MATHEMATICAL MODELING AND COMPUTING, Vol. 10, No. 2, pp. 518–523 (2023)

Road users detection for traffic congestion classification

Es Swidi A.[1], Ardchir S.[2], Daif A.[1], Azouazi M.[1]

[1] *Ben M'Sik Faculty of Science (FSBM), University of Hassan II Casablanca, Casablanca, Morocco*
[2] *National School of Commerce and Management (ENCG), University of Hassan II Casablanca, Casablanca, Morocco*

(Received 2 March 2023; Accepted 20 April 2023)

One of the important problems that urban residents suffer from is Traffic Congestion. It makes their life more stressful, it impacts several sides including the economy: by wasting time, fuel and productivity. Moreover, the psychological and physical health. That makes road authorities required to find solutions for reducing traffic congestion and guaranteeing security and safety on roads. To this end, detecting road users in real-time allows for providing features and information about specific road points. These last are useful for road managers and also for road users about congested points. The goal is to build a model to detect road users including vehicles and pedestrians using artificial intelligence especially machine learning and computer vision technologies. This paper provides an approach to detecting road users using as input a dataset of 22983 images, each image contains more than one of the target objects, generally about 81000 target objects, distributed on persons (pedestrians), cars, trucks/buses (vehicles), and also motorcycles/bicycles. The dataset used in this study is known as Common Objects in Context (MS COCO) published by Microsoft. Furthermore, six different models were built based on the approaches RCNN, Fast RCNN, Faster RCNN, Mask RCNN, and the 5th and the 7th versions of YOLO. In addition, a comparison of these models using evaluation metrics was provided. As a result, the chosen model is able to detect road users with more than 55% in terms of mean average precision.

Keywords: *traffic congestion; traffic jam; objects detection; machine learning; deep learning.*

2010 MSC: 68T45, 68U10 **DOI:** 10.23939/mmc2023.02.518

1. Introduction

Traffic congestion (TC) is a worldwide phenomenon, it has affected most cities, even small ones. It is caused by several reasons, including the increase in the number of vehicles on roads, the movement that the world has known in recent years, and the poor services provided by public transportation. Due to these, the concerned authorities required to find solutions. The losses because of this phenomenon are increasing rapidly. For example, according to transport data company INRIX. The cost of TC in the United States (US) was estimated at 87 billion dollars [1] in 2018.

Many approaches are employed to manage TC, one of these approaches is using records of the Closed-circuit television (CCTV) fixed on roads. It provides real-time records of roads, which makes data scientists anxious to exploit it. These CCTVs could be injected by detectors of mobile objects to classify road points as a congested point or non. Thus, such information could be useful to inform other users of the points of congestion. This solution can enhance the existing TC detectors that use GPS data.

Machine Learning (ML) and computer vision (CV) are fields which can help to improve the role of fixed CCTVs. One common way is using historical data, like images of vehicles, pedestrians, and

This work was supported by Information Technology and Modeling Laboratory (LTIM).

anomalies. The analysis of these traffic data allows for extracting features such as the average speed in a chunk of road, and the density; which means the number of users (vehicles and pedestrians) on a particular road. With these and other features, patterns that indicate congestion could be recognized.

The rest of the document is organized as follows. First, Section 2 of this paper represents related work. The third gives a description of our stereo vision system, then the dataset and the algorithms used in this study. Next, results and discussion section. Finally, the last section is a conclusion of this work.

2. Related work

Many different studies have touched on the topic of congestion detection using CCTVs. Several detection technics were used over the years to identify both road users: vehicles, and pedestrians. Regarding the advancement of Computer Vision. It can be considered as the prevalent and cost-cheaper compared with any other sensors.

In [2], 3k and drones cameras tools were used to capture roads. The researchers combine change detection approaches as a mask; all mobile objects are identified and represented in white color, while the fixed objects are identified as a background and represented in black color. In addition, this study provides a visual representation of the congestion level. According to the amount of white color in masks, roads are affected and colored by the level of congestion; if the color is near to the red color, that means we have congestion (see Figure 1).

The rectangle (a) represents a section of a road divided into four parts. Each part is illustrated in the images (b), (c), (d), and (e) to provide more details and clear visualization about the level of congestion.

In another works, researchers use Computer Vision technics, either in traditional computing or using the Artificial Intelligence (AI) especially ML and DL. In the first, developers try to describe images and identify objects in the entire image or in a region of the images. To this end, they employ descriptors such as Content-based image retrieval (CBIR) [3], Histogram of Oriented Gradients (HOG) [4], Local Binary Pattern (LBP) [5], and so on. Otherwise, ML and DL automate detection, for ML, each image will be converted to a numerical vector using the previous descriptors, then the result uses as an input of Fully Connected Layers (FCL). Furthermore, DL automates also features extraction step. Hence, to deal with images, developers need to choose an architecture of Convolutional Neural Networks (CNN) and feed it by a dataset of images with special parameters. Then, to choose the accurate model, training and evaluation tasks are required.

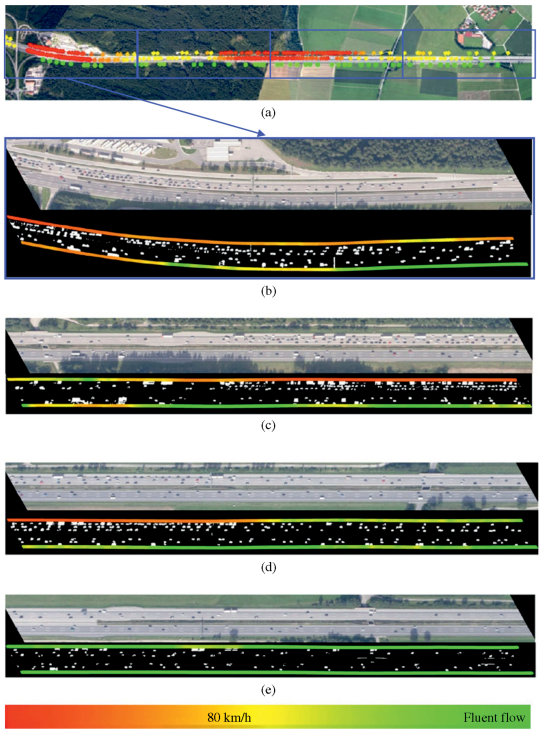

Fig. 1. Example of detecting traffic road congestion.

3. Methods

3.1. Proposed flowchart

Detecting road users is the main mission for traffic jam classification, by identification of mobile actors on sequence of frames, we were able to count number of pedestrians and the density in a road point; the number of vehicles considering road conditions. To this end, generally DL demonstrate its performance for object detection, or especially persons and vehicles. It is the most appropriate solution compared with traditional programming. With DL, we were able to build models to recognize, identify and track

road users in videos captured by CCTVs and cameras. The proposed flowchart (in Figure 2) shows the process to achieve this end, the process for providing a reliable traffic congestion system starts by collecting and preparing the data, training, and evaluating models using DL algorithms, and finally, adding services such tracker and density calculator.

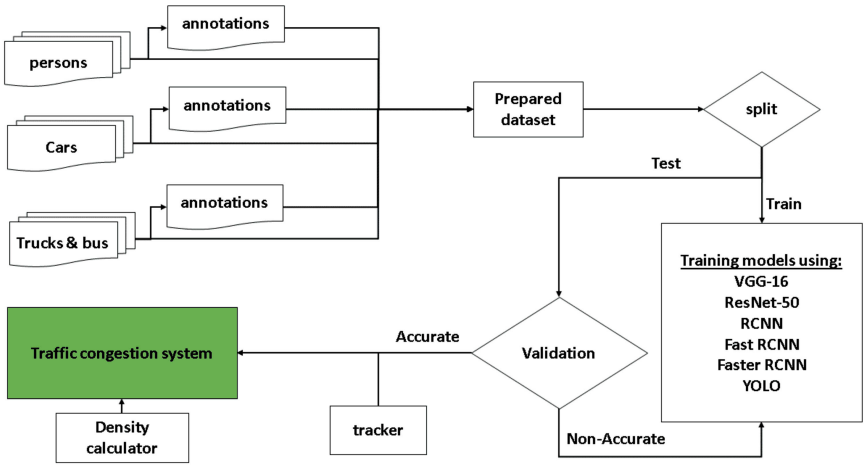

Fig. 2. Proposed flowchart for traffic congestion detection.

3.2. Dataset for road users detection

Real-time road congestion detection requires an accurate model. A model capable to detect mobile objects on road either vehicles or pedestrians with reliable precision, in order to achieve that, the model needs to be trained on a powerful dataset. Therefore, this study used the most popular MS COCO (Microsoft Common Objects in Context) [6] dataset. It is a large-scale object detection, segmentation, key-point detection, and captioning dataset. In this work, we were interested in vehicles and pedestrians detection. MS COCO allows us to prepare a custom dataset of images for four categories persons, cars, motorcycles/bicycle, trucks/bus. Each image contains multiple objects, these objects are localized and annotated in images, with bounding box and segmentation coordinates area. Table 1 shows the distribution of the custom dataset.

Table 1. Distribution of categories.

Category	No. images	No. labels
Persons	10777	66808
Cars	1918	12786
Motorcycles & Bicycle	3381	478
Trucks & Bus	6907	963

3.3. Data annotations

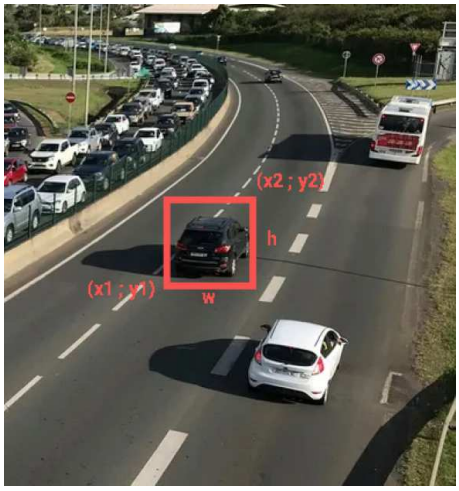

Fig. 3. Bounding boxes annotation.

To train supervised machine learning models, annotations or labeling the data is an essential task. Moreover, the quality of the model could be related to the quality of the data chosen, and also to the type of labeling. It varies according to the type and the structure of the dataset. Labeling is affecting tags (person, car, truck, ...) to each input observation. Annotate images to detect objects is not a common issue, especially that an image can contain multiple objects. Therefore, we were required to affect a label to each object in a single image. For this effect, there are several types of annotations including Bounding boxes for object detection, Polygonal Segmentation and Semantic Segmentation for image segmentation [7], 3D cuboids, and Key-Point and Landmark this type of annotation is useful for detecting emotions, human body

parts and poses. In this work, we were satisfied with Bounding boxes, we draw a rectangle around the object, this rectangle can be determined by either two coordinates (x1, y1) and (x2, y2) or by one coordinate (x1, y1) and width (w) and height (h) (see Figure 3).

3.4. Algorithms

To this end, using ML or DL is the most appropriate solution compared with other solutions. In traditional computing, developers were demanded to describe an image or a part of the image, then transform it into a numeric vector in order to apply mathematical functions, equations, and statistical analysis. With the proposed approach, we use the entire image as input without any computing, except the annotation task. DL automates feature extraction, and it allows removing some of the dependency on human experts. In general DL learning include three types of neural networks; Artificial Neural Networks (ANN) for numerical datasets, Convolution Neural Networks (CNN) initially where the proposed problem is related with images, and Recurrent Neural Networks RNN for sequential data or time-series data. According to the topic of the paper, we have used CNN and its algorithms as a solution, these approaches include Regions-CNN, Fast RCNN, Mask R-CNN, and YOLO.

— Region-Based CNN (RCNN) [8] is a proposed method to detect objects within 2000 regions in the frame, it uses the selective search algorithm to extract those regions. Furthermore, it enhances the mean average precision (map) by almost 30%. But it needs about 47 seconds to provide a result of a frame, which means that the method will not be doing well in real-time detection.
— Fast RCNN [9] instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map. Due to this reason, we do not need to pass 2000 region proposals to the fully connected layers.
— Faster RCNN [10] Similar to Fast R-CNN, the difference is that the Fast RCNN uses selective search algorithm on the feature map to identify the region proposals, while Faster RCNN uses Region of Interest (RoI) pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes.
— Mask R-CNN [11] is an object instance segmentation. It classifies each pixel into a fixed set of categories for example vehicle, pedestrian and so on. In addition to the category prediction and the bounding boxes, the Mask R-CNN outputs a binary mask for each RoI.
— The most relevant algorithm for object detection in recent years is known as YOLO [12–16], it is the abbreviation of "You Only Look Once". YOLO demonstrates its ability in term of both speed and accuracy. It allows real-time detecting of objects in images and their location with height precision. This approach has known rapid development and many versions, so that the latest version YOLOv8 was officially published in January 2023.

4. Results and discussion

In this section, we show the results of testing for proposed algorithms presented in methods section. This experiment used MS COCO dataset for the four categories (see Table 1). As known, to build a trusted model in ML or DL. It must pass through train and test steps: the first, in which we use 80% of the dataset to build from scratch our model. Generally, in DL this step is forward propagation; is the way from the input to the output layer. Moreover, an optional backward propagation. This last allows weights editing after loss calculation task. Besides, this step can include a test called validating. In the second step, we use 20% of the dataset to test the performance of the model built in training step. To this end, we use metrics to calculate the effectiveness of the model, such as precision, recall, and recall,

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN}, \quad \text{F1}-\text{Score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}.$$

In addition, MS COCO provide a metric obtained from the Intersection over Inion (IoU) values, this metric called mean Average Precision (mAP). Table 2 represents the results of the previous metrics

applied on the results of the algorithms used in this work. It is obvious that the YOLO versions achieve high results. YOLOv7 reached a mean precision of 0.53 and 0.55 of mean recall, and more than 55% of mAP on the testing dataset.

Table 2. Evaluation results of proposed algorithms.

Methods	Mean Precision	Mean recall	Mean F1-score	mAP-Val / mAP-Test
RCNN	0.36	0.27	0.3	36.5% / 39.8%
Fast RCNN	0.48	0.36	0.41	55.2% / 38.4%
FASTER RCNN	0.41	0.22	0.28	40.0% / 32.6%
Mask RCNN	0.37	0.42	0.39	41.8% / 43.9%
YOLOv5	0.52	0.48	0.5	61.4% / 47.3%
YOLOv7	0.53	0.55	0.61	58.4% / 55.6%

Furthermore, the reliability of a model can appear through the confusion matrix. It represents the number of predicted objects compared to the actual ones. According to Table 2, the appreciate results are achieved while using YOLOv7 algorithm. Therefore, we cited confusion matrix related of YOLOv7 results. Table 3 shows that the values of the diagonal are higher than other values, which means that most of the objects in images are correctly predicted. For example, the model predicts 7409 actual persons as a persons, and so on.

Table 3. Confusion matrix for YOLOv7 model.

		Prediction			
		person	car	motorcycle	Bus/truck
Actual	person	7406	367	2634	68
	car	73	1205	134	503
	motorcycle	748	698	1469	81
	Bus/truck	113	2780	254	3619

Next, predicting traffic road congestion is calculating the density of the road (i.e. number of vehicles and pedestrians considering road characteristics), this target is based on the quality of users detection on road, good traffic congestion classifier is a result of an accurate detector.

5. Conclusion

Detecting mobile objects on roads is essential to classify a road point if it a congested point or non. The main objects needed to be detected are pedestrians and vehicles including persons, cars, motorcycles/bicycles trucks/bus. A reliable detector allows us to determinate the density of roads. The density metric depends on the road characteristics and the number of mobile objects. Therefore, building an accurate model is the important task to solve the problem of traffic road congestion. In order to avoid using GPS or another confidential information about individuals, this study propose using records of public fixed CCTVs on roads. To this end, we compared six different algorithms of deep learning, these algorithms were trained and evaluated on a custom MS COCO dataset, we were satisfied only with four noticed classes. As a results, YOLO versions especially the 7th version demonstrate their performance, in which they reached 55.4% and 47.3% in term of mean Average Precision on test dataset for YOLOv7 and VOLOv5 respectively. Otherwise, RCNN family algorithms could not exceed 45%.

[1] Traffic congestion cost the US economy nearly 87 billion in 2018.

[2] Palubinskas G., Kurz F., Reinartz P. Model based traffic congestion detection in optical remote sensing imagery. European Transport Research Review. **2** (2), 85–92 (2010).

[3] Alsmadi M. K. Content-Based Image Retrieval Using Color, Shape and Texture Descriptors and Features. Arabian Journal for Science and Engineering. **45** (4), 3317–3330 (2020).

[4] Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). **1**, 886–893 (2005).

[5] Bhosle S., Khanale D. Texture Classification Approach and Texture Datasets: A Review. International Journal of Research and Analytical Reviews. **6** (2), 218–224 (2019).

[6] COCO — Common Objects in Context.

[7] Perreault H., Bilodeau G. A., Saunier N., Héritier M. CenterPoly: real-time instance segmentation using bounding polygons. 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). 2982–2991 (2021).

[8] Girshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. 580–587 (2014).

[9] Girshick R. Fast R-CNN, arXiv:1504.08083 [cs] (2015).

[10] Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems. **28**, Curran Associates, Inc. (2015).

[11] He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN, arXiv:1703.06870 [cs] (2018).

[12] Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640 [cs], arXiv: 1506.02640 (2016).

[13] Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. ArXiv:2207.02696 (2022).

[14] Long X., Deng K., Wang G., Zhang Y., Dang Q., Gao Y., Shen H., Ren J., Han S., Ding E., Wen S. PP-YOLO: An Effective and Efficient Implementation of Object Detector. ArXiv: 2007.12099 (2020).

[15] Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. ArXiv:2004.10934 (2020).

[16] Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. ArXiv:1804.02767 (2018).

Виявлення учасників дорожнього руху для класифікації заторів

Ес Свіді А.[1], Ардчір С.[2], Дайф А.[1], Азуазі М.[1]

[1]*Науковий факультет Бена М'Сіка (FSBM), Університет Хассана II Касабланка, Касабланка, Марокко*
[2]*Національна школа комерції та менеджменту (ENCG), Університет Хассана II Касабланка, Касабланка*

Однією з важливих проблем, від якою страждають жителі міст, є затори. Це робить їхнє життя більш напруженим, впливає на декілька сторін, включаючи економіку: витрачається час, паливо та продуктивність, крім того, психологічне та фізичне здоров'я. Це змушує дорожні органи шукати рішення для зменшення заторів і гарантування безпеки на дорогах. З цією метою виявлення учасників дорожнього руху в режимі реального часу дозволяє надавати функції та інформацію про конкретні точки дороги. Останні корисні для менеджерів доріг, а також для учасників доріг щодо місць заторів. Мета полягає в тому, щоб створити модель для виявлення учасників дорожнього руху, включаючи транспортні засоби та пішоходів, за допомогою штучного інтелекту, особливо технологій машинного навчання та комп'ютерного зору. У цій статті пропонується підхід до виявлення учасників дорожнього руху, використовуючи як вхідний набір даних із 22983 зображень, кожне з яких містить більше одного з цільових об'єктів, загалом близько 81000 цільових об'єктів, розподілених на людей (пішоходів), автомобілі, вантажівки/автобуси (транспортні засоби), а також мотоцикли/велосипеди. Набір даних, використаний у цій статті, відомий як Common Objects in Context (MS COCO), опублікований Microsoft. Крім того, було створено шість різних моделей на основі підходів RCNN, Fast RCNN, Faster RCNN, Mask RCNN, а також 5-ої та 7-ої версій YOLO. Крім того, було надано порівняння цих моделей за допомогою оціночних метрик. Як результат, обрана модель здатна виявляти учасників дорожнього руху з більш ніж 55% середньою точністю.

Ключові слова: *затори на дорогах; дорожній затор; виявлення об'єктів; машинне навчання; глибоке навчання.*