Truck Classification on Swiss Highways Using Vision and Weigh-In-Motion Systems

Oussama Zayene[1,*], Maël Vial[1], Marc-Antoine Fénart[2] and Beat Wolf[1]

[1] *iCoSys, HEIA-FR, HES-SO University of Applied Sciences and Arts Western Switzerland*
[2] *iTEC, HEIA-FR, HES-SO University of Applied Sciences and Arts Western Switzerland*

Abstract

Weight-In-Motion (WIM) systems are crucial for detecting vehicle overloads and preventing infrastructure damage. However, their accuracy can be influenced by environmental factors and sensor limitations. This study proposes a vision-based approach for classifying heavy vehicles using the YOLOv5 deep learning model, providing an additional layer to verify and support WIM system outputs. Experimental results demonstrated test accuracy ranging from 96% to 100% for all truck classes. These findings highlight the potential of the proposed approach to improve WIM system reliability.

Keywords

truck classification, weigh-in-motion, YOLO

1. Introduction

Context: The Federal Roads Office (FEDRO) has observed a major issue of overloading in heavy vehicles (>50 tons) based on data collected from the Weigh-In-Motion (WIM) station network. This overloading is a serious concern as it can cause substantial damage to transport infrastructure. The WIM stations generate annual statistical reports based on the collected data, with filters applied to exclude potentially inconsistent records and ensure data integrity.
Objective: Research mandates highlight the importance of carefully verifying WIM station data to assess the effectiveness of the applied filters. While annual WIM statistical reports remain reliable due to the large volume of vehicles providing weighted averages, extreme values continue to pose significant challenges. Without visual verification of the vehicles, it becomes difficult to accurately determine their shape, which in turn affects the reliability of the associated data.
Opportunity and Solution: This study presents a solution-oriented approach by proposing a comprehensive set of tools, including data acquisition, filtering, annotation, and truck classification. While a thorough comparison of related literature is important, this paper focuses primarily on the practical application of the proposed solution to address a real-world use case.

The approach involves several steps, from data acquisition to the alignment of visual and WIM outputs. The paper is organized as follows: Section 2 outlines the dataset creation process. Section 3 presents the vision-based approach, followed by an analysis of the results. Section 4 discusses the synchronization of visual and WIM data. Finally, the conclusion summarizes the findings and proposes directions for future research.

2. Dataset Creation

While the FDOT[1] vehicle classification scheme, proposed by the Federal Highway Administration (FHWA), is widely used in transportation research, it has been designed around truck regulations

AI days HES-SO '25 January 27–29, 2025, Switzerland
*Corresponding author.
oussama.zayene@hefr.ch (O. Zayene); mael.vial@master.hes-so.ch (M. Vial); marc-antoine.fenart@hefr.ch (M. Fénart); beat.wolf@hefr.ch (B. Wolf)
0000-0001-9529-925X (O. Zayene); 0000-0002-9307-7212 (B. Wolf)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1] Florida Department of Transportation, Traffic Data and Analytics, www.fdot.gov/statistics.

and classifications specific to the U.S., Canada, and Australia. Due to differences in truck shapes, categorization standards, and load regulations in Switzerland and Europe, a new dataset is needed to better reflect local conditions. This section outlines the process of acquiring and annotating this dataset.

2.1. Data Acquisition and Collection

The data acquisition process involved selecting the appropriate equipment, including a sophisticated IP camera and router, which were installed near an existing WIM station to monitor traffic at an undisclosed location on a Swiss highway. A highly-secured storage server was set up at our premises, connected to the IP camera via the D-Link router and utilizing DDNS services for remote access. The system was designed to capture real-time truck data, which was then used to train the machine learning model for truck classification.

To monitor the video feed from the camera, manage the footage and easily configure the recording settings, we chose to use the ZoneMinder platform [2]. This open-source solution offers an integrated suite of applications to capture, analyze, record, and monitor any CCTV camera connected to a Linux-based machine. It was configured to connect to both the IP camera and the storage server. The video recording was set up in two modes: DAYTIME and NIGHTTIME, to avoid capturing footage during the night. The maximum duration for a ZoneMinder event (video segment) was set to 30 minutes. More than 2,100 events were recorded in 3 months, totaling about 1,050 hours.

2.2. Data Characteristics and Statistics

Before starting data collection and annotation, we reviewed related work to identify the best classification scheme (number of classes), annotation methodology, dataset size, and AI models. The commonly-used FDOT classification scheme includes 13 vehicle classes, with 9 truck classes and 1 bus class. Datasets typically range from 1,200 to 7,106 images, with 80% used for training. Vehicle classification commonly uses deep learning methods, such as Convolutional Neural Networks (CNNs) and their derivative architectures [1, 2, 3, 4, 5]. Most annotation tools rely on graphical interfaces for manual labeling, which can be time-consuming and inefficient. These tools often follow standard formats such as PASCAL VOC like in [2] or COCO as in [3].

Characteristics: This study focuses on 11 out of 17 classes recognized by the FEDRO, selected for their significant traffic frequency. The frequency varies considerably between classes, with the most common being class '5_329' (5-axle trucks, GVWR[3] = 40 tons) at 20%. The second most frequent are trucks of type '2_219' (GVWR = 18 tons) and '4_326' (GVWR = 32 tons, Maximum Length = 16.5 meters), each with a frequency of 13%. Figure 1 shows examples of the heavy vehicle classes analyzed in this study. Although vehicles are clearly visible in the dataset images due to the high camera quality, several

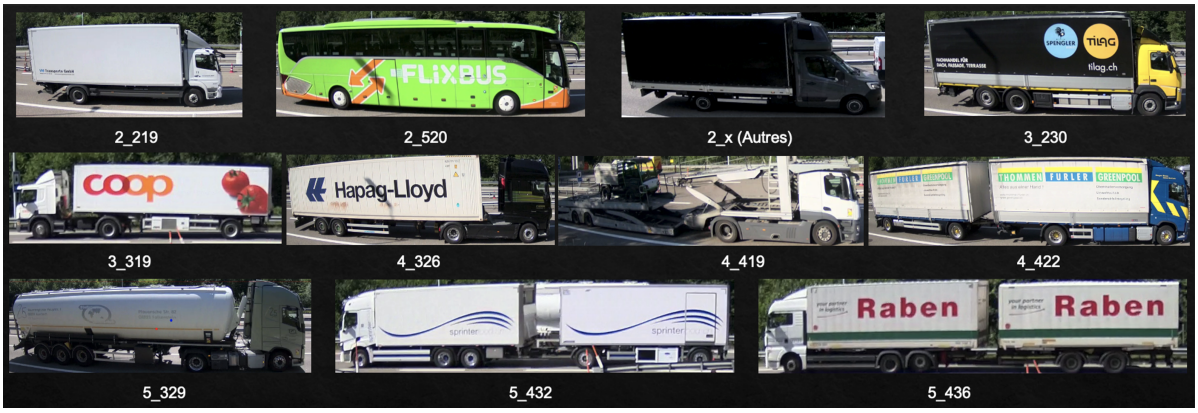

Figure 1: Examples of the 11 used heavy vehicle classes

[2] https://zoneminder.com/
[3] Gross Vehicle Weight Rating

scientific challenges have been identified, making the classification task more complex. The following outlines the main challenges to consider:

- *Inter-class similarity:* There is significant resemblance between certain truck classes, such as 2_219 and 3_230, 3_319 and 4_326, or 4_422, 5_432, and 5_436 (Figure 1).
- *Shadows:* Shadows directly impact wheel detection, as seen in classes 4_419 and 5_329 (Figure 1), which can affect classification accuracy, especially for vehicles with similar shapes where differences lie in the number of wheels. This issue is more pronounced with black vehicles partially in shadow, as seen in class 2_x (Figure 1).
- *Lighting variations:* Day-long recording (5:30 am to 9:30 pm) introduces lighting inconsistencies, particularly during early morning or sunset, affecting image quality. However, objects of interest remain generally visible and detectable.
- *Motion blur:* Detecting the actual shape of moving objects is challenging due to dynamic scene changes (e.g., other vehicles), lighting variations, and motion-induced blurring.

Statistics: A total of 3,616 'truck frames' were selected from images extracted from 70 hours of video recording. Additionally, 64 neutral images (no trucks, but cars or background) were collected. The dataset consists of 3,680 images, divided into 3 subsets: 3,130 for training, 414 for validation, and 136 for testing. The classes '5_329' and '2_219' dominate in quantity with 805 and 641 frames, respectively. For example, the number of frames in class '2_219' is twice that of classes like '3_230', '4_422', and '5_432', and four times higher than classes such as '5_436'. The least frequent classes are '4_419' and '2_520' with 92 and 77 frames, respectively.

2.3. Semi-automatic Annotation Methodology

YOLO_Label [6] tool was used to annotate the collected data. The generated GT file contains five elements per line: the Class ID and four spatial coordinates of the bounding box (BB). To further speed up the annotation process, we introduced a preliminary data filtering step to reduce the large number of frames extracted from the input video (typically around 30,000 frames per video). We leveraged the pre-trained YOLO object detection model on the COCO dataset, which includes 80 classes, among them trucks and buses. The result of this filtering step is a set of frames of heavy vehicles automatically detected. These frames are first manually verified to correct any misdetections and Class ID errors. Road trains (e.g., classes '4_419', '4_422', '5_432', and '5_436') are often poorly detected, typically resulting in a fragmented detection of the truck and its trailer. These cases are then processed with YOLO_Label [6] to adjust the positions of their BBs.

3. Vision-based Approach for Truck Classification

This section presents YOLOv5 [7] for truck classification, covering the model, experiments, and results.

3.1. Overview of YOLO for Real-time Object Detection

We have selected YOLO [7] due to its real-time detection capabilities, speed, and accuracy. Its end-to-end architecture employs a single-stage detector that predicts BBs and class labels simultaneously, ensuring efficiency and reduced computational cost. YOLO utilizes key techniques such as anchor box optimization, non-maximum suppression for post-processing, and multi-scale predictions.
The model has been fine-tuned on our custom dataset through transfer learning, adjusting only the final layer while retaining the robust feature extraction capabilities learned from COCO dataset.

3.2. Experiments and Results

Training was conducted with various model parameter configurations on a machine equipped with an Nvidia GeForce RTX 2080 Ti GPU. The optimal parameters for achieving the best results were a

batch size of 32, a normalized image size of 640 x 640, and 50 iterations. The resulting accuracy on the validation set ranged from 96.7% for class '3_230' to 99.5% for classes '5_329', '2_520', '3_319', '4_422', and '5_432'. For the test set, accuracy rates ranged between 96% and 100%, indicating good performance across both sets. Table 1 presents detailed quantitative results, showing high precision and recall across

Table 1
Detailed Results of Truck Classification on the Validation Set

Class	# Images	Precision	Recall	mAP@0.5	mAP@0.5:0.95
all	406	0.988	0.983	0.989	0.928
2_219	89	0.970	0.978	0.982	0.958
2_x	56	0.975	0.964	0.978	0.934
2_520	11	0.990	1.000	0.995	0.793
3_230	33	0.995	0.970	0.967	0.915
3_319	12	0.989	1.000	0.995	0.987
4_326	54	0.962	0.994	0.956	0.874
4_419	8	0.978	1.000	0.995	0.874
4_422	16	0.986	1.000	0.995	0.982
5_329	94	0.989	0.975	0.987	0.905
5_432	20	1.000	0.961	0.995	0.972
5_436	13	0.995	1.000	0.995	0.937

Note: 8 out of the 414 validation images are neutral and were excluded from the analysis

most categories, with the overall mAP@0.5 reaching 98.9%. However, performance slightly decreases for certain less frequent truck classes, often due to misclassifications of trucks into dominant categories, a common issue in real-world datasets. Despite this, the overall performance validates the strength of the model and supports the use of a vision-based approach for WIM data validation.

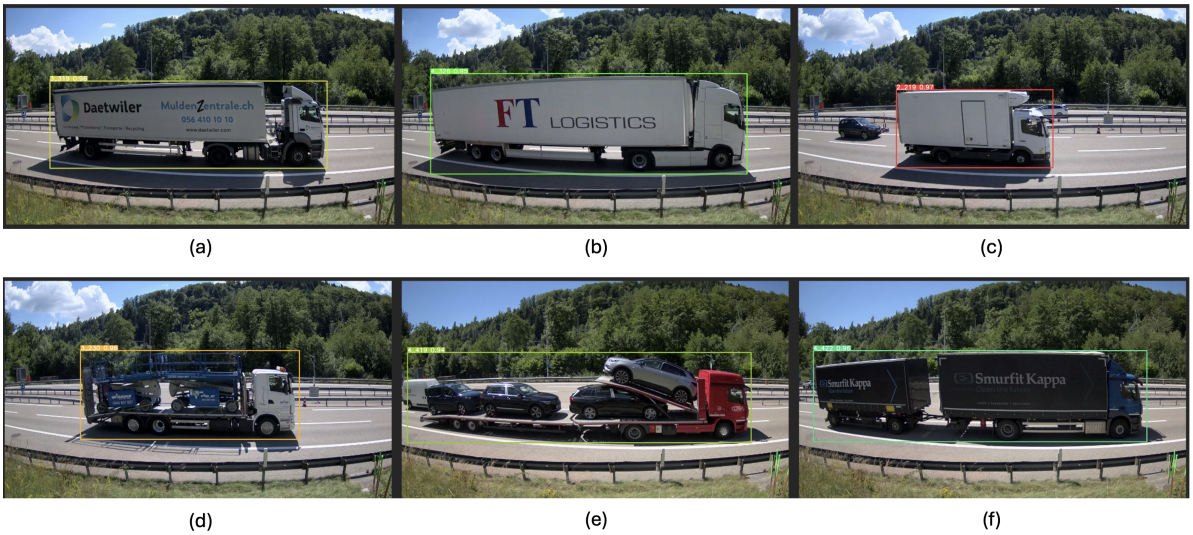

Figure 2: Classification results highlighting the model's ability to distinguish similar truck classes

These numerical results are further complemented by qualitative findings. Figure 2 presents visual results of the classification, demonstrating the model's ability to distinguish between similar truck classes, such as the differences between classes '3_319' and '4_326' in figures (a) and (b), respectively. The model also effectively filters out non-truck entities, such as cars, as shown in (c). Additionally, we tested the model's performance in more complex scenarios, such as towing—where trucks carry one or more vehicles—which often presents classification challenges. In most cases, YOLO delivers excellent results, as illustrated in (d)-(e). Furthermore, the model accurately identifies road trains, specifically the classes '4_419' and '4_422', as shown in figures (e)-(f).

4. Synchronizing Camera Detections with WIM Data

The trained model is applied to all monthly videos using a GPU-equipped machine with direct storage access to speed up processing. Each hour of video requires approximately 30 minutes to process. Given 15 hours of daily recording, the total processing time amounts to 9 days per month. Once detection processing is complete, a script merges and formats the CSV files from each video into a single output. This is achieved by exporting key metadata from ZoneMinder, including the date and start time of each recording. An example of the merged CSV output is shown in Figure 3.

EVENT_ID	DDMMYY	HHMM	SS	µS	TIMESTAMP_VIDEO [H-MM-SS.µS]	CLASSE
137	050722	0938	10	047711	0-02-28.047711	5_329
137	050722	0938	10	097700	0-02-28.097700	5_329
137	050722	0938	10	147700	0-02-28.147700	5_329
137	050722	0938	47	397122	0-03-05.397122	4_326
137	050722	0938	47	447111	0-03-05.447111	4_326

Figure 3: Example of a merged CSV containing truck detections per video

With the truck detections extracted, they are then aligned with corresponding data from the WIM system to enable validation and comparison between the two sources. Since the camera and the WIM sensor are not positioned at the same location, there is always an offset between their detections. To account for this, the sensor detection timestamps are adjusted to approximate the camera's timestamps. Multiple offsets within the range of [-1500ms, 1500ms] are tested per video to minimize matching errors. Further post-processing is required for the final matching, which will be addressed in future work.

5. Conclusions

In this study, we developed a vision-based approach using YOLOv5 to classify heavy vehicles. The model demonstrates strong performance in truck classification and vehicle filtering, enhancing real-time monitoring and WIM system reliability. While achieving high accuracy, challenges remain with highly similar classes, out-of-distribution vehicles, and varying weather conditions.

Future work will explore zero-shot Vision and Language Models (VLMs) to improve adaptability and recognize novel vehicle types without extensive retraining.

References

[1] L. Chen, et al., Identification and classification of trucks and trailers on the road network through deep learning, in: 6th IEEE International Conference on Big Data Computing, Applications and Technologies, 2019.

[2] P. He, et al., Truck and trailer classification with deep learning based geometric features, IEEE Transactions on Intelligent Transportation Systems (2021).

[3] A. Almutairi, P. He, A. Rangarajan, et al., Automated truck taxonomy classification using deep convolutional neural networks, Inter. Journal of Intelligent Transportation Systems Research (2022).

[4] J. Sun, J. Su, Z. Yan, Z. Gao, Y. Sun, L. Liu, Truck model recognition for an automatic overload detection system based on the improved mmal-net, Frontiers in neuroscience 17 (2023).

[5] P. Maleki, et al., Object detection for vehicles with yolo, in: IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics, 2024.

[6] Y. Kwon, Yolo_label, 2021. URL: https://github.com/developer0hye/Yolo_Label.

[7] G. Jocher, et al., ultralytics/yolov5: v4. 0-nn. silu activations, weights & biases logging, pytorch hub integration, Zenodo (2021).