Real Time Vehicle Detection, Tracking, and Inter-vehicle Distance Estimation based on Stereovision and Deep Learning using YOLOv3

Omar BOURJA*[1], Hatim DERROUZ[2], Hamd AIT ABDELALI[3], Abdelilah MAACH[4],
Rachid OULAD HAJ THAMI[5], François BOURZEIX[6]
RIME Departement, Mohammadia School of Engineers, Mohammed V University in Rabat, 10100, Morocco[1,4]
Embedded Systems and Artificial Intelligence Departement, MAScIR, 10100, Morocco[1,2,3,6]
IRDA team, ADMIR Laboratory, Rabat IT center, ENSIAS, Mohammed V University, Rabat 10100, Morocco[2,5]
These authors contributed equally to this work[1,2,3]

***Abstract*—In this paper, we propose a robust real-time vehicle tracking and inter-vehicle distance estimation algorithm based on stereovision. Traffic images are captured by a stereoscopic system installed on the road, and then we detect moving vehicles with the YOLO V3 Deep Neural Network algorithm. Thus, the real-time video goes through an algorithm for stereoscopy-based measurement in order to estimate distance between detected vehicles. However, detecting the real-time objects have always been a challenging task because of occlusion, scale, illumination etc. Thus, many convolutional neural network models based on object detection were developed in recent years. But they cannot be used for real-time object analysis because of slow speed of recognition. The model which is performing excellent currently is the unified object detection model which is You Only Look Once (YOLO). But in our experiment, we have found that despite of having a very good detection precision, YOLO still has some limitations. YOLO processes every image separately even in a continuous video or frames. Because of this much important identification can be lost. So, after the vehicle detection and tracking, inter-vehicle distance estimation is done.**

Keywords*—Stereovision; stereo image; YOLOv3 deep neural network; convolutional neural network; vehicle detection; tracking; bounding boxes; distance estimation

I. Introduction

Today, there are millions of vehicles authorized on the roads and their number is constantly increasing. Consequently, traffic efficiency, reducing congestion and the human and material damage related to accidents, has become a major challenge in cities. However, this has been progressively improved in the last decade using ITS Intelligent Transport Systems (ITS). As a result, the incorporation of new information and communication technologies into vehicle interiors and transportation infrastructure has significantly revolutionized the way we travel today. These tools improve traffic flow by reducing travel time and congestion, detect road violations, support drivers, and reduce the risk of road accidents, and minimize the damage resulting from unavoidable accidents. These applications also impose demands, requiring credible dedicated hardware and reliable and timely communications. In addition, most traffic management systems are based on camera-based video surveillance because of their low cost, ease of maintenance, and ability to capture high quality images of the traffic scene [1]–[4]. This allows the dissemination and collection of useful information between vehicles, and transport infrastructure and vehicles to help drivers travel safely and comfortably.

However, these systems are reliable under normal conditions. In fact, they may not work very well in special circumstances, such as occlusion, bad weather, changes in lighting, and so on [1]–[3], [5]–[9].

In the present study, we are mainly interested in the calculation of inter-vehicle distance, which is an important traffic factor to be studied in intelligent transportation systems [2], [5], [8][1,5,6]. The aim of this paper is to present the possibility of using stereo cameras instead of LIDAR to estimate the inter-vehicle distance [1], [5]. Indeed, this method is advantageous because the recordings made by the cameras can be adapted for many algorithms. In addition, cameras are a much cheaper and therefore more cost effective solution compared to LIDAR [3]. Therefore, the images were captured by a stereoscopic system using two cameras placed above the traffic lanes. Indeed, our system consists of two slightly displaced cameras that obtain two images and go through a measurement algorithm based on the principles of stereoscopy in order to estimate the distance to the detected vehicles. The proposed solution is notably based on YOLO Deep Learning which will allow us to detect and delimit vehicles in a stereo image [7], [8].

The rest of the document is organized as follows. First, Section 2 gives a description of our stereo vision system. Then, we talk about calibration and stereo synchronization. In Section 3, we present our method of inter-vehicle distance estimation based mainly on the YOLO V3 Deep Neural Network algorithm and then we remove the general structure of our computational algorithm. In Section 5, a literature review on CNN models and YOLO model is given. The brief explanation of background of fully connected neural network is also provided. In Section 6 explanation of the experiment flow and methodology is given and overall system design is explained. Then, the test validation results, and the experimental results are explained. Finally, a conclusion is drawn.

II. Stereoscopic Vision System

A stereovision system is characterized by the acquisition of two images of the object to be observed from two different angles. After acquiring two images of an object from two different angles, the image coordinates of the points to be measured are determined on each of them. The matching of similar points is usually done automatically. The result is a list of 3D coordinates In fact, each of the two acquired images is processed by classical image processing tools to produce a list of 2D points characteristic of the objects. Each image having produced a list of points, a matching is then necessary to determine which points of the left and right lists correspond to each other. This matching can be based on a priori of the observed scene such as the preservation of the order of the elements from one image to the other. This constraint most often imposes to have two similar images or, in other words, to position the cameras very close to each other. Once the $(x, y)_left$ and $(x, y)_right$ pairs have been created, the calibration models can be used to calculate the corresponding (x,y,z) world points.

Therefore, the goal of stereovision is to calculate the spatial position of points from the coordinates of their images in two different views, in order to make measurements or to reconstruct the three-dimensional structure of the scene. So, the problematic of stereovision revolves around two essential points: camera calibration and synchronization. Indeed, at the shooting level, it is obviously necessary to obtain images of the same scene [3]–[5].

To acquire stereo images, we have designed a stereoscopic system with two similar cameras fixed and aligned on a stereo bar. The whole system was installed on a bridge over the highway, as shown in the following Fig. 1 [4], [5].

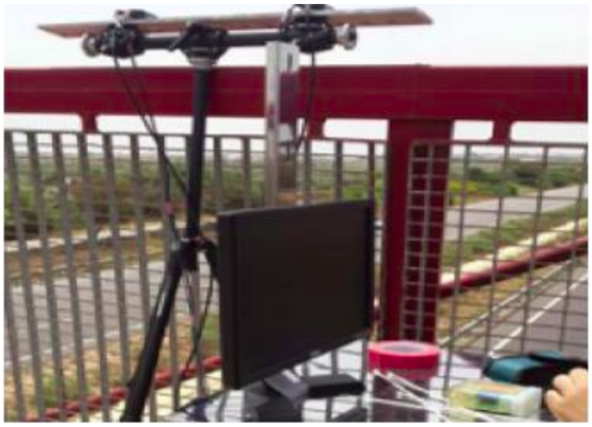


Fig. 1. Stereo System Installed on a Bridge over the Highway [1,8]. Stereo System Hardware Platform.

The calibration of a single camera (monocular applications) is equivalent to estimating its intrinsic parameters and its position in relation to the world reference frame. Configuring a stereoscopic sensor means calibrating both cameras (intrinsic parameters of each camera) and the relative position and orientation of the two cameras. Thus, calibrating a camera means estimating the transfer function that transforms a 3D point of the scene into a 2D point of the image.

Therefore, calibration is a very important step before the acquisition of the stereoscopic image [6], [10].

It allows to determine the intrinsic and extrinsic parameters of each camera. A bad camera calibration can influence the quality of the distance estimation. Therefore, the cameras must be installed accurately, otherwise measurement errors may occur (Fig. 2).

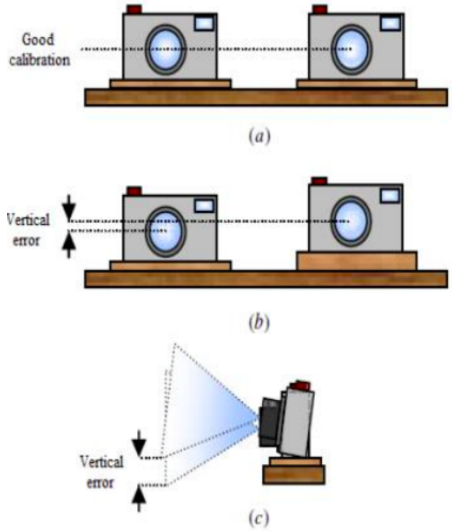


Fig. 2. Well-Calibrated Cameras (a), (b, c): the Most Common Calibration Errors.

The accuracy of the system depends on its correct calibration. Generally, we have two types of calibration [3]–[5]:

- Internal calibration to adjust the internal parameters of the camera (focal length, lens aperture, etc.) in order to eliminate image distortion. The intrinsic parameters of the camera are the projection of the optical center in the image frame, the focal length, and the image distortion parameters.
- External calibration to adjust the position and orientation of the two cameras in order to make their optical axes parallel. The extrinsic parameters are the translation and rotation between the camera frame and the world frame. They allow to position each camera in the same reference frame; to avoid any kind of non-calibration of both right and left image (Fig. 3).

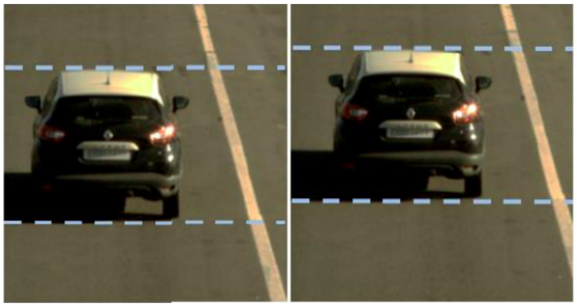


Fig. 3. Non Synchronized Pair of Stereo Images: Right Image is Captured After Left Image.

It should be noted that the use of stereo cameras in our system simplifies the calibration process since it is performed once and for all in the laboratory, whereas a monocular camera requires calibration for each road scene [3].

Thus, after calibration (2), we used a trigger card to constantly generate an electrical signal that will activate both cameras simultaneously to capture images at the same time [5]. Once the camera is activated, an image of the road scene is captured.

Note that the cameras must be correctly synchronized to obtain adequate results.

We will see in the following paragraph the detail of a stereovision method based on the geometry of the sensor seen previously. Fig. 4 illustrates the important parameters for stereoscopic measurements: S_L and S_R represent two cameras that are at distance B from each other.

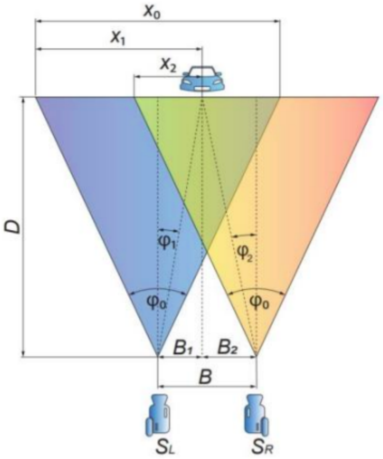


Fig. 4. Parameters for Stereoscopy Measurements. System Placed Horizontally on the Road [6].

φ_0 represents the field of view FoV of the cameras. The distance to the object (in our case a vehicle). D can be expressed by geometrical derivations leading to the following expression (equation 1) [6]:

$$D = \frac{B}{\tan\varphi_1 + \tan\varphi_2} \tag{1}$$

Where:
φ_1 and φ_2 are the angles between the axis of the camera lens and the direction of the object. After further derivation, we arrive at the following expression (equation 2) [6]:

$$D = \frac{B \times X_0}{2\tan\left(\frac{\varphi_0}{2}\right)(X_1 - X_2)} \tag{2}$$

X_0 is the number of horizontal pixels of the images, X_1 and X_2 are the numbers of pixels between the midpoint of the horizontal edge of the bounding box of the object and the left edge of the image (X_1 is for the left image and X_2 for the right one).

Finally, we can estimate the distance to any object appearing in both images if we know the distance between the cameras (B), the number of horizontal pixels of the image (X_0), the FoV of the cameras φ_0 and the horizontal difference between the same object in both images (X_1-X_2) also known as disparity. In fact, the disparity refers to the difference in image location of an object seen by the left and right cameras, resulting from the cameras' horizontal separation [5].

To calculate the actual distance between the stereo system and the vehicle, we need to calculate the angle between the road and the orientation of the system (Fig. 5).

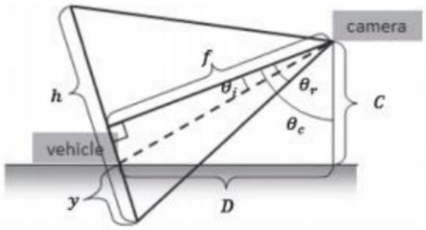


Fig. 5. Parameters for Stereoscopy Measurements. System Placed above the Road [2].

This distance is calculated as follows (equation 3):

$$D = D' \cos(\alpha) \tag{3}$$

Where:
- α: is the angle between the orientation of the system and the road
- D': is the distance between the object and the horizontal plane of the cameras. Then, the proposed method for estimating inter-vehicle distance involves three major steps. The first consists in preparing the two images generated by our stereovision system to detect the vehicles present in each image and to delimit them by bounding boxes [2], [8].

Then, the second step consists in finding the objects that appear respectively in the two images to precisely determine the value of X_1 and X_2 (cf. section 2) [6], [8]. Thus, if a vehicle is detected in the left image, the algorithm will have to search for it in the right image; the following criterion must be perfectly respected. This step is especially the most complicated of the whole algorithm.

The third step of the algorithm consists in calculating the distance between our stereovision system and each vehicle. At this stage, all the necessary parameters are already obtained, and the distance is calculated from the second equation. Finally, we deduce the inter-vehicle distance from the third equation by subtracting the estimated distance between each vehicle and the two cameras.

III. YOLO V3 Deep Neural Network Architecture

The heart of our distance estimation algorithm is the vehicle detection and recognition block, which allows us to locate and delimit vehicles in a stereo image by drawing a bounding box around the vehicles in the image. In this respect, we have opted for a variety of the YOLO Deep Neural Network (YOLOv3) algorithm for vehicle detection and recognition [4,5]. This neural network can recognize several objects in the same image, belonging to the same class or to different classes. In our case of study, we are mainly interested in the third version of the YOLO model, because it has the advantage of being able to run in real time on stereo images/video streams, while keeping a good predictive performance. This version has been developed by Joseph Redmon and researchers at the University of Washington [7].

The YOLOv3 algorithm is an improvement of YOLOv1 and YOLOv2 because it has advantages of high accuracy in detecting, recognizing, and locating objects as well as its speed of execution [7]; it has become a crucial point of current research. However, it still lags the most powerful object detection algorithms in terms of accuracy. Moreover, the principle of the model is to scan the image only once, by passing it through a deep neural network, hence the name YOLO (You Only Look Once) unlike methods based on CNN convolutional or RNN recurrent neural networks [7].

In addition, the latest version of the model has also focused on increasing the number of network layers as well as on the implementation of three scales of bounding boxes to detect smaller objects. These types of algorithms make it more possible to detect overlapping bounding boxes for the same object. The authors therefore apply a method called Non-Max Suppression to keep only the most significant bounding boxes [7]. After implementing and running our model, we obtain as output the bounding box coordinates of all detected vehicles. This information is very useful to obtain the parameters X_1 and X_2 which are used in the mathematical expression of distance estimation by stereovision [6]–[8].

A. Comparison of YOLO with Other Detection Algorithm

In comparison to recognition algorithms, a detection algorithm does not only predict class labels but detects locations of objects as well. So, it not only classifies the image into a category, but it can also detect multiple Objects within an Image [11]. It is extremely fast and accurate. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required [12]. And this Algorithm does not depend on multiple Neural networks. It applies a single Neural network to the Full Image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities [11] (Fig. 6).

IV. Literature Review

A. Evolution of Image Recognition

Particularly in recent years, image processing has come a long way. Evolution can be majorly seen in technological fields like computer vision and software. First computer vision and study on images started in 1960s. Before this, image analysis was done manually. The major improvement in deep learning techniques and in image recognition technology took place in 2010. Now it is so advanced that we can write a program for supercomputers to train themselves [13]. In early days, Feature extraction and classification paradigm was followed for object detection. Manually people need to define a specific feature which needs to be identified for extraction. After extraction of features, the objects or those features were represented in vector forms. These vector forms were used for training a model and for detecting an object while testing a model. It was a difficult task for detection of multiple objects since we have to find a general feature which can be found in multiple objects and can fit in different objects for training the model. The disadvantage is choosing the general feature which was a complex task, and the detection accuracy was not that great. In 2012, compared to other models which were already there, CNN gave a satisfactory results and good accuracy. Though there was CNN model developed in 1990s, the accuracy was low due to improper training examples and fragile hardware. CNN model became strong when GPUs were prevailing. The CNN model built in 2012 was trained by the dataset which consists of 1.2 million images and 1000 categories. The experiment conducted by Krizhevsky proved CNN's powerful ability in images classification [13]. CNN methods can build feature filters while training the process which cannot be done in traditional methods. When compared to other models, CNN models are more friendly and have self-learning ability [14]. Because of all these advantages, CNN became a major tool for image classification. To enhance the performance of CNN model,other regression heads were attached to the current model. This regression head is used to predict 4 coordinated after training it separately. Hence CNN allows both classification and regression head. While testing the model both classification and regression works simultaneously. Classification predicts the class score, and Regression helps for positioning. During 2012 to 2015, the experiments conducted were successful in attaching both classification and regression to CNN models Overfeat-Net, VGG-Net and ResNet. The error rate was reduced from 34% to 9% in these experiments. Since multiple object detection was failed in the experiments conducted in 2012, Research in 2014 started conducting experiments to achieve the task of multiple object detection. In a single image more than five objects were to be detected. This can be done only when the system figures out object's class and location of the object. Usually, deep convolution neural network works with the fixed size of image (e.g. 520 x 520). Because of this recognition accuracy might go low for the images and subimages of arbitrary size or scale [15]. To overcome this issue Spatial Pyramid Pooling was introduced in 2014. Fixed length representation regardless image size or scale is achieved by developing a network structure called SPP-net. By removing the size or restriction, accuracy of the convolution neural network can be achieved. In SPP-net feature maps are computed only once to generate fixed length representations to train the detectors by pool features in the sub-images. Repetition of computation of convolution features can be avoided by this method. This method was better than R-CNN and it gave satisfactory accuracy. Most of the ideas regarding CNN approach and classification came out in 2014. The main idea was to perform classification on every region that possibly contains objects. The region proposals and classification approaches achieved high accuracy and precision. But these region proposals take a very long time to process which makes the speed of the entire system to go low. Because of this timeconsuming limitation, the region proposal approaches cannot be deployed in applications which are time critical like autodriving, surveillance systems etc. Recently, YOLO (You Look Only Once) a unified object detection model was proposed by Joseph [7]. Frame Detection in YOLO is considered as regression problem. It is a pretrained model which does not require a dataset to train the model. It consists of weights and object detection is done as boxes. The image which is inputted is regressed to tensor from the model directly which signifies the digit of every object's position and class score of the object. The images which are inputted need not go through the YOLO network more than once. Because of this, processing of images is faster in this model. When compared to other object detection models, Yolo has

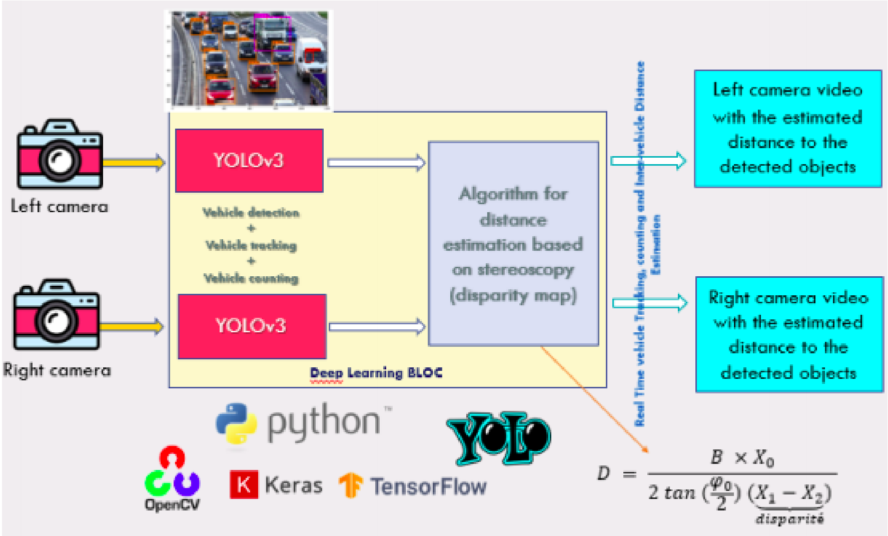

Left camera
Right camera
YOLOv3
Vehicle detection
+
Vehicle tracking
+
Vehicle counting
YOLOv3
Deep Learning BLOC
Algorithm for distance estimation based on stereoscopy (disparity map)
Real Time vehicle Tracking, counting and Inter-vehicle Distance Estimation
Left camera video with the estimated distance to the detected objects
Right camera video with the estimated distance to the detected objects
python
YOLO
OpenCV
Keras
TensorFlow
$D = \frac{B \times X_0}{2\tan(\frac{\varphi_0}{2})\underbrace{(X_1 - X_2)}_{disparité}}$

Fig. 6. Two-Camera/ YOLO Real Time System. Algorithm Architecture.

accomplished more than 50 times better accuracy. So currently YOLO is one of the best choices for real time object detections [7].

B. CNN based Object Detection: Benchmarking

1) R-CNN: R-CNN stands for Region-based Convolution Neural Network. It combines region proposals with Convolution Neural Networks (CNN). R-CNN aids in focusing objects with deep neural network. It trains a model of high capacity with fewer amounts of annotated detection data. To categorize the object proposals deep convolution network is used and due to this RCNN attains outstanding accuracy for object detection. Ability of R-CNN is high because numerous object classes can be scaled without resorting to estimated methods together with hashing [16]. The researchers projected a multi stages purpose followed by classification. And classification was done using regions paradigm. The three main components of the developed system is feature vector extraction by CNN, classifier used which is Support Vector Machine and the last one is region proposal component [17]. Feature vectors extracted from CNN are used to train the SVM classifier. Training is done on two datasets where CNN supervised is trained on one large dataset (ILSVRC) and one small dataset (PASCAL). During the testing time, the region proposal component used in this experiment is Selective Search. 2000 fixed size category independent regions which contain objects is produced by Selective Search [14]. SVM is used fot domain specific classification after a completely trained extractor of CNN converts every potential vector into feature vectors. The two main problems that may arise are intersection-overunion (IOU) and duplicate detections.

IOU will overlay the higher scoring region. These problems are eliminated by greedy non maximum suppression and refining the bounding box by using a linear regression model at the end. Satisfying accuracy for detection was accomplished by RCN when compared to any other detecting methods found in 2014. But RCNN also has many drawbacks because of complex multi-stage pipeline. The main role of CNN is to act as a classifier. The region prediction is totally depended on exterior region proposal methods. This slows down the whole system while both training and detecting objects. Since RCNN has a separate training manner for every component which results in CNN, it is very difficult for optimization. Besides, CNN cannot be updated during the training of SVM classifier [14].

2) SSD: Single deep neural network is used for detecting the objects in images by Single Shot Detector (SSD). The output spaces of bounding boxes are varied in SSD method. These boxes are set of default boxes over different aspect ratios. The approach is scaled to every feature map location after it varies. The predictions from multiple feature maps are combined in Single shot detector. Multiple feature maps are combined to handle objects of different sizes naturally. Some of the benefits of SSD are SSD totally removes the proposal generation. The following pixels or feature resampling stages are also eliminated which encapsulates every computation in a single network. Training in SSD is easy when compared to other models and it is forthright to assimilate into systems which needs a detection component. SSD accuracy can be increased by adding an additional method for object proposals. Since it is combined with other models, the training and inference is much faster.

3) R-FCNN: R-FCNN stands for region-based, fully convolutional networks. It is a simple framework used for efficient and accurate object detection. The other region-based network detectors like FCNN and Faster RCNN [18], are based on per region sub network. But R-FCNN is entirely convolutional with every computation shared on the whole image. There is a predicament between image classification and object detection. Image classification has translation invariance issues and object detection has translation variance issues. To overcome this issue positive sensitive score maps are proposed. Thus, region based fully conventional network can accept fully conventional image classifier like latest residual networks for detection of object [14]. PASCAL VOC datasets are used to show the modest results. ResNet with 101-layer is used. The results achieved by RFCNN are 20x better and faster than faster RCNN while both inference and training

4) Faster R-CNN: Faster region-based convolution neural network is similar to RCNN which is an object detection algorithm. The features are extracted from the input image through convolution layers. Region proposal network (RPN) is used in Faster RCNN which shares the convolution features for each spatial location like objectness classification and bounding box regressor. The FRCNN network is cost effective than RCNN. It basically predicts the object boundaries and objectness scores for every position of the object. High quality region proposals are created, and end-to-end training is done then this technique is used by Fast RCNN method for object detection. When compared to other object detection methods, faster region-based convolution neural network has less running time for detection of object. When feature maps are sent into RPN, feature maps projected region proposals are extracted. RoI pooling is done on feature maps. The result of Faster RCNN classification will be multiclass classification and bounding box regressor for each RoI [18].

5) Fast R-CNN: Fast RCNN stands for Fast Region Based Convolution network. It is a training algorithm for detection of objects. Fast RCNN is better than RCNN and SPP net as it resolves almost all disadvantages and increases the speed and accuracy of RCNN and SPP net. When compared to RCNN and SPP net, Fast RCNN has higher detection quality that is mAP. Training in Fast RCNN is done in single stage by means of multi-task loss. All the network layers can be updated during the training process. Disk storage is not utilized for feature caching by fast RCNN. The Convolution feature map is of Deep Convolution Network and RoI projection. The RoI pooling layer is extracted from the convolution feature map in RoI feature vector. RoI feature vector is extracted for each RoI.The output will be softmax and bbox regressor [11]. This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection [18]. By using the work of algorithms which are built previously, fast RCNN uses deep convolution network to classify object proposals efficiently. This helps Fast RCNN to achieve better detection accuracy and increase training and test speed. The training done by fast RCNN on deep VGG16 network is 9x faster than RCNN and 213 x faster when compared to the test time. A good mAP on PASCAL VOC 2012 is achieved. When Fast RCNN is equated with SPPnet, test accuracy is 10x faster and accurate and training of VGG16 is 3x times faster than SPPnet. Because of the detailed work carried out in this experiment, new insights are provided. The improved detector quality is achieved at the end. The main issue with other object detection algorithms is they are too expensive in time analyze in the past.

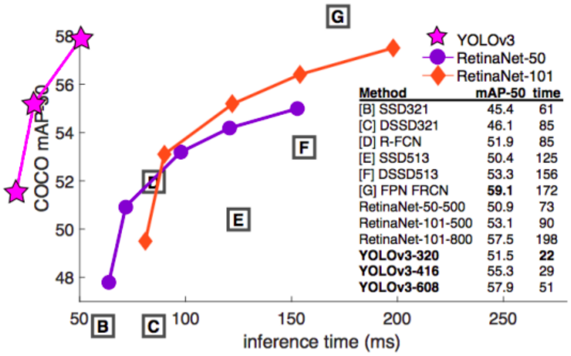
6) SPP-net: CNN models work only with the fixed size of input image like 520x520. Because of this the recognition accuracy will go low. To overcome the above-mentioned issue Spatial Pyramid Pooling was equipped. The fixed length of symbolization irrespective of size or scale can be generated by a Spatial Pyramid Pooling (SPP-net) network structure. Object deformation can be achieved by Spatial Pyramid Pooling. When compared all CNN based methods, Spatial Pyramid Pooling is an improved structure. Feature maps for the whole image can be computed at once in Spatial Pyramid Pooling method. Pool features in sub images of fixed length is also computed to train the detectors. In the other methods, convolution featured are repeatedly computed which can be overcome in Spatial Pyramid Pooling. SPP-net is more weighted in object detection. When compared to RCNN method, SPP-net is 30-170× faster and when both the models were tested on Pascal VOC 2007, SPP-net gave better accuracy than RCNN.

C. Real Time Objects Detection and Tracking: Benchmarking

Moving object detection and tracking is presented in [16]. Intuitive graphic interphase is achieved by means of new algorithm during the extraction of Silhouette. For the fast detection following algorithms were combined, frame difference method, background subtraction method, Laplace filter and Canny edge detector. The multivision dataset is used for testing the sequence images. The better performance object tracking algorithm is proposed. The detection algorithms and basic operation techniques are integrated, and graphic user interface is used to make the process simple and straight forward. World is adapting to artificial intelligence from past few years with influence of deep learning. Many object detection algorithms have been compared like Region-based Convolutional Neural Networks (RCNN), Faster RCNN, Single Shot Detector (SSD) and You Only Look Once (YOLO). And the result id faster RCNN and SSD gives better accuracy with Yolo. Efficient implementation and tracking are done by combining deep learning with SSD and mobile nets. SSD helps in detecting the object and tracking them in a video sequence. They achieved in enabling good security utility for enterprise and order. The model created can be deployed in drones, detect attacks and CCTV cameras in government offices, colleges, hospitals etc. Distance and estimation of real time video is achieved in [10]. Combinations of two deep learning models are developed to achieve object detection and tracking. The algorithms are tested on both railway and environment. Monodepth algorithm is applied for the estimation of object distance. Stereo image dataset and monocular images are used to train the model. Testing of both the models is done on another two datasets. They are Cityscape and KITTI datasets. Pedestrian and vehicle behavior tracking is done by developing a new method-based SSD. The new SSD algorithm is developed by the coordinates of the output bounding boxes of SSD algorithm. The whole development is tested on the real time data and the main objective is to monitor the tracks of pedestrians and vehicles to make sure it does not lead to any dangerous situations. Real time video of Routen tramway is taken by embedded cameras.

D. YOLO Model

YOLO (You Look Only Once) a unified object detection model was proposed [7]. Detection in YOLO is considered as regression problem. It is a pre trained model which does not require a dataset to train the model. It consists of weights and object detection is done as boxes. The image which is inputted is regressed to tensor from the model directly which signifies the digit of every object's position and class score of the object. The images which are inputted need not go through the YOLO network more than once. Because of this, processing of images is faster in this model. When compared to other object detection models, Yolo has accomplished more than 50 times better accuracy. So currently YOLO is one of the best choices for real time object detections. The base YOLO model can process real time images up to 45 frames per second whereas Fast YOLO processes can process nearly 155 frames per second. The base version is the smaller version of the network. The natural images can be generalized very well using this model. According to recent studies, YOLO is one of the fastest detecting models when compared to other CNN object detection models [7] (Fig. 7).



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Fig. 7. Comparison to Other Detectors [7].

E. Kalman Tracking

In recent decades real time object tracking has been applied in multiple areas like human computer interaction, security, surveillance, video communication, etc. Object tracking is the process of locating one or multiple moving objects in the scene during continuous time. Some of the challenges faced are, Initial moving object segmenting. The goal of segmentation is to simplify or change the representation of the image into something that is more minimal and easier to analyze [12]. Rapid appearance changes are caused by image noise, illumination changes, non-rigid motion, and different poses. Tracking the moving target is complex in background. When tracking an object in real world background can be quite complicated for various depths in the background which can interfere their tracking. So Kalman filter was introduced which is also called as linear quadratic estimation. It is an algorithm which uses the series of observing measurements over time [13].

There are two main parts that contributes in Kalman tracking. They are Prediction and correction. Prediction will predict the project current state and estimate the next state. If there is any mistake in prediction, it goes to correction. In correction, Kalman gain is computed. The system state is updated after Kalman gain is found and error covariance is also updated. Correction is in turn connected to prediction. The detecting range can be predicted by Kalman filter in order to accurately track object in occlusion which means a complicated background [12].

F. Unified Detection Model: YOLO

Earlier detection models repurpose the classifiers to achieve detection. The model is applied to many location and scales in the image. If there is a maximum scoring region on the image, then it is detected. But YOLO has an entirely dissimilar technique. A single neural network is applied to a whole image.

In Yolo model, network spits the image into regions. After the splitting, bounding boxes and probabilities for each region is predicted. The predicted probabilities help to weigh the bounding boxes [15]. When compared to classifier-based methods, YOLO has several advantages. The predictions are made by the global context in an image because Yolo takes whole image for the testing. RCNN requires thousands of predictions for a single image. But in YOLO the predictions are made by the single neural network assessment. Because of this yolo is tremendously fast. It is 1000 x times faster than RCNN and 100 x times faster than Fast RCNN. Thus, Training of YOLO model can be done in two ways. One can either use their own dataset to train the model or can use the pre-trained weights. These pretrained weights are available for public use.

Some of the dependencies required to build a YOLO in Tensorflow:

- Tensorflow/Keras (GPU version preferred for Deep Learning)
- NumPy (for Numeric Computation)
- OpenCv (for Image Processing)
- IPython (for displaying images)
- Glob (for finding pathname of all the files)

Anaconda is suggested as it contains many libraries of machine learning and deep learning and interaction with Spyder, and Jupiter are easier.

V. Methodology

A. Object Detection

There are mainly two ways of object detection. First one is to take object images and train our own Machine Learning model. When we train the machine learning model, main input is features. Based on the features, the model will learn and create weights for that object. But there are some disadvantages in this method. For example, when we consider the object as car, there are different types of cars based on their shapes. Sometimes even a truck might look like a car in the video. To overcome this issue, the feature extraction must be very much robust. Like the model should be trained by all the aspects like size, dimension, and shape. This requires a large Data. Because of this training will depend on our system. If the system's GPU is low, then we cannot train our model at all or it might take a very long time to process. If we go for SVM, neural networks or Random Forest models or any basic type of modeling which takes less amount of data, it does not work with the real time data. So, we use a YOLO model which can be defined by a concept of convolution neural network. The one difference between YOLO and other CNN models are, YOLO has a moving or floating window. That means a window is created in YOLO which keeps moving from left to right. While moving if any object which is needed occurs on the screen, YOLO will highlight that object. With the weights which are already present in the model, it will try to detect the object and recognize it. For each object there exists a different weight in YOLO. There are different types of YOLO. Some models may have 150 different objects, and some might have 80. There is an option to limit the number of objects to whatever is required, or one can use all the objects present in the model. In this research we have used a model with 80 weights. Every time the code runs, we have to load the weights. Since we do not want our model to detect all 80 weights or objects, we limit the weights for first 10. This is how detection of object takes place in YOLO.

B. Object Tracking

TensorFlow used can help to detect the object but it will not track the object. To track the object, bounding box is given to all the objects present in the video or on the screen. TensorFlow gives the kernel dimension of the weights. Out of eight coordinates in kernel dimension we extract four coordinates. We will multiply the width and height of the coordinates of the kernel dimension because to fix the dimension of the object with is detected

C. Distance Calculation

Distance is calculated by the bounding boxes. In fact, Yolo detect the objects of interest and give their regions. Next, it takes the left and right images and construct a dense disparity map. Finally, it takes different regions of the objects and work out an average disparity value to then work out a distance using the focal length and camera baseline of the stereo capture device.

In addition, the movement when other object is completely detected, the size becomes bigger. If the portion of the object detected is less, then we can assume that either the object is far from us or it is in the sideways like left or right. The bigger the size, the closer the vehicle or object is. And the smaller the vehicle, the distance is more. So, from the bounding box distance is calculated. We are taking some constant and we are predicting the distance.

Then, we display on the screen the estimated inter-vehicle distances each time a vehicle crosses the delimited area.

D. Architecture of our System

Before any experimentation could occur, a baseline system needed to be created. We began by stripping the given yolo.py to its main functions and modularizing it for use on individual frames. We then used parts of the given stereo_disparity.py to develop two functions for dense disparity distance calculation. First, yolo would detect the objects of interest and give their regions. Next, it would take the left and right images and construct a dense disparity map. Finally, it would take different regions of the objects and work out an average disparity value to then work out a distance using the focal length and camera baseline of the stereo capture device. Once we had this basic system working, we could begin to experiment with different techniques of optimization.

The first thing we noticed was that many of the images had a low contrast. To remedy this, we used a form of histogram equalization called Contrast Limited Adaptive Histogram Equalization (CLAHE) which works by taking small regions (tiles) and applying equalization on those, rather than the entire image. Whilst the filter did not seem to have much effect on distance values, it had some success in helping yolo detect objects in poor light conditions.

The next experiment was to apply a filter to the disparity map. The filter we tried was the Weighted Least Squares (WLS) filter. The WLS filter smooths the disparity map and makes it more uniform. This seems like it should help with the distance calculation though, in practice, not much change was seen and in some cases the filter made things worse. This could be because smoothing causes the image to lose detail and thus lose valuable information that could have helped with distancing. We also apply a noise filter to the disparity map to lower the amount of noise as this would help to provide a better distance average.

In summary, we have experimented with various techniques to attempt to increase the robustness of the system. Whilst not all of these have proved effective, they have all lead to a solution that is a suitable prototype for object distance detection. YOLO has been able to find most of the objects in the scene (helped a little by CLAHE histogram equalization) and the disparity maps seem to have given enough information to get a reasonable distance estimate.

Moreover, the results are assumed to be accurate when the model detects the objects correctly.

During the validation of objects detected by the model, we have got the object accuracy of 84.89% for 0.031 seconds per image of processing (Fig. 8).

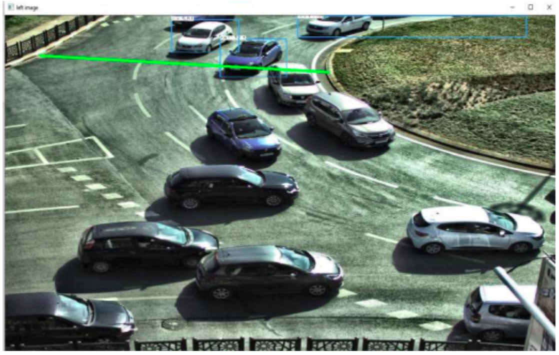


Fig. 8. Execution of the Algorithm.

VI. Conclusion

When we analyze the results, we have got for precision and recall, it can be said that YOLO is one of the best models used in detection of vehicles. YOLO model has achieved 85% of precision with 62% of recall with the time rate of 30 frames per second. We have also successfully found the distance between the vehicles.

The YOLO model is in the top place in the object detection speed when compared to other convolution neural networks. The detection speed that we have achieved is 0.03 seconds per image, which is 10 times faster than the already present object detection models. The YOLO model is the only model that has achieved this accuracy in real-time video streaming. From the computation of orientation estimation, we have found that YOLO has a good precision in prediction of object orientation. By all the experiments conducted, it is proved that performance of YOLO is high in both object detection and orientation precision. Since object's orientation has a main role in intelligent transport systems, with the accuracy we got for orientation estimation we can state that YOLO fits in the best for them. We have also successfully found the distance between the vehicles.

A. Implementation of our System in Real World

Layer 1: *Acquisition and pre-treatment*
The main function of this layer is to ensure the acquisition of images from a stereoscopic system. Then, there is image processing which consists in improving the quality of the image by removing noise, camera vibrations, lighting changes, etc.

Layer 2: *Attribute extraction and analysis*
From the images obtained from Layer 1, this layer extracts the static and dynamic attributes of the vehicles necessary for road traffic management: vehicle detection, trajectory extraction, vehicle recognition (license plate, mark, and color detection), and the measurement of inter-vehicle distance. Then, the extracted attributes are analyzed to understand traffic conditions and behaviors.

Layer 3: *Detection of illegal activities and anomalies and analysis of traffic flows*
Based on the results of the previous layers, this layer provides services for efficient management and control of road traffic. It can detect traffic violations (such as stop violations, red light violations, speeding, overtaking, fake license plate, unauthorized change of direction, etc.) and anomalies on the road (accidents, obstacles blocking traffic, traffic light malfunction, etc.).

B. Future Work

In this paper, we present a method of inter-vehicular distance estimation based on stereoscopic vision. Indeed, after detecting, locating, and delimiting vehicles using the YOLO V3 Deep Neural Network algorithm, we estimate the distance separating a vehicle from the cameras based on stereo vision principles to finally deduce the inter-vehicular distance. As a perspective, we plan to extend the technique to estimate the inter-vehicle distance in urban areas, we aim to satisfy the real-time constraint to be able to deploy our system in the real world.

References

[1] O. Bourja, K. Kabbaj, H. Derrouz, A. El Bouziady, R. O. H. Thami, Y. Zennayi, and F. Bourzeix, “Movits: Moroccan video intelligent transport system,” in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*. IEEE, 2018, pp. 502–507.

[2] G. Kim and J.-S. Cho, “Vision-based vehicle detection and inter-vehicle distance estimation for driver alarm system,” *Optical review*, vol. 19, no. 6, pp. 388–393, 2012.

[3] A. EL BOUZIADY, R. O. H. THAMI, O. BOURJA, F. BOURZEIX, and S. EL FKIHI, “Estimation du vitesse du trafic routier avec la stéréovision.”

[4] A. El Bouziady, R. O. H. Thami, M. Ghogho, O. Bourja, and S. El Fkihi, “Vehicle speed estimation using extracted surf features from stereo images,” in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2018, pp. 1–6.

[5] F. Bourzeix, O. Bourja, M. A. Boukhris, and N. Es-Sbai, “Speed estimation using stereoscopic effect,” in *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*. IEEE, 2014, pp. 147–151.

[6] P. Charbonnier, V. Muzet, P. Nicolle, N. Hautiere, J.-P. Tarel, and D. Aubert, “La stéréovision appliquée à l’analyse de scènes routières,” *Bulletin des laboratoires des ponts et chaussées*, no. n272, pp. p57–74, 2008.

[7] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[8] B. Strbac, M. Gostovic, Z. Lukac, and D. Samardzija, “Yolo multi-camera object detection and distance estimation,” in *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE, 2020, pp. 26–30.

[9] H. Derrouz, A. E. Hassouny, R. Oulad Haj Thami, and H. Tairi, “Hybrid method for background modeling and subtracting,” in *2017 Intelligent Systems and Computer Vision (ISCV)*, 2017, pp. 1–5.

[10] H. Derrouz, A. Elbouziady, H. Ait Abdelali, R. Oulad Haj Thami, S. El Fkihi, and F. Bourzeix, “Moroccan video intelligent transport system: Vehicle type classification based on three-dimensional and two-dimensional features,” *IEEE Access*, vol. 7, pp. 72 528–72 537, 2019.

[11] Y. Li, Z. Zhao, Y. Luo, and Z. Qiu, “Real-time pattern-recognition of gpr images with yolo v3 implemented by tensorflow,” *Sensors*, vol. 20, no. 22, p. 6476, 2020.

[12] J. Ciberlin, R. Grbic, N. Teslić, and M. Pilipović, “Object detection and object tracking in front of the vehicle using front view camera,” in *2019 Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE, 2019, pp. 27–32.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[14] D. Jifeng, L. Yi, H. Kaiming, and S. Jian, “Object detection via region-based fully convolutional networks,” *Advances in Neural Information Processing Systems*, pp. 379–387, 2016.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *arXiv preprint arXiv:1506.01497*, 2015.

[16] S. K. Chadalawada, “Real time detection and recognition of construction vehicles: Using deep learning methods,” 2020.

[17] J. B. Guapacha and S. C. A. Mantovanni, “Real time object detection and tracking using the kalman filter embedded in single board in a robot,” in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Ieee, 2017, pp. 1–6.

[18] Z. Chen, R. Khemmar, B. Decoux, A. Atahouet, and J.-Y. Ertaud, “Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility,” in *2019 Eighth International Conference on Emerging Security Technologies (EST)*. IEEE, 2019, pp. 1–6.