix can be generated by processing warehouse layout images. The last and most dynamically growing group is the red cluster, which includes the keywords AI, robotics, and modern warehousing. This cluster contains the main technological keywords (e.g., deep learning—DL, reinforcement learning—RL), which is a good indication of the technological capabilities and optional development opportunities associated with the topic [13]. Keywords related to this scientific study can also be found in this cluster, as the terms image processing, layout recognition, and automation all fit this approach. The published articles show that artificial intelligence techniques significantly improve warehouse operations, especially order picking and route planning, but generally assume predetermined warehouse layouts [14].

To examine the use of artificial intelligence, we performed a new search, filtering for the keywords “warehouse,” “logistics,” and “AI,” and found a total of 174 scientific articles. The small number indicates the specialized field of research and the potential for new studies and innovative solutions. We performed another cluster analysis (see Figure 2) on the keywords that appeared and created a network diagram containing a total of 73 keywords divided into 5 clusters.

The keyword co-occurrence map clearly demonstrates the dominance of technological and analytical concepts in warehouse and logistics research. Core terms such as artificial intelligence, optimization, warehouse, and supply chain occupy central positions, indicating strong interconnections between analytical methodologies and operational logistics problems.

The red cluster is centered around artificial intelligence and optimization-related topics, including forecasting, electronic commerce, and order picking. In these studies, artificial intelligence primarily functions as a decision-support and analytical tool rather than a spatial modeling component. Predictive analytics, for example, significantly improves supply chain decision quality and warehousing efficiency [15]. However, efficient decision support requires properly structured and reliable data sources, without which analytical tools cannot be effectively implemented [16].

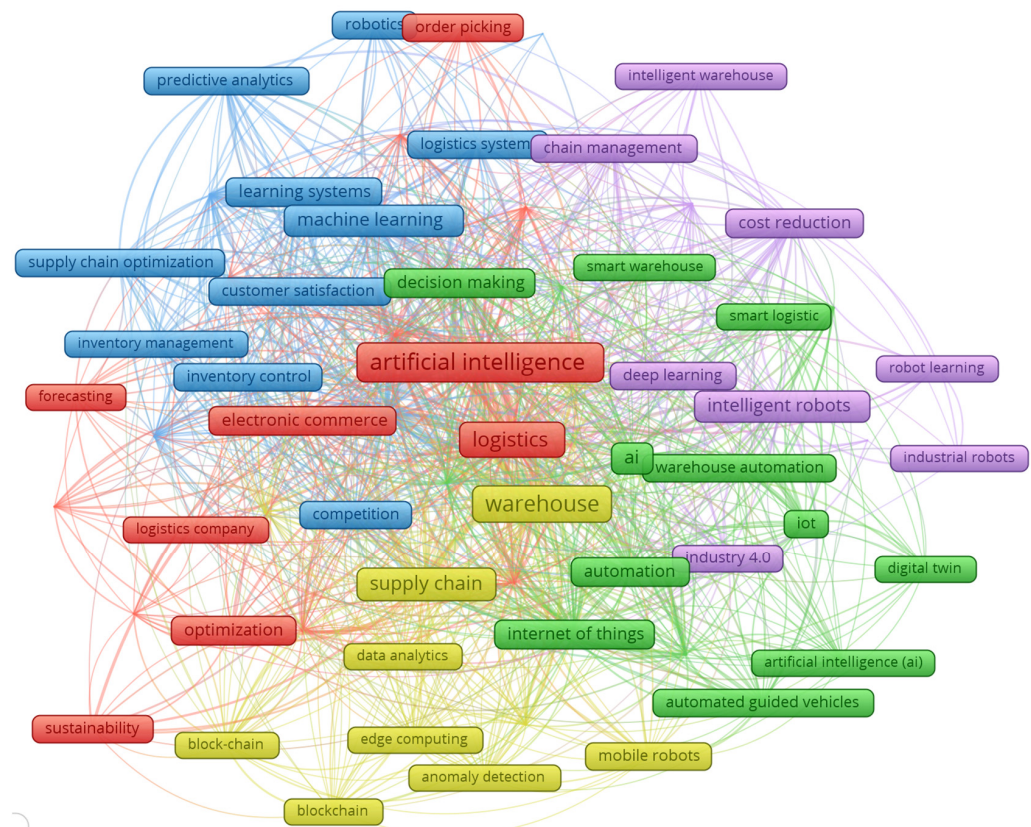


Figure 2. Most popular keywords in warehouse and logistics optimization with AI research (source: VOSviewer).

The green and purple clusters reflect Industry 4.0-related concepts such as digital twin, IoT, automation, robotics, and intelligent warehouse systems. The essential feature of the fourth industrial revolution is the continuous generation and utilization of real-time data from physical processes, enabling automated and adaptive decision-making [17]. The digital twin paradigm represents physical systems in digital space and continuously updates them with real-world data, allowing simulation, forecasting, and optimization [18]. Mixed reality (MR) technologies can further enhance this architecture by enabling real-time interaction between physical and digital environments [19]. Similarly, the Internet of Things (IoT) connects physical devices such as sensors, RFID systems, machines, and vehicles, facilitating data exchange and decentralized decision-making [20]. Although these technologies are widely studied, their integration with segmentation-based layout interpretation and graph-based routing optimization remains limited.

The blue cluster groups machine learning, predictive analytics, routing, and robotics-related concepts. Some studies apply visual inputs and deep learning techniques for robotic perception and control [21], while convolutional neural networks (CNNs) are capable of extracting high-level spatial features from raw images [22]. However, these applications typically focus on perception or control tasks rather than integrating visual layout understanding with combinatorial routing optimization.

The yellow cluster represents classical supply chain management (SCM) topics, including inventory management, customer satisfaction, and forecasting. AI and machine learning tools have significantly enhanced predictive performance in these domains [23], yet their application is often restricted to demand forecasting or inventory control. Importantly, the effectiveness of such analytical methods strongly depends on data availability and quality [24].

At the same time, several concepts central to the present research—such as image-based layout extraction, automated distance matrix generation, and layout-to-graph

transformation—do not appear explicitly among the dominant keywords. This suggests that while AI and digital technologies are extensively investigated in logistics, the integration of visual layout digitization with routing optimization represents a relatively underexplored intersection in the literature.

Mahroof states that existing warehouse research has primarily focused on warehouse design, performance, and technology use, ignoring the determining factors for AI application in warehouses [5]. Some studies classify AI as an Industry 4.0 technology and put it on the same level as IoT and cloud-based computing. It is a fact that, despite the undeniable importance of the topic, the primary research looking at the impact of IoT technology is inconsistent and scattered [25]. The key to the effective operation of the IoT is the integrated application of reliable and energy-efficient wireless networks and the key technologies that support them [26].

The use of AI is surrounded by many concerns, and most employees fear losing their jobs. Soumpenioti et al. emphasize in their research that AI technology only transforms roles in logistics through automation and the appearance of new positions. Routine tasks such as data entry and inventory management will be automated, resulting in streamlined processes and increased operational efficiency. At the same time, the introduction of AI will require the creation of new jobs, such as AI system trainers, data analysts, and AI strategists [27]. In human–robot collaboration (HRC), safety is of paramount importance in the fields of intelligent manufacturing and automated logistics [28]. The most popular articles apply reinforcement learning (RL) [28], time-series analytics [29], cyber-physical systems [30], machine learning [31] and Tabu Search [32] solutions during integration.

A detailed review of the literature shows that our research is fully in keeping with the Industry 4.0 trend and that existing research covers warehousing and SCM tasks well. However, there are still many new research opportunities available, so our research may also be able to offer a new and innovative solution.

2.2. Research Objectives

The aim of this research is to create a complex, end-to-end framework that can generate a route matrix suitable for optimization from visual input, and then use this to determine a delivery route or vehicle route. The study results in an automated, discrete spatial representation model that provides an optimizable input (see Figure 3). The main advantage of the methodology is that it works in dynamic environments, making it highly scalable.

In the traditional process, the manual layout is first interpreted, during which an expert examines the floor plan and decides where the shelves, paths, and obstacles are located. After interpretation, manual or semi-automatic digitization takes place, which can be supported by various software, but manual intervention is always necessary in interpreting the cases in question. After digitization, it is necessary to manually construct the route matrix, define passable and impassable areas, and calculate distances. Optimization can be performed on the resulting road matrix, but the departure and arrival points and various restrictions must be decided manually. Existing errors are also handled manually. These processes are time-consuming, depend on human factors, are difficult to scale, and require the entire process to be repeated if the layout changes.

The first and only manual step in the methodology developed as a result of the research is to send the layout to the model, which automatically processes the received image and recognizes the passable area, shelves, and obstacles. This is followed by automatic path matrix generation, where each object takes on a unique value at the type level, thus creating a representation ready for optimization. The model checks for optimizability, i.e., the accessibility of targets and the connected path. The optimization algorithm created is capable of recognizing the picking point and determining the shortest route. As a result of

full automation, redesign is also possible without human intervention. The methodology can be fully integrated into the digital twin approach.

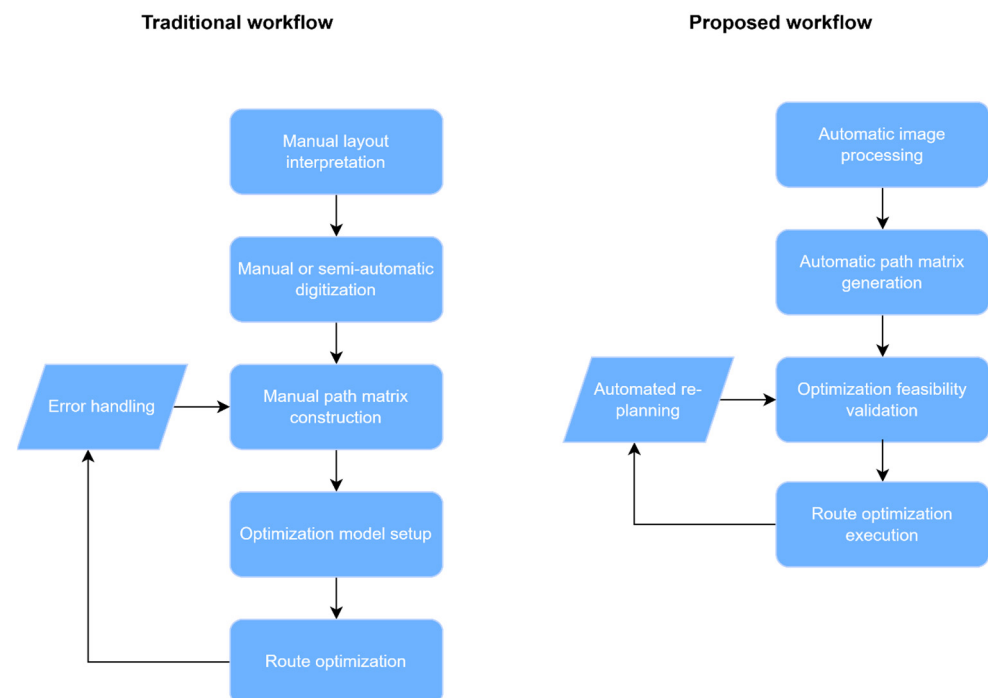


Figure 3. Workflows of optimized route planning from layout (source: own elaboration).

The contribution of the research is that the presented methodology completely replaces manual layout interpretation, manual route matrix creation, and pre-processing steps, and can reduce modeling time and error correction cycles. In addition to reducing process time, the model can be used for mass simulation and testing of different scenarios (e.g., number of shelves, locations).

The number of publications closely related to the research is limited, and no previous research has been conducted on a similar topic covering the entire framework. However, several articles have been published on elements of the research (image processing, route matrix creation, optimization), so it is advisable to analyze them.

Image processing has already appeared in warehouse environments, but its purpose was to detect barcodes, for which (in accordance with our research) CNN was used, and the information was processed and applied by UAVs. As a result, the process reduces the time required for warehouse inventory and the number of errors that occur during barcode scanning [33]. Image detection has also appeared in pharmaceutical warehouse management, where the Faster R-CNN method was used to identify pharmaceutical products. With the expansion of AI, the results suggest that intelligent warehouse management can lead to cost savings and increased efficiency in pharmacies [34]. Beyond logistics, there is research dealing with the processing of floor plans. These studies are mainly based on floor plans of general real estate (houses, apartments). The need for detection already existed before the appearance of AI, with the main reason being the development of Location Based Services (LBSs) [35]. According to Jang et al., the main problem is that there is not much indoor data available, but demand is growing fast due to digitalization and data-driven operations [36]. Based on this statement, the main obstacle to automated operation is the lack of data. The summary of main research fields is shown in Table 1.

Table 1. Summary of main research fields.

Approach	Main Focus	Limitations	Novelty
CNN-based barcode detection with UAV trajectory optimization [33]	Autonomous warehouse stocktaking using barcode-driven UAV localization and path planning	UAV energy limits; barcode dependency	Barcodes used as localization landmarks for optimized UAV trajectories
Faster R-CNN visual object detection [34]	Accurate identification and counting of pharmaceutical products in warehouses	High computational and data requirements	Application of Faster R-CNN with extensive comparison to classical and modern detectors
CNN-based floorplan segmentation and vectorization [36]	Automatic extraction of walls and doors from floorplan images and conversion to IndoorGML/CityGML	Sensitive to floorplan quality; static environments only	Standard-compliant indoor models with preserved wall thickness from images
Database-driven CAD model processing [35]	Extraction of indoor spatial objects from CAD for indoor LBS queries	Requires structured CAD data; no learning capability	High-precision indoor map generation via database modeling
This research	Transforming warehouse layout images into graph/matrix representations for route optimization	Synthetic layouts; real-world generalization under study	End-to-end pipeline from images to optimization-ready representations, enabling automated logistics benchmarking

Despite the input warehouse layout image, the research is not a visual application, but rather a mathematical representation problem. We examine how a continuous or discrete spatial structure can be represented as a formal, optimizable mathematical object (graph, path matrix). This problem is classical applied mathematics, involving discrete mathematics, graph theory, and optimization, where image processing is only a tool. As the methodology developed is not warehouse-specific but highly scalable and applicable to a variety of logistics spaces, the research is not an industrial case study but a generalized mathematical mapping framework that fits well with the subject matter of *Mathematics* and is consistent with a multidisciplinary approach.

3. Results

In this section, we present the structure of the mentioned framework and the research methodology. The structure of the chapter follows the research process and provides a complete picture of the operation, results, and limitations of the processes implemented. In the first subsection, we introduce the process of creating randomly generated route matrices used in the research, the standards followed, and the images generated. The second subsection describes the image processing, in which we used CNN and U-Net architectures. The quality of image processing is demonstrated using classification metrics (accuracy, confusion matrix, precision, recall). Based on the created route matrix, we perform optimization in the third subsection and discuss the specific parameters of the present research.

3.1. Layout Generation

The most important prerequisite for the study is the layout images, the relevance of which has been emphasized in several publications [8–11,16,24]. As a first step in our research, we looked for layout images in numerous sources and on various platforms, but

after a long search, we concluded that there is no publicly available database with sufficient data to train a neural network, so another solution is needed to generate the training and test datasets.

As warehouse layouts are composed of well-defined elements and their structure can be easily generalized, we created randomly generated layout images based on a specific logic, applying several restrictive conditions. An important precondition is that we only created floor plans of the same size (24–30) surrounded by walls. In industrial practice, there may be more complicated floor plans, but the goal of the study was to detect layout images, which can be researched with sufficient quality even with regular-sized floor plans. The floor plans created have a raw material warehouse layout and correspond to the general material flow process in warehousing (receiving, storage, order picking, production). To generate the layout images, we analyzed several real industrial floor plans to define the elements [37–41]. After examining these images, the following objects were defined, which will also appear in the randomly generated images:

- wall: the object surrounding the warehouses, its purpose is to enclose and close off the area;
- shelf: the main element of the warehouse, the storage location for materials; this object is the main element of optimization, as the route must reach the shelves;
- pillar: a fundamental obstacle in the warehouse that must be avoided during the route;
- receiving area: the place where received products are stored before storage; it does not have an important role in the optimization process;
- picking area: the picking point, which serves as both the starting and ending location of the optimized route, thereby formulating the TSP;
- entrance/exit: an important element of the layout, but not relevant in optimization
- docking station: an important element of the layout, but not relevant to optimization.

We used these components to create randomized layout images. To illustrate the structure, we generated a 3D image of some layouts using Plant Simulation software (Tecnomatix Plant Simulation 2302), which can be seen in Figure 4. The layout created looks like a realistic and actual raw material warehouse. The image shows the shelves, the receiving area, and the picking point.

We created a total of 1000 layout images to train the neural network. We also generated a .csv file to accompany the images, as the detection problem can also be interpreted as a classification task, as the CNN recognizes objects and compares them with the .csv file using various metrics, giving us the model's performance. The images are created in 2D using Unity software (Version 2022.3.59f1).

The layout images are created using a randomization algorithm that follows precise and strict rules to ensure realistic results. First, the dock gates and receiving area are defined. The width of the dock gates is determined randomly, and their position is aligned with the northern wall of the warehouse. This is paired with the receiving area, whose dimensions are also random within a certain range. The entrance/exit door is located at the eastern or western end of the warehouse. These objects do not participate in the later route optimization, but they play an important role in generating a realistic layout and can also be examined during detection.

Shelf orientation plays a key role in layout generation. There is a permanent main aisle in the middle of the warehouse, which divides the warehouse into two parts. The orientation of the shelves can be horizontal or vertical, and it is determined randomly. This binary decision has a major impact on the graph structure and subsequent routes. In addition to the main aisle, cross aisles also appear at random intervals to achieve a more realistic and not entirely regular traffic network. The picking area is created inside the warehouse with random dimensions and positions, overwriting the shelves located there

when it is created. Finally, pillars are placed for realistic layouts, which are arranged in a deterministic, grid-like pattern. During optimization, these columns appear as obstacles that the algorithm must handle. In addition to the outer walls, the width of the main aisle, the grid-like placement of the columns, and the basic cell hierarchy are not affected by random values, so it can be concluded that every generated layout remains consistent, navigable, and optimizable. The 2D layout generated by Unity can be found in Figure 5.

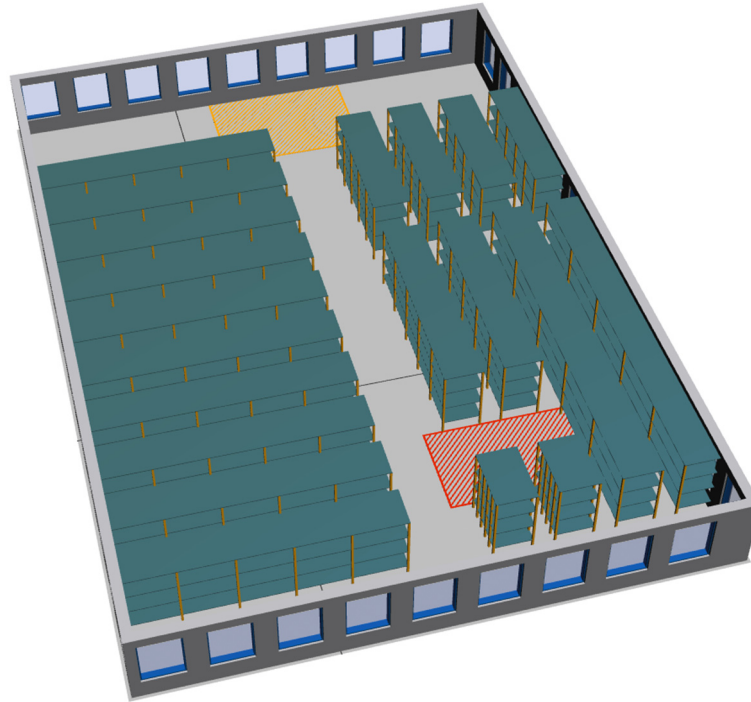


Figure 4. Generated layout example (source: Plant Simulation).

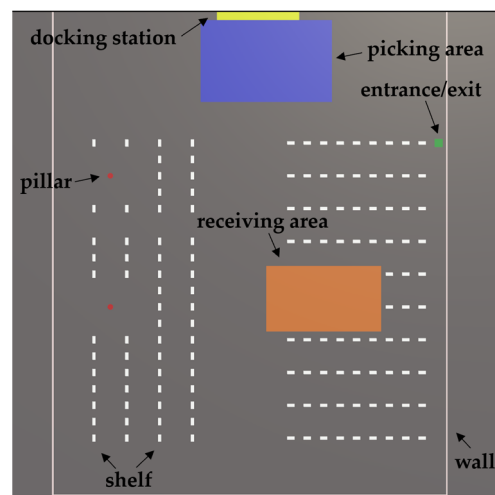


Figure 5. Generated randomized 2D layout (source: Unity).

3.2. Image Segmentation

After creating 1000 warehouse layouts, we will analyze the processing possibilities of the layout images as the main focus of our research. In this subsection, we present the steps and elements of neural network configuration in a step-by-step approach. First, we describe the characteristics of the problem, the requirements, and the input and output expectations. Next, we examine and select the architecture that best suits the problem, providing a comprehensive and general overview. Using the model, we create the entire

framework on which we perform training and optimization. We evaluate the modeling using classification evaluation metrics.

3.2.1. Problem Formulation

The model receives synthetic warehouse images as input (X), as presented in the previous section, and the output (Y) is a discrete, cell-based spatial representation. During input, H and W are the height and width of the image, and each pixel has 3 different color values (red, green, blue—RGB). Finally, a *ToTensor()* transformation normalizes the values, so the input formula is

$$X \in [0, 1]^{3 \cdot H \cdot W}, \quad (1)$$

where $H = W = 256$.

The model converts the input image into structured spatial data, resulting in a discrete grid whose values represent objects and whose size is determined by the size of the grid's rows and columns:

$$Y \in [0, \dots, C - 1]^{R \cdot S}, \quad (2)$$

where $R = 30$, $S = 24$, $C = 8$.

Due to the transformation, the input and output dimensions are not equal.

$$(H, W) \neq (R, S). \quad (3)$$

During the image generation discussed in Section 3.2, we also created CSV files linked to the layout, whose structure is identical to the output presented here.

The neural network represents the input values in a logit space (see Figure 6).

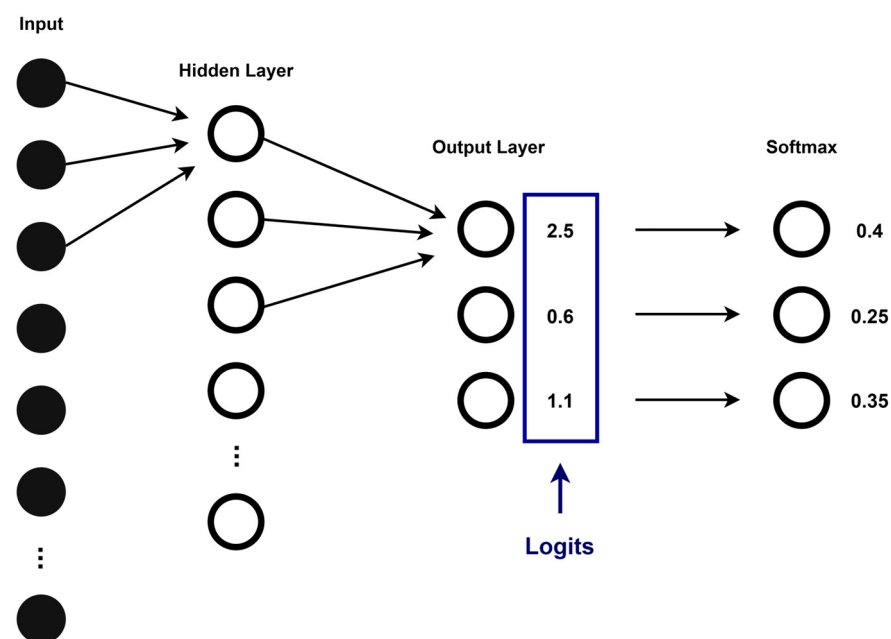


Figure 6. Neural network with logit values (source: own elaboration).

Here, each class and pixel has a logit value, which is an intermediate value in the neural network before the probability value. The purpose of this is to express the given class as a raw score, indicating how strong it is compared to the others in the case of the given pixel. If the network learned probability directly, it would be numerically and mathematically unstable, as it would have to be limited to $[0, 1]$, the sum would have to be 1, and gradient problems would arise. Instead, the network represents logit values, and a softmax function converts them into probabilities [42].

Network parameterized mapping with learned weights:

$$f_{\theta} : [0, 1]^{3 \cdot H \cdot W} \rightarrow \mathbb{R}^{C \cdot H \cdot W}, \quad (4)$$

where $f_{\theta}(X) \in \mathbb{R}^{8 \times 256 \times 256}$.

After determining the logit values, we represent the pixel-wise probabilities of the image on the grid (D operator). We use logit values for representation because the logit space is the only mathematically correct space where this representation can be performed without distortion and instability. D is continuous and differentiable.

$$\begin{aligned} D : \mathbb{R}^{C \cdot H \cdot W} &\rightarrow \mathbb{R}^{C \cdot R \cdot S} \\ Z_{\theta}(X) = D(f_{\theta}(X)) &\in \mathbb{R}^{8 \cdot 30 \cdot 24} \end{aligned} \quad (5)$$

Finally, the prediction is performed at the cell level, and the (r, s) grid cell is assigned to the class with the highest logit value:

$$\widehat{Y}_{r,s} = \underset{c}{\operatorname{argmax}} Z_{\theta,c,r,s}(X), \quad (6)$$

where $r \in \{1, \dots, R\}, s \in \{1, \dots, S\}, c \in \{0, 1, \dots, C-1\}$ and

$$c = \begin{cases} 0 & \text{if the cell represents an empty area} \\ 1 & \text{if the cell represents a wall} \\ 2 & \text{if the cell represents a pillar} \\ 3 & \text{if the cell represents an entrance or exit} \\ 4 & \text{if the cell represents a dock door} \\ 5 & \text{if the cell represents a picking area} \\ 6 & \text{if the cell represents a receiving area} \\ 7 & \text{if the cell represents a shelf} \end{cases}. \quad (7)$$

3.2.2. Model Selection

During the model selection process, the most important consideration is to choose a model that is appropriate for the specific characteristics of the task. The goal is to predict a structured, discrete grid, where the topology of the grid is also an important consideration; not only pixels, but also minimal error and offset can be critical from an optimization perspective.

As an initial baseline, a simplified encoder-based segmentation network was used to solve the problem. The *SimpleSegNet* model is an encoder-type CNN consisting of a series of Conv-ReLU-Pooling blocks, but it requires tedious and long-lasting fine-tuning in order to draw an optimal configuration for the task at hand [43]:

$$f_{\theta}(X) = E_L \circ E_{L-1} \circ \dots \circ E_1(X), \quad (8)$$

where $l \in \{1, \dots, L\}$.

where every E_l contains convolutional and pooling layers. However, this model has several limitations, which are critical given the structure of the current task. The purpose of pooling layers is to subsample the input image in order to reduce the computational load, memory usage, and number of parameters (thereby limiting the risk of overfitting) [44]. Repeated pooling causes localization to be lost, and this uncertainty is intolerable in a segmentation model. If we use L pooling layers, then the height and width of the image is

$$(H, W) \rightarrow \left(\frac{H}{2^L}, \frac{W}{2^L} \right). \quad (9)$$

Another limitation is the absence of skip connections, as the decoder only uses the output of the immediately preceding layer and not the earlier representations, resulting in distorted information:

$$\tilde{x}_{l-1} = U_l(\tilde{x}_l). \quad (10)$$

where x is an output representation of a layer (feature map), U_l is the decoder operator at the level l . Due to the limitations mentioned above, we use the U-Net segmentation model in our research, where the decoder does not rely solely on pooled representation, but also uses localized information prior to pooling and incorporates it into the decoder:

$$\tilde{x}_{l-1} = U_l(\tilde{x}_l, x_{l-1}), \quad (11)$$

where $\tilde{x}_l$ contains deep, abstract information, while x_{l-1} contains localized information prior to pooling.

3.2.3. U-Net Architecture Overview

U-Net is a Fully Convolutional Network (FCN) architecture designed specifically for pixel-level image segmentation tasks, especially in fields where only limited training data is available. The main motivation is that traditional CNN models are suitable for global classification, but do not provide localization, i.e., they do not indicate where the object is located in the image. The key idea behind U-Net is that it is built on a symmetric encoder–decoder model (see Figure 7), which first learns features and then restores spatial resolution using upsampling, while directly transferring localization information from the encoder to the decoder using skip connection steps [45]. Due to its topological and global recognition capabilities and its ability to transform to other dimensions, U-Net is a sufficiently good choice for segmenting and transforming warehouse layout images [46].

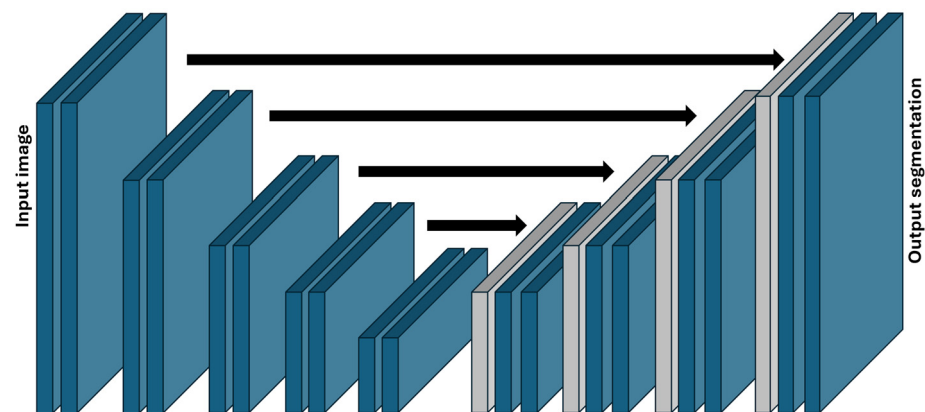


Figure 7. U-Net architecture (source: own elaboration).

The first element of the architecture is the encoder; its task is to extract spatial patterns from the input image while gradually increasing the level of abstraction of the representation. The encoder contains convolutional, nonlinear activation (e.g., ReLU), and pooling layers. During this process, the image resolution decreases, and the number of channels increases, allowing the network to learn the location of objects and topological relationships. In the decoding, the model converts the learned abstract representation back into a spatially interpretable form. At this point, the image resolution increases, the network repositions the objects in space, and refines its decisions [45].

FCN is a neural network based on the encoding–decoding process, does not contain fully connected layers, is capable of processing images of any size, and returns spatial outputs [47]. The key feature of U-Net compared to general FCNs is that it has a *Skip*

Connections element that connects the high-resolution feature maps of the encoder to the corresponding levels of the decoder. This step solves the localization problem lost during the pooling layer and restores geometric accuracy. This step makes the U-Net architecture a fitting choice, as it allows for a topologically correct and optimizable path matrix.

One of the main unique features of this task is that the input and output sizes do not match, as the input is an image representation and the output is a discrete graph structure. This situation differs from the general approach of U-Net, but regardless of this, the architecture can handle this difference by specifically designing the decoder step for this purpose. The model transforms RGB-based image representations into object- and cell-level predictions by first producing dense spatial features with U-Net and subsequently aggregating them onto a structured grid representation. This logic also appears in other studies, according to the FCN output can be of a different nature (e.g., sparser, more abstract, and more structured), and it is not only suitable for pixel-level representation [48].

In this study, the first element of the architecture used is the input layout image, which the model receives on three channels due to RGB color coding. The basic building block of U-Net is the double convolution block, which consists of two consecutive 3×3 convolution layers. During convolutions, the padding value is 1, so the number of spatial dimensions does not change during the block. At the first level, the network produces a 64-channel feature map that represents the low-level structural features of the input image. The first double convolution block is followed by max pooling, where the resolution is reduced by half, but the number of channels remains unchanged. This is followed by another double convolution block, where the number of channels doubles and the spatial resolution remains unchanged. This step is one of the basic principles of U-Net, where the reduction in resolution means that fewer spatial details are visible, but the increase in the number of channels allows for the parallel processing of multiple abstract patterns. We repeat this step four times, bringing the final number of channels to 1024. The principle of increasing the number of channels while reducing spatial resolution has appeared in early convolutional neural networks [49] and later became a common pattern in the encoder parts of many architectures [45,49,50]. The model turns around at the Bottleneck Layer level, which is the origin of the name U-Net. At this point, the resolution is doubled during an upsampling step, and the number of channels is reduced by half. Next, padding is performed because the pooling and upsampling operations can cause minimal discrepancies, and matching sizes are particularly important for concatenation. This is followed by a skip connection, where the feature maps from the encoder and decoder are concatenated. Finally, a double convolution block integrates global and local information, while the number of channels continues to decrease. After the final upsampling step, the conversion is performed on eight channels, as this is the number of objects contained in the images, and the result of the conversion is an object-level segmentation map. The final, discrete output grid is produced by interpolation (see Figure 8).

3.2.4. Training Configuration and Optimization

After obtaining the necessary data and initializing the model, we perform the training. The existing data is divided into training and test datasets in an 80–20% split. We compare the test dataset based on the ground truth grid data generated in Unity, so we can also consider the task as a supervised learning classification problem. Training is performed in the PyTorch (Version 2.10.0) framework with CUDA-based GPU support. During training, we use a multi-class cross-entropy loss function, which is a well-known selection for segmentation tasks. To calculate the loss function, we first need to convert

the logit values discussed in Section 3.2.1 using the softmax transformation, which returns class probabilities:

$$\hat{p}_{i,p,c} = \frac{\exp(z_{i,p,c})}{\sum_{k=1}^C \exp(z_{i,p,k})}, \quad c = 1, \dots, C, \quad (12)$$

where $\hat{p}_{i,p,c}$ is the estimated probability of class c ; $z_{i,p} \in \mathbb{R}^C$ is the logit vector predicted by the model for the p -th cell of the i -th sample. Cross-entropy loss for a single cell:

$$\mathcal{L}_{i,p} = -\log(\hat{p}_{i,p,y_{i,p}}), \quad (13)$$

where $y_{i,p} \in \{1, \dots, C\}$ is the class label for the p -th cell of the i -th sample. During an optimization step, the neural network learns on the *batch* subset, so the total loss must be determined at the batch level:

$$\mathcal{L} = \frac{1}{N \times |\Omega|} \sum_{i=1}^N \sum_{p \in \Omega} \mathcal{L}_{i,p}, \quad (14)$$

where N is the number of training samples in a batch, and Ω is the set of cells in the grid [51]. After defining the loss function, we selected the optimization algorithm. The model is trained using the Adam algorithm, which is straightforward to implement and is based on adaptive estimates of lower-order moments of the gradients [51]. Training took place over 30 epochs in order to log evaluation metrics. During the training loop, the predicted grid values differ in size from the ground truth matrix values, so the values are rescaled using bilinear interpolation. This step ensures that the loss function compares the predictions and ground truth labels on the same grid. Interpolation does not modify the model architecture, but only implements the output alignment used during training (see Figure 9).

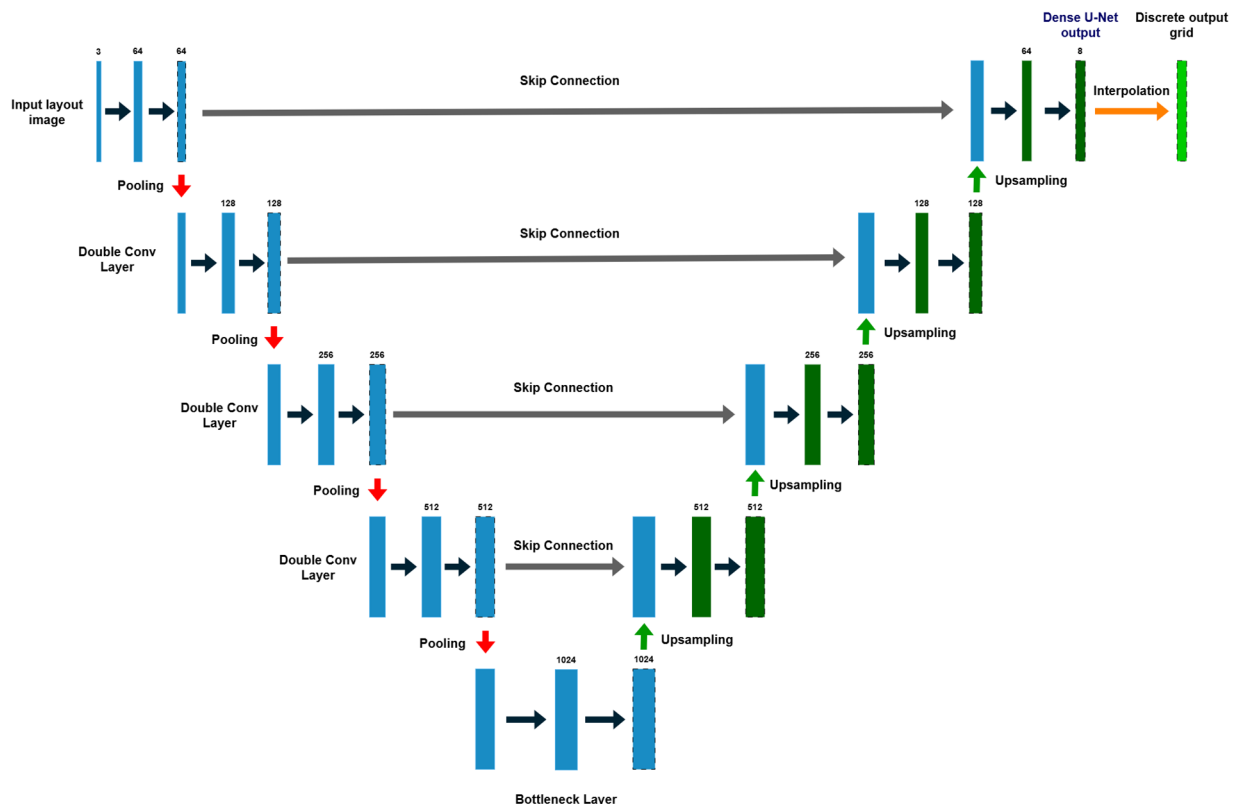


Figure 8. Specific U-Net architecture (source: own elaboration).

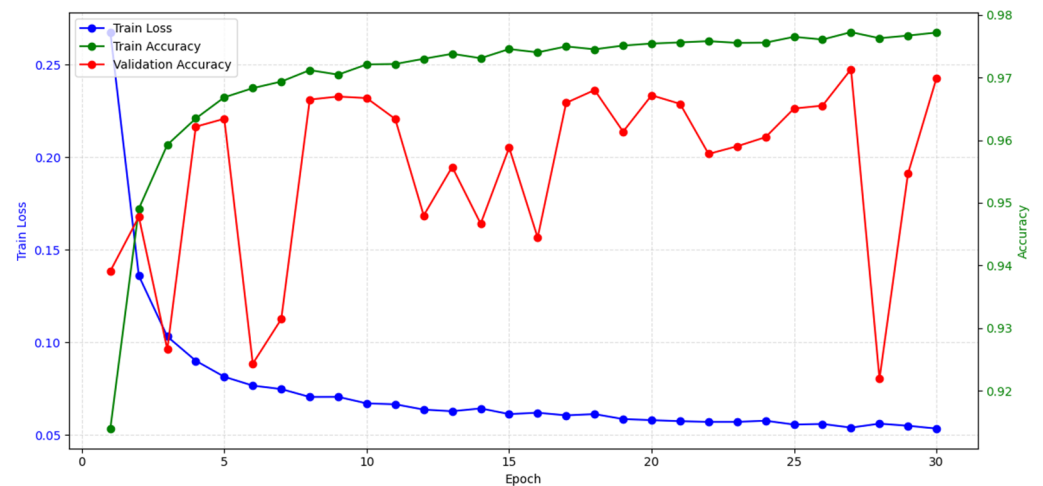


Figure 9. Loss and accuracy values of trained NN model (source: Python).

During training, the loss (blue) decreases monotonically over epochs, indicating that the optimization is consistent and the model fits the training data well. Train accuracy (green) increases to approximately 98% with no decline or training instability, while validation accuracy (red) ranges between 95 and 97%, but with higher instability and a sudden decline in some epochs. At first sight, this seems to be a significant problem, but due to the test dataset (200 samples), even a minimal number of wrong decisions can have an impact. We have examined the sudden validation accuracy drop in detail and identified that it is strongly influenced by the relatively small size of the validation dataset (200 samples). For instance, if 194 samples are correctly classified, the validation accuracy is 97%. However, if only three additional borderline samples are misclassified in a subsequent epoch, the number of correct predictions decreases to 191, resulting in a validation accuracy of 95.5%. We have also observed that in epoch 28, the mean validation accuracy was 97.27%, while the minimum per-image accuracy decreased to 89.72% for at least one validation sample. This indicates that a small number of difficult or ambiguous layout images (e.g., partially occluded shelves or visually similar empty-space regions) can temporarily reduce the overall validation performance. In summary, the evaluation of the model is positive, as train and validation accuracy values above 95% can be considered very good and can represent the warehouse layout images with sufficient quality.

For consistent evaluation of the model, it is advisable to use other metrics as well. For this purpose, we created numerical grid representations during layout generation using Unity in Section 3.1. so that we can compare the values predicted by the model with the original grid values, which can also be considered a supervised learning classification problem, and the same evaluation metrics can be applied. The figure below visually represents the prediction well: on the left are the values generated by Unity, in the middle are the model's predictions, and on the right are the matches (yellow) and differences (purple). The accuracy value in the example below is 98.3%.

Figure 10 clearly shows the quality and accuracy of the segmentation U-Net model in transforming image content. By grouping objects, it is possible to examine which objects the model has identified and at what level. The confusion matrix is a suitable choice for demonstrating this (see Figure 11).

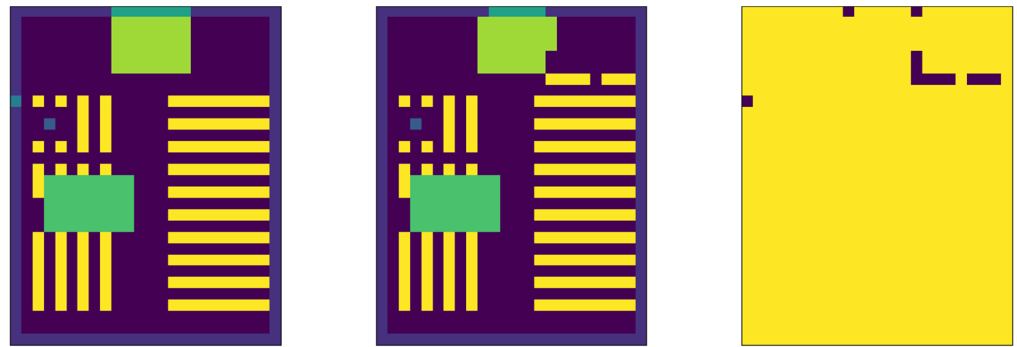


Figure 10. Generated and predicted grid values (source: Python).

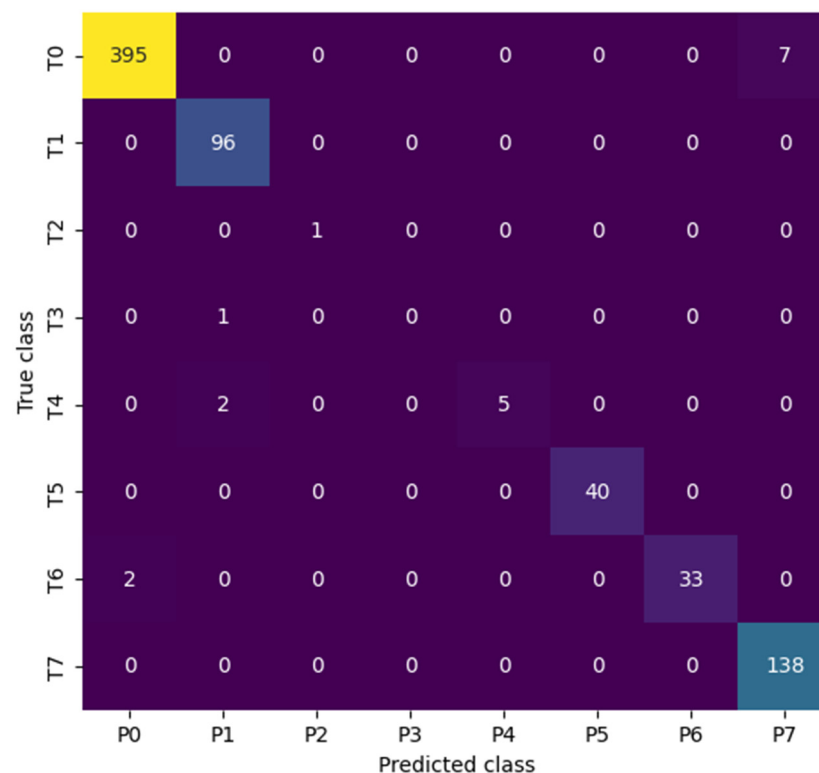


Figure 11. Confusion matrix (source: Python).

In Figure 11, most values are located on the main diagonal, which confirms the 98% accuracy value. The most significant errors of the model are that it failed to hit the 7-pixel shelf objects and identified empty objects. Regardless of this, it can be stated that the objects are identified with sufficient quality and are suitable for logistical optimization.

4. Optimization

Although the main goal of the research is to model layout images and segment objects, it is useful to examine the usability of the results obtained. In this chapter, we examine the optimizability of the predicted grid results and their applicability in the logistics industry.

For optimization, we first use the Breadth-First Search (BFS) graph search algorithm. To find a shortest path from s to v , we start at s and check for v among all the vertices that we can reach by following one edge, then we check for v among all the vertices that we can reach from s by following two edges, and so forth [51]. Let

$$\begin{aligned}
 G &= (V, E), \\
 V &= \{(i, j) \mid \text{grid}(i, j) \in \mathcal{W}\}, \\
 E &= \{((i, j), (k, l)) \mid |i - k| + |j - l| = 1\}.
 \end{aligned} \tag{15}$$

where V is the set of feasible cells. Each vertex corresponds to a grid point with coordinates (i, j) that satisfies the condition that the value of the given cell belongs to the set of passable cell types ($\mathcal{W}$). E is the edges between four neighboring cells. There is an edge between two vertices, (i, j) and (k, l) , if and only if they are orthogonally connected. The resulting graph is an unweighted graph, where each edge represents a movement with a unit step cost. The weight of every edge: $w(e) = 1$, so the target function is

$$\min_{P:s \rightarrow t} \sum_{e \in P} w(e), \tag{16}$$

where $s \in V$ is the starting point, $t \in V$ is the target point. BFS solves this problem, as layerwise traversal guarantees the optimal solution in a graph with uniform costs [52]. We benchmarked BFS runtime across the dataset by instrumenting the distance-matrix construction step. For each layout, BFS is executed for every node pair in the graph (picking point + shelf-adjacent target cells), i.e., $n \cdot (n - 1) / 2$ calls. In a representative layout with 196 nodes, this results in 19,110 BFS calls. The total BFS time for this layout was 8.81 s, while the average time per BFS call was 0.46 ms. This shows that individual BFS queries are computationally lightweight, and the overall runtime is dominated by the large number of pairwise BFS invocations required to populate the full distance matrix.

After determining the shortest partial routes, we define the notable nodes of the optimization problem. The starting point of the route is the picking area (object 5), the empty area (object 0) can be used for the route, and it is a necessary condition that every shelf is reached (object 7). The set of targets can be defined as follows:

$$T = \{(i, j) \in V \mid \exists (k, l) \in S : |i - k| + |j - l| = 1\}, \tag{17}$$

where V is the set of feasible cells, and it contains every grid point (i, j) where the route is available, S is the set of shelves that must be passed by 1 unit, so the cell and shelf are orthogonally adjacent. The BFS algorithm does not need to be executed only once, but rather for every pair of targets (T), so that we can obtain the distance matrix:

$$D_{i,j} = \text{dist}(t_i, t_j), \quad \text{dist}(t_i, t_j) = \text{BFS}(t_i, t_j), \tag{18}$$

We build the TSP-based route on the distance matrix, which requires that the starting point and destination match, and that each point t be visited exactly once:

$$\min_{\pi} \sum_{k=1}^n D_{\pi_k, \pi_{k+1}}. \tag{19}$$

Since the routing algorithm operates directly on the predicted segmentation grid, classification errors propagate to the graph construction stage. Misclassification of shelf objects may alter the target set T , potentially resulting in missing or incorrectly defined visitation nodes. Errors in obstacle or passable-area detection affect the feasible vertex set V , which may distort the connectivity structure of the graph $G = (V, E)$.

In practice, false positive obstacle detection may increase route length due to artificially restricted accessibility, while false negative obstacle detection may create computationally feasible but physically unrealistic paths. In extreme cases, segmentation errors could generate disconnected components, making certain targets unreachable and leading to infeasible TSP instances.

However, due to the high segmentation accuracy ($\approx 98\%$) and the limited spatial extent of misclassifications observed in the confusion matrix, no disconnected graph or infeasible routing instance was encountered in the experimental evaluation.

The TSP is an NP-hard problem for which there is no known polynomial-time algorithm that guarantees an optimal solution in all cases [2,3]. As a result, heuristic approaches are used in practice to solve the problem, as is the case with the method presented here.

We performed route planning and logistics optimization on the created layout image segmentation. Figure 12. also shows that the route was successfully created, which is a positive confirmation that the layout images are not only theoretically successful, but also actually suitable for industrial optimization.

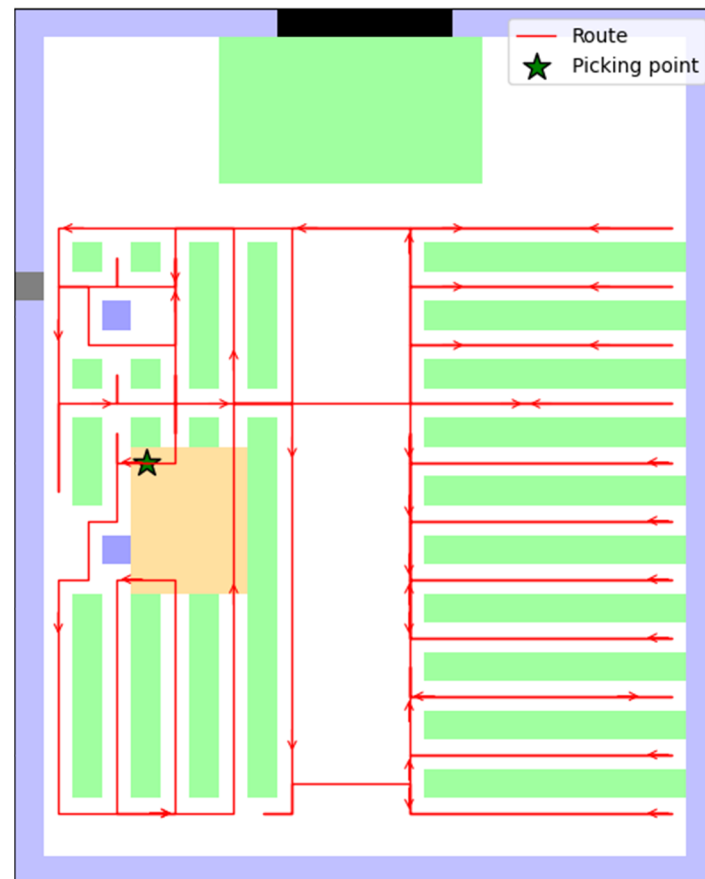


Figure 12. Optimized route (source: Python).

5. Conclusions

During the literature review, it was also found that the integration of artificial intelligence models into the logistics industry is popular and offers numerous opportunities, such as in the entire SCM process or in warehousing operations. AI models can be combined with Industry 4.0, IoT, and digital twin approaches, which can lead to innovative and modern methodologies, thereby reforming today's logistics processes.

One of the main categories of AI models is image processing, which transcends multiple industries and offers sustainable solutions. Research into its role in logistics is relevant and critically important, as despite innovative solutions, there has been little research published on the subject.

This research clearly demonstrates that methodologies for solving industrial problems can be developed using CNN architecture (see Figure 13). Image processing shows the quality and value of data that can be generated from images, which can be used to gen-

erate corporate value. It is important for every company to use its existing data, but the conversion of other types of information into usable data can be extremely valuable and support the company's operations and effectiveness.

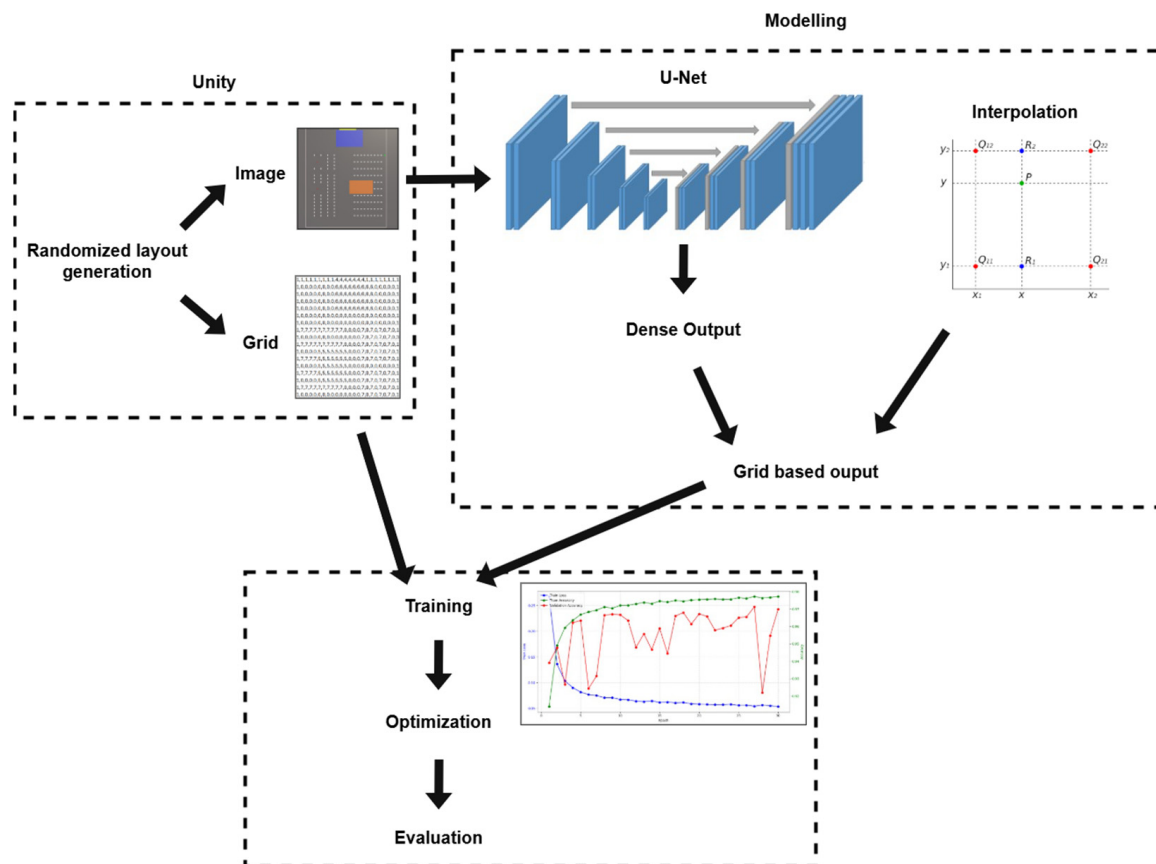


Figure 13. Our proposed framework (source: own elaboration).

By creating a general end-to-end framework, we received specific feedback on the challenges and opportunities of the entire process. During layout generation, we identified the main characteristics, important elements, and randomizable options of the raw material warehouses, and finally created a total of 1000 layout images and the corresponding grids (ground truth). As a next step, we processed the images and segmented the objects in them. During this step, we analyzed the possibilities of convolutional neural networks and selected the architecture best suited to the task. The main advantage of U-Net is that it uses the information stored in the encoding process during the decoding process, allowing local and global information to be combined by merging the two sources, resulting in a high-quality model. During training, we divided the generated images into an 80–20% split and trained the model over 30 epochs. Thanks to the ground truth data, we were able to perform a comprehensive evaluation, as image processing was transformed into a classification task, allowing us to analyze the predicted and actual results at the pixel level. During training, the train loss decreased continuously, and the train accuracy reached 98%.

Upon completion of the modeling, we focused on testing the predicted route in a logistics environment, as the validation of the theoretical model is particularly important and provides good feedback on usability. We built a round-trip planning algorithm on the predicted grid values, where we determined the starting and ending points, the accessible cells, and the warehouse points that the route must reach. The algorithm successfully created the optimal route, which is a good validation that the built model provides a usable end result, completing the end-to-end framework.

6. Discussion

The purpose of the research was to examine the possibilities of processing layout images, as we recognized during our literature review that the application of AI in the logistics industry offers several possibilities, but the number of published studies is small [5], especially in the field of image processing. The study created a complete end-to-end framework to ensure that our results could be fully integrated into logistics applications and not just offer a theoretical model. By generating layout images, fitting them into the model, evaluating them, and applying the optimization algorithm, we successfully created the framework, which was validated at the model level (98% accuracy) and at the practical level (algorithm creation).

The limitation of our research is that we used generated layout images, as there is no database available containing real images, since every company keeps its own layout images confidential and does not publish them. As a further development step, we decided to test the model on real, industrial CAD-based layout images in order to examine its usability.

The routing performance is inherently dependent on the accuracy of the layout segmentation stage. Since the grid representation used for BFS-based pathfinding is directly derived from the segmentation output, classification errors may propagate into the routing process. False obstacle detection can result in artificially blocked corridors, leading to infeasible paths or inflated travel distances. Conversely, missed obstacle detections may create unrealistically short routes that are not physically traversable in practice. Misclassification of shelf locations or picking points may further distort the distance matrix, ultimately affecting the quality of the TSP-based route optimization. Therefore, segmentation accuracy is a critical prerequisite for reliable routing results, and improving segmentation robustness directly enhances routing feasibility and solution optimality. As a future research direction, robustness mechanisms could be incorporated into the pipeline, such as confidence-aware grid filtering, post-processing validation of connectivity constraints, or probabilistic routing models that account for segmentation uncertainty. Integrating error-detection layers or enforcing structural consistency constraints may further reduce the propagation of segmentation errors into the optimization stage.

It is important to note that the routing formulation presented in this study is intentionally simplified to isolate and validate the layout-to-graph transformation framework. The current implementation assumes an undirected four-neighborhood grid with uniform traversal cost and static obstacles. In real-world warehousing environments, additional operational constraints may arise, such as one-way aisles, dynamic obstacles, congestion effects, time windows, multi-robot coordination, or capacity and task-priority constraints.

These constraints, however, primarily affect the definition of the edge set E , edge weights $w(e)$, or the structure of the optimization objective, rather than the image-to-graph transformation itself. The proposed framework is modular and extensible: directed edges can model one-way passages, time-dependent weights can capture congestion, and multi-agent routing can be incorporated by extending the optimization layer. Therefore, while the present study validates feasibility under simplified assumptions, the mathematical mapping from layout images to structured graph representations remains applicable to more complex routing formulations.

Although the presented study applies a new approach and introduces a general framework, our research does not only produce theoretical results. The main value of the methodology is that we were able to verify that the data we created can be used from non-primary sources, opening up new directions for use by teaching the model to various modern tools. The study proves that, using this approach, AGVs can plan navigable routes based on images for order picking or other warehousing processes, and drones can create routes for RFID identification processes. These examples are modern solutions and reflect

the digital twin approach well, but these processes could be even more modern by applying the methodology presented.

Author Contributions: Conceptualization, Á.F. and T.B.; methodology, T.B.; software, Á.F.; validation, Á.F.; formal analysis, Á.F. and T.B.; investigation, Á.F. and T.B.; resources, Á.F.; data curation, Á.F.; writing—original draft preparation, Á.F. and T.B.; writing—review and editing, Á.F. and T.B.; visualization, Á.F.; supervision, T.B.; project administration, T.B.; funding acquisition, T.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets and programming files presented in this study are available at https://github.com/francuzadam/article_layout (accessed on 2 February 2026).

Acknowledgments: The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AGV	Automated Guided Vehicle
AI	Artificial Intelligence
BFS	Breadth-First Search
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
CSV	Comma-Separated Values
DL	Deep Learning
FCN	Fully Convolutional Network
GA	Genetic Algorithm
GML	Geography Markup Language
GPU	Graphics Processing Unit
HRC	Human–Robot Collaboration
IoT	Internet of Things
LBS	Location-Based Service
ML	Machine Learning
MR	Mixed Reality
NN	Neural Network
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RFID	Radio Frequency Identification
RGB	Red, Green, Blue
RL	Reinforcement Learning
SA	Simulated Annealing
SCM	Supply Chain Management
TS	Tabu Search
TSP	Travelling Salesman Problem
UAV	Unmanned Aerial Network

References

1. Chen, W.; Men, Y.; Fuster, N.; Osorio, C.; Juan, A.A. Artificial Intelligence in Logistics Optimization with Sustainable Criteria: A Review. *Sustainability* **2024**, *16*, 9145. [[CrossRef](#)]
2. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W.H. Freeman: New York, NY, USA, 1979; ISBN 0716710447.

3. Toaza, B.; Esztergár-Kiss, D. A Review of Metaheuristic Algorithms for Solving TSP-Based Scheduling Optimization Problems. *Appl. Soft Comput.* **2023**, *148*, 110908. [\[CrossRef\]](#)
4. Albert, P.-W.; Rönqvist, M.; Lehoux, N. Trends and New Practical Applications for Warehouse Allocation and Layout Design: A Literature Review. *SN Appl. Sci.* **2023**, *5*, 378. [\[CrossRef\]](#)
5. Mahroof, K. A Human-Centric Perspective Exploring the Readiness towards Smart Warehousing: The Case of a Large Retail Distribution Warehouse. *Int. J. Inf. Manag.* **2019**, *45*, 176–190. [\[CrossRef\]](#)
6. Mohsen, A. Framework for the Design of Warehouse Layout. *Facilities* **2002**, *20*, 432–440. [\[CrossRef\]](#)
7. de Koster, R.; Le-Duc, T.; Roodbergen, K.J. Design and Control of Warehouse Order Picking: A Literature Review. *Eur. J. Oper. Res.* **2007**, *182*, 481–501. [\[CrossRef\]](#)
8. Drezner, Z.; Hamacher, H. *Facility Location: Applications and Theory*; Springer: Berlin/Heidelberg, Germany, 2004; ISBN 9783540213451.
9. Baker, B.M.; Ayeche, M.A. A Genetic Algorithm for the Vehicle Routing Problem. *Comput. Oper. Res.* **2003**, *30*, 787–800. [\[CrossRef\]](#)
10. Valda, D.T.; Yasuda, M.; Tagaro, J. Logistics Optimization: A Literature Review of Techniques for Streamlining Land Transportation in Supply Chain Operations. In Proceedings of the ICPR 2025: International Conference on Production Research, New York, NY, USA, 7–8 August 2025.
11. Gunasekaran, A.; Ngai, E.W.T. Decision Support Systems for Logistics and Supply Chain Management. *Decis. Support Syst.* **2012**, *52*, 777–778. [\[CrossRef\]](#)
12. Martinho, J.L.; Gomes, C.F.; Yasin, M.M. Information Technology and the Supply Chain Integration: A Business Executives' Context. *Int. J. Bus. Inf. Syst.* **2019**, *30*, 277. [\[CrossRef\]](#)
13. Mahmoudinazlou, S.; Sobhanan, A.; Charkhgard, H.; Eshragh, A.; Dunn, G. Deep Reinforcement Learning for Dynamic Order Picking in Warehouse Operations. *Comput. Oper. Res.* **2025**, *182*, 107112. [\[CrossRef\]](#)
14. Racha, B.; Yousra, E.K.; Mahdi, B.E.; Lotfi, S.; Soufiane, E.; Bachir, E.K. Artificial Intelligence in Logistic Warehousing: A Case Study on Stock Management Optimization. *J. Eur. Syst. Autom.* **2025**, *58*, 1995–2007. [\[CrossRef\]](#)
15. Waller, M.A.; Fawcett, S.E. Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. *J. Bus. Logist.* **2013**, *34*, 77–84. [\[CrossRef\]](#)
16. Choi, T.; Wallace, S.W.; Wang, Y. Big Data Analytics in Operations Management. *Prod. Oper. Manag.* **2018**, *27*, 1868–1883. [\[CrossRef\]](#)
17. Kritzing, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in Manufacturing: A Categorical Literature Review and Classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. [\[CrossRef\]](#)
18. Lasi, H.; Fettke, P.; Kemper, H.-G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [\[CrossRef\]](#)
19. Tang, Y.M.; Kuo, W.T.; Lee, C.K.M. Real-Time Mixed Reality (MR) and Artificial Intelligence (AI) Object Recognition Integration for Digital Twin in Industry 4.0. *Internet Things* **2023**, *23*, 100753. [\[CrossRef\]](#)
20. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [\[CrossRef\]](#)
21. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-End Training of Deep Visuomotor Policies. *J. Mach. Learn. Res.* **2016**, *17*, 1–40.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
23. Carbonneau, R.; Laframboise, K.; Vahidov, R. Application of Machine Learning Techniques for Supply Chain Demand Forecasting. *Eur. J. Oper. Res.* **2008**, *184*, 1140–1154. [\[CrossRef\]](#)
24. Min, H. Artificial Intelligence in Supply Chain Management: Theory and Applications. *Int. J. Logist. Res. Appl.* **2010**, *13*, 13–39. [\[CrossRef\]](#)
25. Kumar, D.; Kr Singh, R.; Mishra, R.; Fosso Wamba, S. Applications of the Internet of Things for Optimizing Warehousing and Logistics Operations: A Systematic Literature Review and Future Research Directions. *Comput. Ind. Eng.* **2022**, *171*, 108455. [\[CrossRef\]](#)
26. Mahmood, N.H.; Marchenko, N.; Gidlund, M.; Popovski, P. (Eds.) *Wireless Networks and Industrial IoT: Applications, Challenges and Enablers*; Springer: Berlin/Heidelberg, Germany, 2021; ISBN 9783030514730.
27. Soumpenioti, V.; Panagopoulos, A. AI Technology in the Field of Logistics. In *Proceedings of the 2023 18th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP 2023)*, Limassol, Cyprus, 25–26 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
28. Terra, A.; Riaz, H.; Raizer, K.; Hata, A.; Inam, R. Safety vs. Efficiency: AI-Based Risk Mitigation in Collaborative Robotics. In *Proceedings of the 2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, Singapore, 20–23 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 151–160.
29. Kalkha, H.; Khiat, A.; Bahnasse, A.; Ouajji, H. Enhancing Warehouse Efficiency With Time Series Clustering: A Hybrid Storage Location Assignment Strategy. *IEEE Access* **2024**, *12*, 52110–52126. [\[CrossRef\]](#)

30. Pikner, H.; Sell, R.; Karjust, K.; Malayjerdi, E.; Velsker, T. Cyber-Physical Control System for Autonomous Logistic Robot. In *Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 699–704.
31. Shamsuddoha, M.; Khan, E.A.; Chowdhury, M.M.H.; Nasir, T. Revolutionizing Supply Chains: Unleashing the Power of AI-Driven Intelligent Automation and Real-Time Information Flow. *Information* **2025**, *16*, 26. [CrossRef]
32. Miao, Z.; Cai, S.; Xu, D. Applying an Adaptive Tabu Search Algorithm to Optimize Truck-Dock Assignment in the Crossdock Management System. *Expert Syst. Appl.* **2014**, *41*, 16–22. [CrossRef]
33. Kalinov, I.; Petrovsky, A.; Ilin, V.; Pristanskiy, E.; Kurenkov, M.; Ramzhaev, V.; Idrisov, I.; Tsetserukou, D. WareVision: CNN Barcode Detection-Based UAV Trajectory Optimization for Autonomous Warehouse Stocktaking. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6647–6653. [CrossRef]
34. Tavakoli, M.J.; Fazl, F.; Sedighi, M.; Naseri, K.; Ghavami, M.; Taghipour-Gorjilaie, M. Enhancing Pharmacy Warehouse Management With Faster R-CNN for Accurate and Reliable Pharmaceutical Product Identification and Counting. *Int. J. Intell. Syst.* **2025**, *2025*, 8883735. [CrossRef]
35. Xu, D.; Jin, P.; Zhang, X.; Du, J.; Yue, L. Extracting Indoor Spatial Objects from CAD Models: A Database Approach. In *Database Systems for Advanced Applications*; Springer: Cham, Switzerland, 2015; pp. 273–279.
36. Jang, H.; Yu, K.; Yang, J. Indoor Reconstruction from Floorplan Images with a Deep Learning Approach. *ISPRS Int. J. Geoinf.* **2020**, *9*, 65. [CrossRef]
37. The Role of CAD in Modern Warehouse Layout Design. Available online: <https://ecseco.com/blog/the-role-of-cad-in-modern-warehouse-layout-design/> (accessed on 1 January 2026).
38. Warehouse Racking Layout & Design Experts. 3D Storage Systems. Available online: <https://www.3dstoragesystems.com/the-warehouse-layout-experts/> (accessed on 1 January 2026).
39. Warehouse Layout & CAD Design Services. American Surplus. Available online: <https://www.americansurplus.com/services/layout-design-cad-drawings/?srsltid=AfmBOoreqOPoOYeacMQRwdt1rvFMzzfocrwUMnl2c8Ap9fuRvl0RvgeR> (accessed on 1 January 2026).
40. Warehouse Layout. Available online: <https://www.inventoryops.com/consulting-services/services-provided/warehouse-layout.html> (accessed on 1 January 2026).
41. Warehouse Layout Design—AK Material Handling Systems. Available online: <https://www.akequipment.com/system/warehouse-layout-design/> (accessed on 1 January 2026).
42. Illuri Sandeep Logits vs. Probabilities: Understanding Neural Network Outputs Clearly. Available online: <https://illuri-sandeep5454.medium.com/logits-vs-probabilities-understanding-neural-network-outputs-clearly-0e86a4256a0e> (accessed on 3 January 2026).
43. Syrris, V.; Hasenohr, P.; Delipetrev, B.; Kotsev, A.; Kempeneers, P.; Soille, P. Evaluation of the Potential of Convolutional Neural Networks and Random Forests for Multi-Class Segmentation of Sentinel-2 Imagery. *Remote Sens.* **2019**, *11*, 907. [CrossRef]
44. Géron, A. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2023; ISBN 9781098125974.
45. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241.
46. Shibuya, E.; Hotta, K. Cell Image Segmentation by Using Feedback and Convolutional LSTM. *Vis. Comput.* **2022**, *38*, 3791–3801. [CrossRef]
47. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 3431–3440.
48. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
49. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556. [CrossRef]
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385. [CrossRef]
51. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980. [CrossRef]
52. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; The MIT Press: Cambridge, MA, USA, 2022; ISBN 9780262046305.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.