REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche scientifique
Ecole Nationale Polytechnique

Electronic department

End of studies dissertation

For obtaining the electronic engineering diploma

Artificial Intelligence-Based System for Accident Prevention on Railways

Presented publicly on 08/07/2020 by:
Hanane Hamar & Sarah Si Youcef

Jury members:

President	Mourad Haddadi	Professor	ENP
Examiner	Mourad Adnane	Professor	ENP
Supervisor	Sid-Ahmed Berrani	Doctor	ENP
Co-promoter	Sid-Ali Zerrouki	Director	Rail TELECOM

ENP 2020

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche scientifique
Ecole Nationale Polytechnique

Electronic department

End of studies dissertation

For obtaining the electronic engineering diploma

Artificial Intelligence-Based System for Accident Prevention on Railways

Presented publicly on 08/07/2020 by:
Hanane Hamar & Sarah Si Youcef

Jury members:

President	Mourad Haddadi	Professor	ENP
Examiner	Mourad Adnane	Professor	ENP
Supervisor	Sid-Ahmed Berrani	Doctor	ENP
Co-promoter	Sid-Ali Zerrouki	Director	Rail TELECOM

ENP 2020

ملخص

يعاني قطاع النقل بالسكك الحديدية من مشكلة كبيرة، الا وهي الحوادث التي تؤدي الى خسائر مادية و بشرية هائلة.

للصد لهذه المشكلة او التقليل من حالات الحوادث ، اقترحت سلامة السكك الحديدية العديد من الأعمال باستخدام طرق وتقنيات مختلفة.في هذا العمل سنقترح حل باستعمال الذكاء الاصطناعي.
لتجنب كل هذه الخسائر ، هناك العديد من التحديات مقارنة بقطاع السيارات الذكية.

نظرا للشروط المختلفة للإضاءة وظروف الطقس وضرورية الكشف عن العوائق المتواجدة بجوار السكة الحديدية على بعد مسافة تفوت ال 500 م، يُقترح نظام يجمع بين نوعين من الكاميرات.

ينقسم هذا العمل إلى قسمين .الاول، كشف العوائق المتواجدة على السكك الحديدية، و للقيام بذلك ، تم تقييم ومقارنة ثلاث خوارزميات مشتركة للكشف عن الأشياء(مقارنة الفعالية والكفاءة) و اختيرت الخوارزمية التى تعطي نتائج على اصغر نسبة للانذرة الخاطئة واكبر نسبة للكشف عن العوائق و كذلك أقصر وقت للتنفيذ ، لإنشاء نظام موثوق به يمكنه العمل و تحليل المشاهد على المباشر.

الخطوة التالية هي تقدير المسافة بين العوائق و الكاميرا المركبة على القطار باستخدام خوارزمية DisNet مع طريقتين لتقييم النتائج.

الكلمات المفتاحية:
القطار ، السكك الحديدية ، الرؤية عبر الحاسوب، التعلم الآلي ، التنبؤ بالحوادث ، الكشف عن الاجسام ، التقييم ، تقدير المسافة.

Résumé

Le transport ferroviaire est confronté à un grand problème; les accidents sur le chemin de fer qui engendrent des pertes humaines et matérielles.

Pour résoudre ce problème ou au moins réduire les cas d'accidents, la sécurité ferroviaire a proposé de nombreux travaux en employant différentes méthodes et technologies. Dans ce travail, une solution utilisant l'intelligence artificielle est proposée.

Le système développé doit respecter de nombreuses conditions contrairement aux systèmes dédiés pour le secteur des véhicules autonomes. L'un des principaux défis est la détection d'obstacles à longue portée dont elle est limitée à 200 mètres dans les systèmes des véhicules autonomes.

Il y'a plusieurs exigences à respecter et de nombreuses contraintes à prendre en concidération, y compris les changements dans les conditions d'éclairages et météorologiques, ainsi que la portée de détection d'objets qui dépasse les 500 m, c'est pour cela un système combinant deux types de caméras (une camera RGB et thermique) est suggéré.

Ce travail combine deux parties. La première partie concerne la détection d'obstacles. Pour cela, trois algorithmes de détection d'objets ont été considérés, évalués et comparés (comparaison d'efficacité et d'efficience). Le choix s'est porté sur celui qui donne le taux de fausses alarmes le plus faible, le taux de détection le plus élevé, et dont le temps d'exécution est le plus court, pour créer un système fiable qui peut fonctionner en temps réel.
Après la détection des obstacles présents sur le chemin de fer, la partie suivante consiste à estimer leurs distances au train avec un algorithme nommé "DisNet", les résultats obtenus sont évalués avec deux méthodes (l'erreur moyenne quadratique et le taux d'erreur).

Mots clés:
Train, Chemin de fer, Vision par ordinateur, Apprentissage automatique, Prévision des accidents, Détection d'objets, Evaluation, Estimation de distance.

Abstract

Railway transport suffers from a major problem which is accidents, where the losses in human lives and the materials are enormous.

To solve this problem or at least reduce the number of accidents, railway safety has proposed numerous initiatives with different methods and technologies. In this work, a method using artificial intelligence is proposed.

The system developed must comply with numerous conditions contrary to the systems designed for the autonomous vehicle sector. One of the key challenges is long-range obstacle detection. Sensor technology in current land transport research is able to look 200 m ahead.

A system that combines two types of cameras; RGB and thermal; is suggested to consider the different requirements. these ones consist of the different illumination and weather conditions, and also the range needed to detect obstacle on time.

This work is divided into two parts. The first one is the obstacle detection. To do so, three common object detection algorithms have been evaluated and compared (effectiveness and efficiency comparison). The algorithm with the lowest FAR(false Alarm rate), the best DR (Detection Rate), and that achieves the shortest execution time has been chosen. The objective is to build a reliable system that can operate in real-time.
The second part focuses on the estimation of the distance between objects and the train using the DisNet algorithm. Here, two evaluation methods have been used in order to assess the precision of the estimation.

Key Words:
Train, Railway, Computer Vision, Machine learning, Accident prevention, Object Detection, Evaluation, Distance Estimation.

Acknowledgements

We are grateful to God for the good health and wellbeing that were necessary to complete this research work.

*We present our sincere thanks to **Dr.Sid-Ahmed BERRANI**, for providing guidance and feedback and overall insights in this field have made this an inspiring experience for us.*

*We wish to express our thanks to the director of Rail Telecom **Sid-Ali ZERROUKI** who trusted us and proposed this interesting problematic.*

A special thanks to our family members for their encouragement, moral support, personal attention and care.

Dedicaces

We dedicate this modest work

To our dear parents

To our families and friends

To all those who encouraged and Supported us...

Hanane & Sarah

Contents

List of Tables

List of Figures

Acronyms And Abbreviations

ANN Artificial Neural Network.

AoV Angle of view.

Bbox Bounding Box.

CNN Convolutional Neural Network.

COCO Common Objects in Context.

CPU Central Processing Unit.

DR Detection Rate.

FAR False Alarm Rate.

FC Fully Connected.

FN False Negative.

Fov Field of view.

FP False Positive.

fps frame per second.

GPU Graphics Processing Unit.

GT Ground truth.

IoU Intersection over Union.

IP Internet Protocol.

LiDAR Light Detection and Ranging.

mAP Mean Average Precision.

OpenCV Open source Computer Vision.

PoE Power over Ethernet.

RGB Red, Green, Blue.

RMSE Root Mean Square Error.

ROI Region Of Interes.

RPN Regional Proposal Network.

SLAM simultaneous positioning and mapping.

SSD Single Shot Detector.

TN True Negative.

TP True Positive.

VOC Visual Object Class.

Yolo You Only Look Once.

Yolo V3 You Only Look Once Version 3.

Introduction

Problem description

The safety of rail traffic is the most important activity and occupancy of railways companies. According to statistics, the accident on railways depends on several factors: The technical condition of the railway infrastructure, the rolling stock, the organization of traffic and rail transport, and also the unpredictable obstacles that can be found on railroads, such as animals, humans, and vehicles on the railroad crossing.

The statistics of accidents on railways caused by those obstacles are not negligible. It is a major problem that generates human, material and environmental losses. In Algeria, 64 railway traffic accidents were recorded at level crossings in 2017, resulting in 8 deaths and 46 injuries.

To propose a solution for this huge problem that generates risks for humans and even for the economy of the company, there are many constraints and conditions to be respected. It is not as easy as the solutions used in autonomous vehicles because of the distance required to detect obstacles when the train is travelling.

The testification of the railway company; a train takes up to one kilometer to stop, even in emergency braking situations. Trains cannot stop quickly or deviate from their path to avoid hitting a person or an object. For that, the obstacles have to be detected at long range(up to 1000m). Thus commonly used range sensors like LiDAR and stereo vision, have their issues of sparse data and short-range detections. Hence, they cannot be used in this context.

The trains travel during the day and night, they go through the tunnels, and in the hardest weather conditions. Thus, it is difficult for the train conductor to see clearly. So, the solution should not have difficulty to operate in these conditions. This is an eventual requirement.

In the case where the conductor cannot warn the pedestrian or another train, the solution should operate in real-time to make a decision as quickly as possible, and save lives.

With the contribution of Railway Telecom company, a solution has been proposed to this problem in this project. Two types of cameras have been considered that are RGB camera and thermal camera, associated with artificial intelligence based algorithms for obstacle detection on railways, and the estimation of their distance to the train.

Company Presentation

Rail telecom is a joint venture between SNTF "Société nationale des transports ferroviaires" 51% and KAPSCH Carrier Com AG 49%. This company was founded in 2015. It is specialized on GSM-R, SDH, Fiber Optic, FLM, Managed Services, Engineering, Deployment, Design & Optimization, Repair, Telecom construction, management of railway telecommunication maintenance as well as all other design, development and related activities.

The company has an interest in railway safety. One of the biggest challenges is the detection of obstacles for safe autonomous driving.

Contribution

The proposed solution is a combination of obstacle detection algorithm based on convolutional neural network (CNN), and distance estimation algorithm called "Disnet" network.

Our contribution is summarized as follow:

1) An evaluation of the different algorithms used for obstacle detection as well as the evaluation of the distance estimation algorithm, in order to verify the results according to the requirements.

2) The collection and the manual labeling of data-set of images from the Internet. This data-set contains the different possible obstacles that can be found on the railroad.

3) Development of a user interface application to warn the conductor in case of obstacle detection. The interface shows both the detected obstacles and their estimated distances to the train.

Report Structure

This report contains three important chapters.

State of art, where we summarize the different methods used to establish railway safety. Also, we introduce different foundations to be used in this work.

In our solution, we have tried to respect all the requirements already detailed, in the choice of the equipment to use, and also the algorithms. For the material, two types of cameras are employed, color and thermal cameras to create the system.
The method combines two major parts: Obstacle detection and object to camera distance estimation based on machine learning. We need to make this process not just reliable but fast and not so computationally expensive either. Because of that, evaluations are done for both algorithms, to finally utilize the one which is better to the considered problem. All these details are presented in the second chapter.

In the last chapter, our experiments have been presented as well as the user interface which has been built to alert the train conductor in the presence of obstacles.

Chapter 1

State of the art

This chapter introduces related works relevant to obstacle detection and distance estimation systems. These methods are mostly used in autonomous driving applications. We focus on techniques dedicated to railways in order to prevent the accidents caused by the different objects present in the railroad.

1.1 Object detection in images

Object detection is a computer technology that processes digital images and videos, to detect different shapes and objects on it. In this section, some classical approaches and others based on machine learning are presented.

1.1.1 Classical methods

The classic detection method can be turned up to the 1960s. The principle is to define the structural relationships between objects and to use manual engineering to model the interaction between object placements and their context. Some widely used classical methods include Haar-like features [1], SIFT [2], SURF [3], and HOG detectors [4].

Object Tracking and Analysis for Detecting Various Situations at Level Crossings of Railway-Road [5], a method to identify and isolate moving items into a given zone is proposed in this work. Firstly, identifying moving pixels by contrasting particular vitality vectors related to each moving pixel. The next stage is to track them using Harris detector [6]. Finally, providing the directions of the moving articles.

Advanced thermal camera based system for object detection on rail tracks [7]. this work demonstrates the advantages of the thermal camera, which can operate in different illumination and weather conditions (day, night, rain, fog...).
The system is divided into two parts. The detection of the rails then the detection of different

obstacles present in rail tracks. For the rails detection, edges detector is used; edges detection is a process to detect significant local changes in pixel intensity in images, this change is measured by the gradient. For that canny edge detector has been chosen by [7] after the test of the different edge detectors such as: Roberts [4], Sobel [4], Canny [4], Log detection [4].

For the detection of objects, a segmentation method has been used, which is based on breaking the image into a set of disjoint regions that seem similar according to a set of predefined criteria. There are multiple methods like thresholding, region growing, and region splitting and merging. For this case, thresholding have been used. As shown in Figure 1.1.

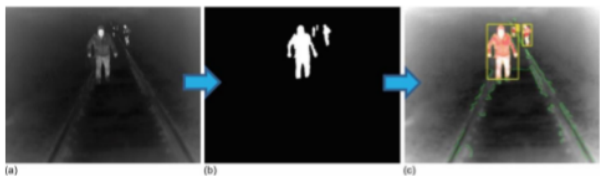


Figure 1.1: Input image (a), detected objects (b), and located objects (c) [7].

Detecting Rails and Obstacles Using a Train-Mounted Thermal Camera [8], this system includes a thermal camera connected to a computer and a screen with a graphical interface. The computer is used to run thermal image acquisition. After that, the detection of railways and obstacles is processed.
For railway detection, the curvature of the track has been estimated by mathematical relationship according to its geometry.

If the rails are not detected correctly, the obstacles would also be erroneously detected. To eliminate this problem, an adaptive filter has been used for correction. The original image of the rails and a binary mask are the inputs of this filter. To detect obstacles thresholding has been done using the previous filter data.
Figure 1.2 represents rail detection before and after applying the filter.

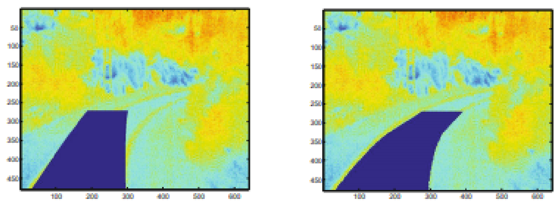


Figure 1.2: Example of a rail mask correction, left: detection before the mask, right: after applying the mask [8].

The moving camera background subtraction method [9], this method detects obstacles by comparing the input and the reference image. The input image is taken while driving the

train, and the reference was taken previously and saved on the same trajectory (generally the trains take the same road).
Before the subtraction of these two image sequences, pixel-level alignment is needed; both spatial and temporal alignments. To solve this, the proposed method first finds a reference frame captured at the most similar location to the current frame by image sequence matching. After that, a subtraction is done to detect the obstacle.
Figure 1.3 presents the actual and the reference frames.

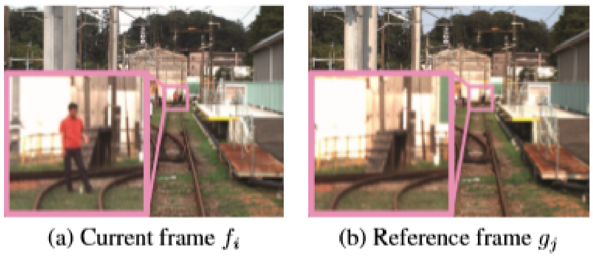


Figure 1.3: Corresponding frames (f_i,g_j) in the current and the reference image sequences [9].

1.1.2 Machine learning methods

The concept of artificial neural networks is to reproduce neurons that work like human brains and model them to treat a given problem. Those models can learn from training data and take decisions.

Artificial neural networks generalities

The fundamental unit of a neural network is a neuron. Each neuron has a vector input x_i, multiplies it by the weight w_i, and adds a bias b to it. After that, an activation function f is applied to give the output of the neuron. This activation function determines whether to trigger neurons or activate it. The activation function could be a Sigmoid, Tanh, etc. Figure 1.4 shows the architecture of a neuron [10].

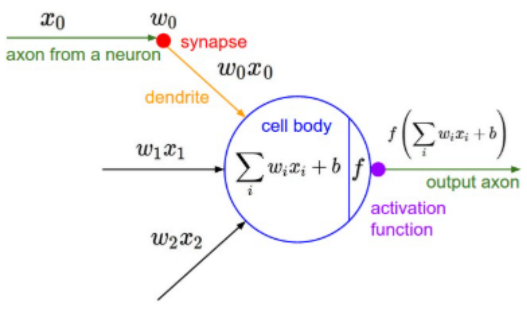


Figure 1.4: Architecture of a single neuron, where x is input, w is weight, b is bias, f is activation function [10].

Machine learning algorithms learn from data. This data should contain ground truth with labels. For training, machine learning algorithms use examples (data-sets) to teach models. The data-set is usually divided into training, validation, and test sets. The training set is a set of examples used to fit the parameters (e.g. weights of connections between neurons in Artificial Neural Network) of the model. The fitted model is used to predict the responses for the observations in the validation set. Test data is used to generate the final result of the proposed algorithm and measure accuracy. After training the model, it will be able to treat new unknown data.

Deep neural networks are multi-layer structures, where each layer has a fixed number of neurons. The number of neurons in the first layer is equal to the size of the input vector. The number of neurons in the output layer is equal to the number of outputs (e.g classes in object detection algorithms).

Convolutional Neural Networks CNN

CNN [11]: is a deep leaning algorithm used in image recognition, image classification, etc. Some algorithms that use CNN to train their models are objects detection, faces recognition. Deep learning CNN is divided into a series of convolutional neurons with filters (kernels), Pooling, fully connected layers (FC), as shown in Figure 1.5:

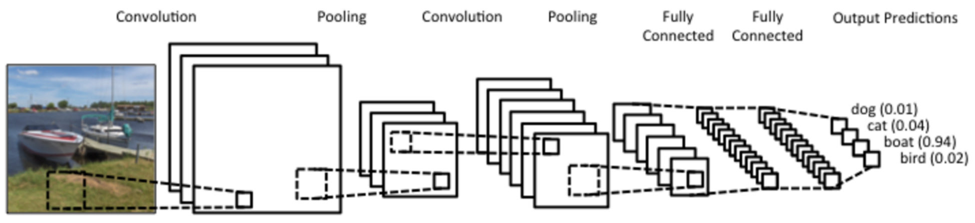


Figure 1.5: Complete CNN architecture.

Convolutional neurons:
CNN takes as input a digital image (matrix) of dimension (h × w × d) and a filter of dimension (f_h × f_w × d). This filter is called feature map. Then the CNN returns an image of dimension ((h - f_h+ 1) × (w - f_w + 1) × 1).
Some applications of these filters: edge detection, blur, sharpen...

Pooling Layer:
This layer is used to reduce the spatial size of the convolved features. There are three forms of pooling:

- Max pooling takes the largest element from the rectified feature map.
- Average pooling takes the average of rectified feature map's elements.
- Sum pooling replace the feature map with the sum of the elements.

Fully Connected Layer:
FC (Fully Connected): is a feed forward neural networks. The input to the fully connected layer is the output from the final pooling or convolutional layer, which is flattened and then fed into the fully connected layer.

Data-set used in object detection

The creation of the data-set is the first thing that should be done for training the network. There are many tools for labelling the data-set. The resulting data is stored in files to be used while training. In this section, two data-sets are introduced, MS COCO [12] and Pascal VOC [13]. They are used in object detection, segmentation, captioning algorithms...

Microsoft Common Objects in COntext (MS COCO) [12]:
This data-set contains 91 common object categories with 82 of them having more than 5,000 labeled instances. In total the data-set has 2,500,000 labeled instances in 328,000 images [12].

PASCAL Visual Object Class (VOC 2007) [13]:
This data-set consists of about l,500,000 images were retrieved from flickr. For each of the 20 object classes.

Object detection algorithm

This algorithms take an image as input and returns bounding boxes in the image (x, y, w, h) which delimits the detected objects, and class label and its confidence.
Three popular object detection algorithms are presented in the paragraphs below.

Faster RCNN

The faster R-CNN has two networks: A regional proposal network (RPN) used to generate regional proposals and a network that uses those proposals to detect objects.
Regional Proposal Network (RPN): It takes images of different shapes as input, and returns a set of proposals rectangles where objects are likely to exist.
Region Of Interest (ROI) pooling: The outputs of RPN are a suggested regions with different sizes. Regions of different sizes need CNN feature maps of different sizes for classification, which makes it hard to build an efficient structure. ROI Pooling can simplify this problem by reducing the feature map to the same size.
The predicted anchors are used as input for the second network which is trained to predict the real class of the object. This network calculates the overlaps between ground truth and the predicted anchors, and compares it with a threshold value. If the overlap is higher than the defined threshold this anchor is classified as an object and the network gives its class and the confidence, else the anchor is classified as background.
Figure 1.6 presents the architecture of Faster R-CNN.

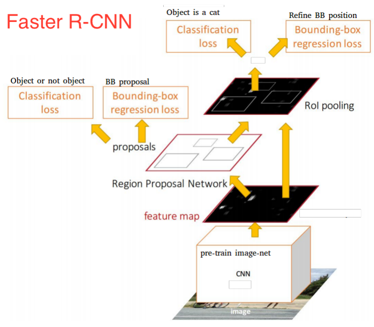


Figure 1.6: The architecture of Faster R-CNN algorithm [14].

Loss Function:

Loss function in Faster R-CNN for an image is the combination of the classification loss and the regression loss, it is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*)[14], \tag{1.1}$$

where:

i is the index of an anchor.
p_i is the predicted probability of anchor i being an object.
p_i^* the ground-truth label is 1 if the anchor is positive, and 0 otherwise.
t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box.
t_i^* is a vector representing the 4 parameterized coordinate of the ground-truth box associated with a positive anchor.
The classification loss L_{cls} is log loss over two classes (object vs. not object).
For the regression loss, we use $L_{reg}(t_i,t_i^*) = R(t_i - t_i^*)$ where R is the robust loss function. The term $p_i^* L_{reg}$ means the regression loss is activated only for positive anchors $p_i = 1$ and is disabled otherwise $p_i = 1$.
The outputs of the cls and reg layers consist of p_i and t_i respectively.
The two terms are normalized by N_{cls} and N_{reg} and weighted by a balancing parameter λ.

Single Shot Detector (SSD)

The SSD method is based on a feed-forward convolution network, which generates a collection of fixed-size bounding boxes. Then perform non-maximum suppression steps to generate final inspection.
Multi-scale feature maps for detection: These layers decrease in size progressively and allow predictions of detections at multiple scales. The SSD is one of the first methods using a convolutional network's pyramidal feature hierarchy for different size object detection. Grid of 1 × 1 is dedicated for detecting objects taking up the whole image, and 38 × 38 is used to detect the smallest object.
Figure 1.7 presents the architecture of Single Shot Detector SSD.

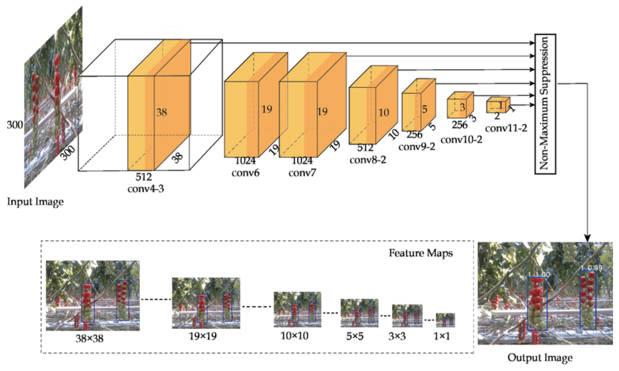
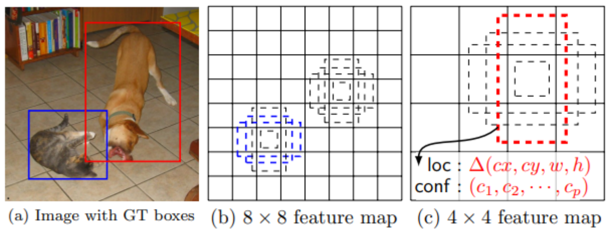


Figure 1.7: Architecture of Single Shot Detector (SSD).

Training:
During training, all default boxes according to boxes in the ground truth are selected, that contains different scales, locations, and ratio. Default boxes that have a higher overlap with boxes of GT are selected.
Choosing scales and aspect ratios for default boxes: to handle the difference between object scales a different feature maps of different layer sizes are used. Lower layers capture more fine details of the input objects. Figure 1.8 represents the result of two sizes of features 4 × 4 and 8 × 8.



loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$
(a) Image with GT boxes
(b) 8×8 feature map
(c) 4×4 feature map

Figure 1.8: Two exemplar feature maps (8 × 8 and 4 × 4) [15].

In Figure 1.8, the dog matches the default box in the 4 × 4 feature map but does not match any default box in the 8 × 8 feature map. This is because these boxes have different proportions and do not match the dog's box.

Hard Negative Mining: to achieve stable training and faster optimization, the negative examples (false positive) are sorted in the ascending order and most of the wrong examples are picked up in the loss function.
Data Augmentation: a zoom in and zoom out actions are used to increase the performance of detecting large and small objects, respectively.

Loss function [15]:
$x^p_{i,j} = \{1, 0\}$ is the indicator for matching the i-th default box to the j-th ground truth box of category p.

The loss function is a weighted sum of the localization loss (loc) and the confidence loss (conf):

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))[15], \tag{1.2}$$

where:

N is the number of matched default boxes.
The localization loss is a smooth L1 loss between the predicted box (l) and the ground truth box (g) parameters.
(cx, cy): the center of the default bounding box (d) and its width (w) and height (h).

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{i,j}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m)[15] \tag{1.3}$$

$$\hat{g}_j^{cx} = \frac{g_j^{cx} - d_j^{cx}}{d_i^w} \qquad \hat{g}_j^{cy} = \frac{g_j^{cy} - d_j^{cy}}{d_i^h}$$

$$\hat{g}_j^{w} = log(\frac{g_j^w}{d_i^w}) \qquad \hat{g}_j^{h} = log(\frac{g_j^h}{d_i^h})$$

The confidence loss is the softmax loss over multiple classes confidences (c).

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{i,j}^{p} log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^O) where : \hat{c}_i^p = \frac{exp(c_i^p)}{\sum_p exp(c_i^p)}[15] \tag{1.4}$$

You Only Look Once (Yolo)

Yolo applies a single neural network to the complete image. The network divides the image into multiple grids of s × s sells and predicts the bounding box and probability of each grid. These bounding boxes are weighted by the predicted probability.
The structure of Yolo model is divided into four main steps: [16]
Data Pre-processing: Before defining the model the images are resized and significant information from them are extracted; the coordinates of the bounding boxes and the confidence of each class.

$$\hat{y} = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, ..., c_m]^T[16], \tag{1.5}$$

where p_c is the confidence of a detected object. b_x and b_y is the coordinates of the centre of bounding box calculated as:

$$b_x = \frac{x_{min} + x_{max}}{2W}, b_y = \frac{y_{min} + y_{max}}{2H} \tag{1.6}$$

b_w: width of bounding box defined as:

$$b_w = \frac{x_{min} - x_{max}}{2W} \tag{1.7}$$

b_h: height of bounding box.

$$b_h = \frac{y_{min} - y_{max}}{2H} \tag{1.8}$$

W, H: the dimensions of the image.

Intersection Over Union: This metric is used to evaluate the accuracy of the algorithm. It represents the overlap between the boxes detected. This score shows how much the predicted bounding box matches the ground-truth bounding box in the image.

Model Definition: To create the model a transfer learning is applied. (The transfer learning is the knowledge acquired from the training data-set, called the source data-set, is transferred in order to be able to properly process the new data-set, called target).

Multiple Detection Handling:
After getting all bounding boxes two filters are used to correct the detection:

1) Threshold Filtering: Is used to eliminate bounding boxes that do not contain an object. This filter eliminates those bounding boxes according to them confidence. By defining a threshold value to compare the confidence, if p_c is less than the threshold chosen the corresponding box is eliminated from the list of boxes.

2) Non-max Suppression: This filter is used to eliminate the double detection.
Anchor boxes: This filter is used to limit the number of detection. If the number of anchor boxes is equal to N, the system can detect at maximum N objects in the input image.
Figure 1.9 presents the architecture of yolo.

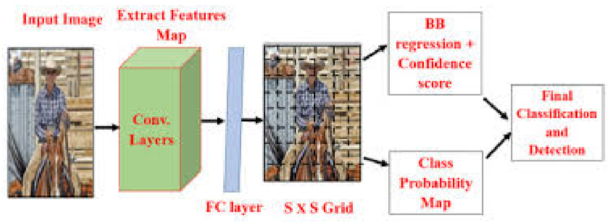


Figure 1.9: Yolo architecture.

Loss function [17]:

The loss function composes of the classification loss, the localization loss, and the confidence loss.

1) The classification loss:
If an object is detected, the classification loss at each cell is the squared error of the class conditional probabilities for each class:

$$\sum_{i=0}^{s^2} l_i^{obj} \sum_{c \in class} (p_i(c) - \hat{p}_i(c))^2, \tag{1.9}$$

where:

$l_i^{obj} = 1$ if an object appears in a cell i, else it is equal to zero.
$\hat{p}_i(c)$ class probability for class c in cell i.

2) The localization loss:

It measures the errors in the predicted boundary box locations and sizes.

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{i,j}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{i,j}^{obj} [(\sqrt{w}_i - \sqrt{\hat{w}}_i)^2 + (\sqrt{h}_i - \sqrt{\hat{h}}_i)^2], \quad (1.10)$$

where:

$l_{i,j}^{obj} = 1$ if the j th boundary box in cell i is responsible of detecting an object otherwise 0.
λ_{coord} increase the weight for the loss in boundary box coordinates.

The confidence loss:
If an object is detected in the box, the confidence loss measuring the objectness of the box is:

$$\sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{i,j}^{obj} (c_i - \hat{c}_i)^2, \quad (1.11)$$

where:

$\hat{c}_i$ is the confidence score of the box j in a cell i.
$l_{i,j}^{obj} = 1$ if the j th boundary box in cell i is responsible of detecting an object otherwise 0.

If an object is not detected in the box, the confidence loss is:

$$\lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{i,j}^{noobj} (c_i - \hat{c}_i)^2, \quad (1.12)$$

where:

$l_{i,j}^{noobj}$ is the complement of $l_{i,j}^{obj}$.
$\hat{c}_i$ is the box confidence of the box j in cell i.
λ_{noobj} weights down the loss when detecting background.

1.1.3 Evaluation metrics

The objective of an object detection method is to identify if an object is present in the image, and to predict the coordinates of the bounding box around the object and its class. Here we compare the coordinates of ground-truth and predicted bounding boxes. So an evaluation of both classification as well as localization of using bounding boxes in the image is needed. For this purpose the mAP [18] is introduced in the paragraph below.
mAP (mean Average precision) is a popular metric in measuring the accuracy of object detectors like Faster R-CNN, SSD, and Yolo V3. Before defining the most common mAPs, first, we need to clarify some important concepts:

Confidence score is the probability that the bounding box contains objects. Usually predicted by the classifier [19].

Intersection over Union (IoU) is defined as the area of the intersection between predicted bounding box and ground truth bounding box divided by the area of their union (Figure 1.10) [19].

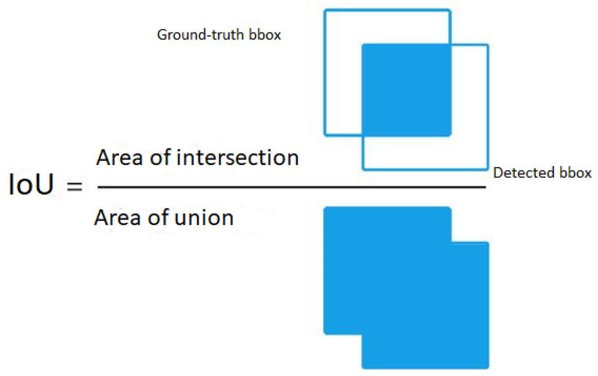


Figure 1.10: Definition of IoU.

Both confidence score and IoU are used as the criteria that determine whether the detection is a true positive (TP) or a false positive (FP).

A detection is considered as TP if its confidence score is higher than the threshold of confidence score, and the value of IoU is higher than the threshold of IoU, and the predicted class matches the class of ground truth. If the two last conditions are not met, the detection is considered as FP.
Note that for some detection metrics like PASCAL VOC, if multiple predictions correspond to the same ground truth, only the prediction with the highest confidence score is counted as TP, the rest are considered as FP.
In case where the confidence score of the detection is lower than the threshold of confidence score, if the object exists in ground truth the detection is considered as FN, else the detection is considered as TN.

So object bbox predictions into a certain class can be divided into four categories as shown in Table 1.1 for a class: Person.

	Prediction: Person	Prediction:No Person
Actual: Person	True Positive(TP)	False Negative(FN)
Actual: No Person	False Positive(FP)	True Negative(TN)

Table 1.1: Distribution of all the predictions with respect to ground-truth (Actual).

Other important concepts are Precision and Recall. Both of them are calculated using true positives(TP), false positives(FP) and false negatives(FN) as shown in Figure 1.11.

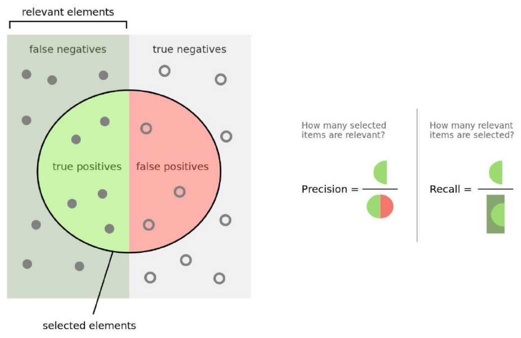


Figure 1.11: Recall vs Precision [20].

Precision is defined as the number of true positives divided by the sum of true positives and false positives (equation (1.14)) [18].

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Predictions} \tag{1.13}$$

Recall is defined as the number of true positives divided by the sum of true positives and false negatives (equation (1.15)) [18].

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{ground - truth} \tag{1.14}$$

Pascal VOC mAP

To calculate the Pascal VOC mAP [19], the AP per class is first calculated by following these steps:

- Sorting the detected bboxes for each class based on their confidence score.

- Notating each detection with TP or FP and calculating the accumulated TP and FP values.
- Calculating the Recall and Precision values where recall is defined as the proportion of all positive examples ranked above a given rank and precision is the proportion of all examples above that rank which are from the positive class.
- Plotting the recall-precision curve with varying the value of confidence.

The AP summarizes the shape of the precision-recall curve. This curve shows the trade-off between precision and recall for different confidence thresholds. In VOC 2007, the AP is defined as the mean of precision values at a set of 11 equally spaced recall levels [0,0.1,...,1] (0 to 1 at a step size of 0.1). However from VOC 2010 the AP is defined by the area under the Precision-Recall curve, and the mAP is the mean value of the AP values for each class.

MS COCO mAP

The same method as PASCAL VOC is used for COCO evaluation [21], the difference is that the mean of precision values are at a set of 101 equally spaced recall levels [0,0.01,...,1] (0 to 1 at a step size of 0.01). Also, the IoU threshold ranges from 0.5 to 0.95 with a step size of 0.05 unlike Pascal VOC which use only one value. After that, the mean of those values is the COCO mAP, so the final mAP COCO1 is calculated as follow:

$$\frac{mAP_{0.5} + mAP_{0.5} + ... + mAP_{0.95}}{10} \tag{1.15}$$

where x in mAP_x refers to the IoU threshold value.

1.2 Distance estimation of objects in images

In many computer vision applications, accurate depth values are critical [22]. In recent years, due to industrial needs, this task has attracted attention. Other computer Vision tasks, such as 3D object detection and tracking [23], 2D or 3D semantic segmentation [24], and SLAM [25] can use these accurate depth hints to improve the accuracy of the mentioned tasks [22]. Next, a brief introduction to the current commonly used sensors or methods for distance estimation.

1.2.1 Lidar

LiDAR (light detection and ranging) is a sensor based on the principle of time of flight [26]. Figure 1.12 illustrates the working principle of LiDAR, the process is as follows [27]:

- The sensor sends out a signal pulse (laser) and records the precise time.
- The reflection of the pulse is detected and precise time is recorded.
- Using a constant speed of light, the delay is converted to a distance (Tilt range) between the source (sensor) and the detected object.

- After knowing the position and direction of the sensor, the reflective surface is calculated.

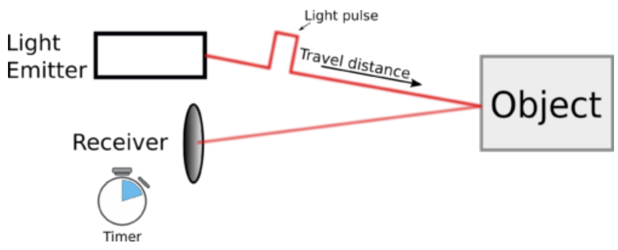


Figure 1.12: Working principle of LiDAR.

The output of LiDAR is a sequence of points consisting of three values: The distance from the surface, the scanning angle, and the reflectivity [28]. The reflectivity of the surface depends on its structure, materials, and colors. The most commonly used notation for LiDAR data is point cloud due to the wide variety of open source libraries [29]. Figure 1.13 shows an example of the point cloud visualization.

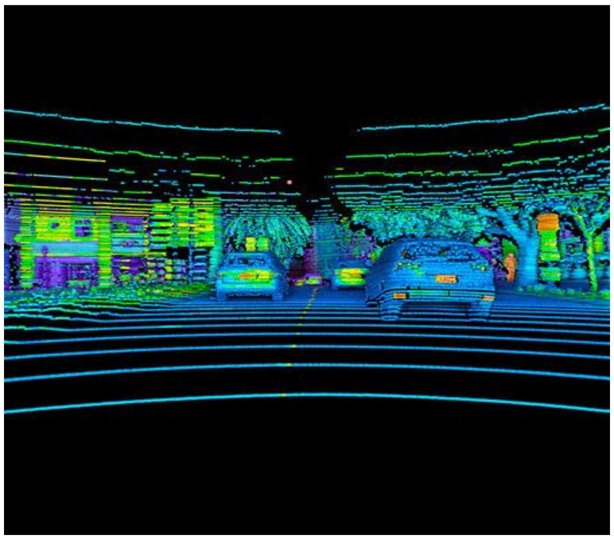


Figure 1.13: Lidar data (point-cloud) collected from vehicle [30].

3D spinning lidar is now widely used in autonomous driving. It usually has 16, 32, or 64 beams with vertical field-of-view (FoV) of up to 30 degrees and horizontal FoV of 360 degrees [31]. The LiDAR generates point clouds around it, but the range is limited. A large number of scan lines leads to high sparsity.

1.2.2 Stereo camera

A stereo camera is a type of camera with two or more image sensors. It is generally composed of two monocular cameras which are both imaging the observed scene. Similarly to how the human eyes are used for perceiving depth [32]. In contrast to a monocular camera, the second perspective allows the 3D location of real-world points to be estimated based on the position of their projections on the two image planes. This process is also known as triangulation and is covered by the epipolar geometry.

An example of a stereo camera with parallel optical axes can be seen in Figure 1.14. For a stereo camera, each of the two cameras has two coordinate systems: An image coordinate system (u,v) and a camera coordinate system (x, y, z).
Another important parameter of the stereo camera is the baseline b, which is the distance between the two cameras.

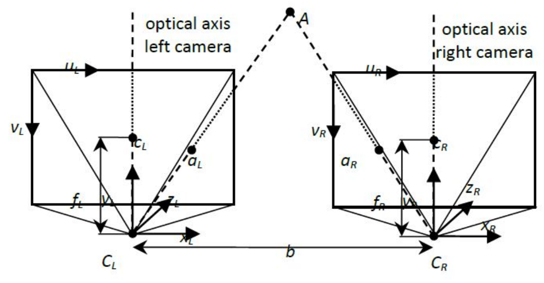


Figure 1.14: Stereo camera system with parallel optical axis with the image and camera coordinate systems of the left and right camera.

It is common to consider the origin of the stereo camera system in the optical center of the left camera. Stereo cameras with parallel optical axes have particular values for the rotation and translation matrices defining their pose with respect to a world coordinate system. The "ideal" extrinsic parameters of the camera:

$$R_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, t_L = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.16}$$

$$R_R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, t_R = \begin{bmatrix} -B \\ 0 \\ 0 \end{bmatrix} \tag{1.17}$$

where R_L and t_L are the rotation matrix and translation vector for the left camera. R_R and t_R are the rotation matrix and translation vector for the right camera. The baseline B is exactly the distance for which the right camera is translated along the x-axis, as can be seen in (1.18).

Stereo rectification

To calculate the disparity map the two stereo cameras with parallel optical axes must have equal focal lengths so that the two image planes lie in the same geometric plane. Due to mechanical misalignment of cameras, this condition is not satisfied so the images must be stereo rectified [33].

To perform stereo rectification, the intrinsic camera parameters are required together with the rotation matrix and translation vector. They describe the rotation and translation of the right camera with respect to the left camera. These parameters can all be determined using camera calibration.

The main cause of using stereo rectified images, is that in those images the correspondent points lie on the same image line. The searching process is thus faster. Using the adequate methods, the disparity map can be then computed rapidly.

Disparity map computation

As described in the previous paragraph the disparity map can be computed using stereo rectified images. Disparity maps are images in which each pixel value encodes the distance between the camera and the 3D point projected on the left image at the same coordinates. In other words, for each image point a(u, v) in the left image, the corresponding disparity value in the disparity map is d(u, v). The representations of the visualizing disparity maps:

- Gray-scale image: For which the pixel intensity is inverse proportional to the distance.
- Colour image: In which the distance is encoded with colours between red and blue where red points are close to the camera (hot) and blue points are far (cold).

One of the biggest problems faced in disparity map computation is finding corresponding points in the two images, a process also known as stereo matching.

There are various ways for computing disparity maps, which can be mainly divided into three categories:

1) Local methods, which only use the close neighbourhood of a pixel in the left image for finding its correspondent pixel in the right image.

2) Global methods which extract features from both full images and match them.

3) Methods, which use a combination of the two.

After two corresponding image points $a_L(u_L, v_L)$ and $a_R(u_R, v_R)$ have been found, the disparity value is defined as:

$$d = u_L - u_R, \tag{1.18}$$

where:

d: the disparity value. u_L, u_R : coordinates of the corresponding points in the left and right stereo image.

Figure 1.15 shows an example of a depth image which is proportional to the disparity.

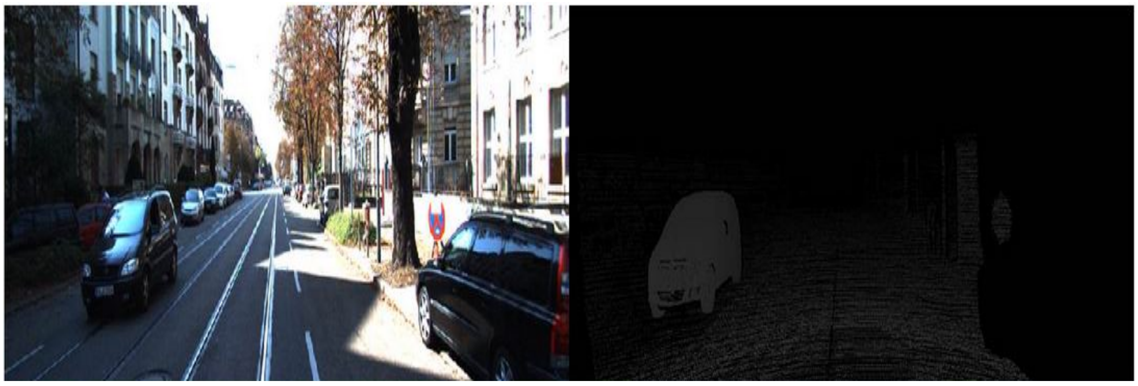


Figure 1.15: Left RGB scene, Right depth image.

3D point reconstruction

If the disparity value d for a point $a_L(u_L, v_L)$ is known, the 3D location of the original point A(x, y, z) with respect to the stereo camera coordinate system is given by:

$$x = \frac{B.(u_L - u_{cL})}{d}, y = \frac{B.(v_L - v_{cL})}{d}, z = \frac{B.f}{d}, \tag{1.19}$$

where:

f:focal length, B: baseline, d: disparity, $c(u_{cL}, v_{cL})$: principal point left cam.

The distance of an object to an on-board stereo camera system can be estimated as:

$$distance = \frac{B.f}{d.p}, \tag{1.20}$$

where:

B: baseline, f: focal length, d: disparity, p: pixel size. The error in estimating the dist is defined as:

$$distance1 = \frac{B.f}{d.p}, distance2 = \frac{B.f}{d.p} \tag{1.21}$$

$$error = distance1 - distance2 = \frac{B.f}{d.(d-1).p} \tag{1.22}$$

Considering a typical pixel size of $6,5\mu m$ and a horizontal resolution of 2048px (maximum disparity is 768px). Figure 1.16 illustrates the variation of the error vs. distance.

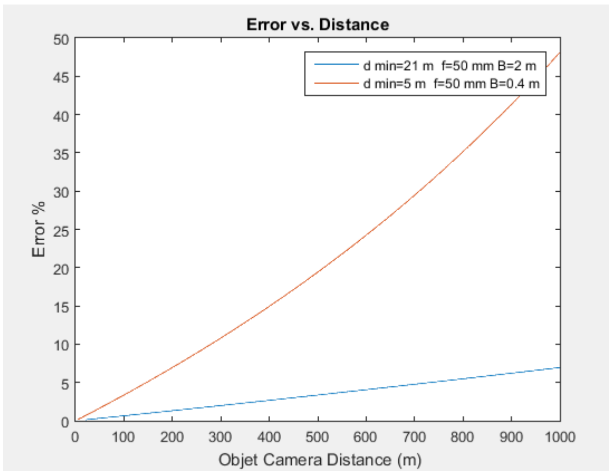


Figure 1.16: Variation of the estimated error vs. distance with respect to focal length and baseline for the long range 20-1000m.

It can be observed that the error increases as the estimated distance increases. Also, it can be seen that as the maximum estimable distance increases, the minimum estimable distance increases too. This indicates the fact that the vision based system can only reliably operate in a certain range. In order to cover the range up to 1000m, the two cameras could be placed on either side of the locomotive, having then a baseline of about 2m. With a focal length of 50mm, the operating range is from about 21m to about 1000m.

In order to cover the short range of 0-50m, a third camera should be added to form a second pair of stereo cameras. If the baseline is 0.4m (the focal length must be equal to the other two cameras), the minimum detectable distance is reduced below 5m.

In conclusion, three cameras set up as two stereo image pairs can cover the range 5-20m with a precision of about 10% and the range 20m-1000m with an error of about 5%.

In order to compensate the error, the stereo-vision system shall be fused with another distance sensor such as 2D laser scanner to increase precision and reliability in distance calculation.

1.2.3 Other methods

Rather than using stereo cameras for depth calculation, there are other methods based on machine learning that estimates depth from a monocular camera [34]. However, the data-sets used for training those algorithms are dedicated to autonomous driving applications. The max range needed is of 100m like tramways or cars [35], while for railway applications we need a minimum range of 500m.

There is also a method called homography, which offers the possibility to map the image plane and the corresponding world plane. As a result of that mapping, the world 3D coordinates of each point in the image world plane can be calculated.

Among other sensors, time-of-flight (ToF) cameras can also provide distance information. The 3D time-of-flight cameras work by illuminating the scene with modulated light sources. The reflected light are then observed [36]. ToF camera produces a depth image, where each pixel encodes the distance of the corresponding point of the captured scene [36]. The distance range is limited by the beam emission power and is also susceptible to motion blur [36].

1.2.4 Evaluation metrics

Root mean square error (RMSE) (1.24) tells us how spread out our data is from our mean. Distance actual is ground truth, distance prediction is the predicted distance, and N is the total number of detections. It is commonly used in regression analysis to verify experimental results. Since distance estimation is a regression problem, RMSE can be used for evaluation [37].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(distance_{actual} - distance_{predicted})^2}{N}} \tag{1.23}$$

1.3 Techniques that combine object detection and distance estimation

There are some works which combine detection of objects and distance estimation based on multi-sensor fusion using the methods described in Sections 1.1 and 1.2.

In [38], a monocular camera for object detection and a millimeter-wave radar for distance estimation are combined. However, their application is short range as it is used in shunting mode where the speed of the train is below 45km/h.

In [39], a system which combines different vision technologies: thermal camera, night vision sensor (camera augmented with image intensifier), multi stereo-vision system and laser scanner is used. Their aim is to create a sensor fusion system for mid (up to 200 m) and long range (up to 1000 m) obstacle detection. This system is complex. It uses traditional computer vision image processing for object detection, stereo images to calculate the disparity, and thermal images to calculate the homography matrix for distance estimation.

The most interesting work is DisNet [40], Long-range obstacle detection from a monocular camera. It is based on learning the change in object appearance in an image (in terms of size) due to the change of the object distance with respect to camera viewing the object. It uses a neural network with 3 hidden layers combined with the Yolo V3 object detection algorithm [40]. It takes captured images from the camera and feeds them to the Yolo V3 object detection. The objects bounding boxes resulted from the Yolo object classification are processed to calculate the features-bounding boxes parameters. Based on the input features, the trained DisNet gives as outputs the estimated distance of the object to the camera coordinate system.

Chapter 2

Algorithm comparison and proposed solution

To detect the obstacles on rails, a multi-sensor system is needed to capture the scenes and an algorithm which combines the object detection with object to camera distance estimation. In this chapter, the different selected sensors are presented together with the chosen approach for the image processing.

2.1 System requirement and specification

The multi-sensor system should satisfy the requirements related to weather and illumination conditions. The fact that the train's speed is so high, the braking distance is about 500 m. The range needed for detection should thus be longer than this specified distance. For that, the range of the camera required is between 800 m and 1000 m.

The system combines two cameras (color and thermal cameras), a computer for the processing, and a switch used to transfer the frames from the cameras to the computer. Figure 2.1 shows the diagram of the proposed system. In this section, the specification of each material composing the system is detailed.

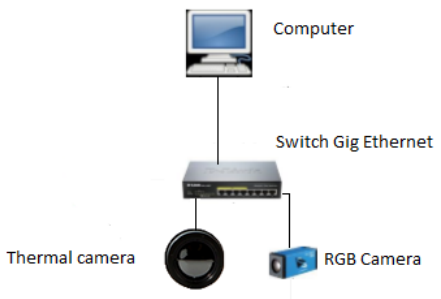


Figure 2.1: The diagram of the proposed system for accident prevention.

2.1.1 RGB Camera

RGB camera is a camera equipped with a standard sensor through, which the colored images of people and objects are acquired. The acquisition of static photos is usually expressed in pixels that define the resolution of an image (length $\times$ height). However, the acquisition of videos is usually expressed with an explicative term which is the number of frames per second.
Before selecting the cameras, some parameters have to be defined.

Focal: The focal length represents the distance from the optical center of the lens to the camera image plane. It is usually expressed in millimeter.

Angle of view (AoV): It describes the angular extent of a given scene that is imaged by a camera.

Field of view (FoV): It represents the length that the lens will cover at a certain distance. It can be measured horizontally, vertically, or diagonally. A shorter focal length gives a wider FoV than a longer one.

Used technology: There are two different technologies, analog technology, and digital technology, also known as IP.

- Analogical technology: This technology is based on the use of coaxial cables as the old video installations.
- IP technology: IP cameras are connected to a network switch and a computer to manage and store videos and related software.

In our case, IP technology is chosen, because it maintains the same resolution of the sent images, with a high rate of transfer, and its wireless opportunities.

The specification of chosen camera

For the choice of the camera, three parameters are studied to suit our work: The resolution of the camera, the number of frames taken per second and focal length. Table 2.1 presents the specification of the camera.

Interface	GigE
Sensor	CMOS 1/ 2.25"
Sensitivity	0.05 lx (light from full moon)
Resolution	5MP
Frame rate	15 frames per second
Focal	4.8mm to 57.6mm
Dimensions	50 mm × 50 mm × 103 mm
Weight	330 g

Table 2.1: RGB camera specification.

The camera of choice is the Gigabyte imaging camera, because of its long-range and high resolution. This camera is categorized as a GigE interface camera that offers a high data transfer rate, high bandwidth, and PoE functionality.

2.1.2 Thermal Camera

The thermal camera is used in difficult weather conditions as large drops of rain, or snowflake, and also at night because its operating range is in the invisible infrared region of the spectrum. Thermal imaging cameras can work by seeing heat instead of reflecting light. It converts thermal energy into visible light, which makes it different from RGB cameras. For this reason, thermal imaging can operate also at dark places or during cloudy or at any abnormal conditions. The generated image is called thermogram and the process of analyzing that image is thermography.

We are interested in the range of the camera; (the distance of a given target that can be seen with a thermal); to satisfy our conditions.
We define the criteria used in thermal imaging to estimate the range of the camera; the Johnson criteria. These criteria allow estimating the range for three different types of view: Detection, Recognition, and Identification (D, R, I).

- Detection: detect the presence of an object. For this, the dimensions of the object should be covered by 1.5 pixels or more.
- Recognition: specific the type of the object. For example, person / vehicle. For this, the dimensions of the object should be covered by 6 pixels at least.
- Identification: This term is often used in the military sense of the word, which means seeing if someone is “friend or foe”. In order to do this, the critical dimension of the object in question needs to be subtended by at least 12 pixels.

Figure 2.2 represents the range of each type of view for different values of the focal lens.

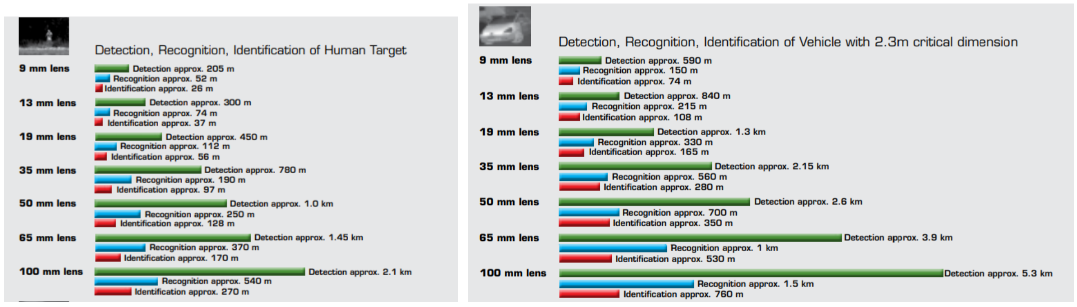


Figure 2.2: Range performances for D, R, and I with 320x240 pixels detector [41].

A critical parameter that affects the range of a thermal imaging camera is the focal length of the lens. The focal length determines the instant field of view (IFoV) of the camera. An excellent method to estimate how far we can detect a target with a thermal imaging camera is to use a nomograph. A nomograph is a graphical calculator that demonstrates the relationships between variables, such as focal length, range, and the number of pixels on target. The following nomographs are simplified models for estimating the range at which a man can be detected, recognized, or identified.

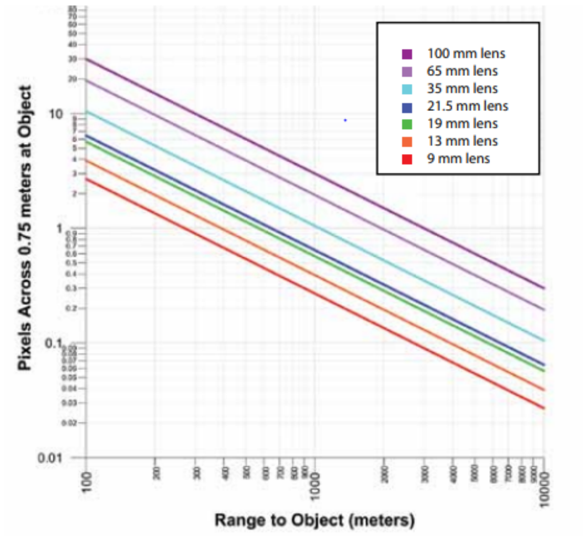


Figure 2.3: Nomograph: uncooled 320 x 240 detector [41].

The specification of chosen camera

In our application, small IFoVs are required, as we need to detect objects the size of a man or car at a distance of one kilometer away from the train. We know that the total field of view is inversely proportional to the focal length. The long lens provides a small field of view and a large distance.

Table 2.2 presents the characteristics of the chosen camera: If the focal length of the lens is large, the IFoV becomes small, which translates into more pixels across a target at a fixed range.

Resolution	640 × 512 Pixels
Fov	6.2° H × 5° V
Focal	100 mm
Pitch	17 um
Weight	479 g

Table 2.2: Thermal camera specification.

The chosen thermal camera is the FLIR TAU2 model with a resolution of 640 × 480 pixels and a lens of 100 mm. It has a small size as well as the possibility of extending it with an adapter for Gigabit Ethernet communication.

2.1.3 Switch

The switch is used to establish a connection between the cameras via Ethernet using Ethernet cables, and likewise. It is connected to the data processing and storage computer.
The static IP addresses should be configured on each device. Tests are required to verify the connectivity between the sensors and the switch, and the connectivity between the sensors and the data processing and storage computer (using ping).

The specification of the switch

Table 2.3 outlines the specifications of the needed switch:

Ports	8 ports 10 Gbit
Processor speed	600 MHz
Number of cores	1
memory size	512 MB
Dimension	20.3 × 43.9 × 4.3 cm
weight	2.61 Kg

Table 2.3: The switch specification.

2.1.4 Processing and data storage computer

The computer is used to store the data and process it in real-time. It is connected to the network switch.
Software setup (installation of required software packages):

- Ubuntu or windows (Operating System).
- OPEN CV [42]: a library used for image processing and computer vision.
- Qt [43]: used to program the interface.

The hardware specification of the computer:

System	
CPU	Intel Gen 3 Core i7 1.7GHz
Memory	8 x DDR3 1600MHz SO-DIMM
Power requirement	
Power input	9V -32V DC
Graphics	
Graphics	Intel HD Graphics 4000 with video acceleration
Resolution	Up to 1920 x 1200
Dimensions	
Dimensions	250 x 150 x 55 mm
weight	1780 g

Table 2.4: The computer hardware specification.

2.2 Data-set collection and labeling

For the evaluation of obstacle detection algorithm accuracy, a data-set is needed to calculate the different evaluation metrics described in Chapter 1.

2.2.1 Data-set collection

Due to the lack of data-set dedicated to railways application, we have manually collected a set of 200 images from Google Images. These images have been selected with respect to the following criteria:

- The viewpoint of images; the scenes should be front view captured from the train.
- The images quality; the object should be clear and not blurry.
- The resolution; the images should have a sufficient (like the one we would have in practice).

These images contain the most common obstacles that can be present on railways including the three super categories: person, vehicle, and animals with 65 to 67 images per category.

Where:

- Vehicle category contains car, truck, bus, train, motorcycle, and bicycle.
- Animal category contains cow, horse, dog, and cat.

The Table 2.5 represents the number of object in each category:

Category	Objects per category
person	408
vehicle	255
animal	174

Table 2.5: Description of the different categories in the collected test data-set.

Few images from the collected data-set are depicted in Figure 2.4:

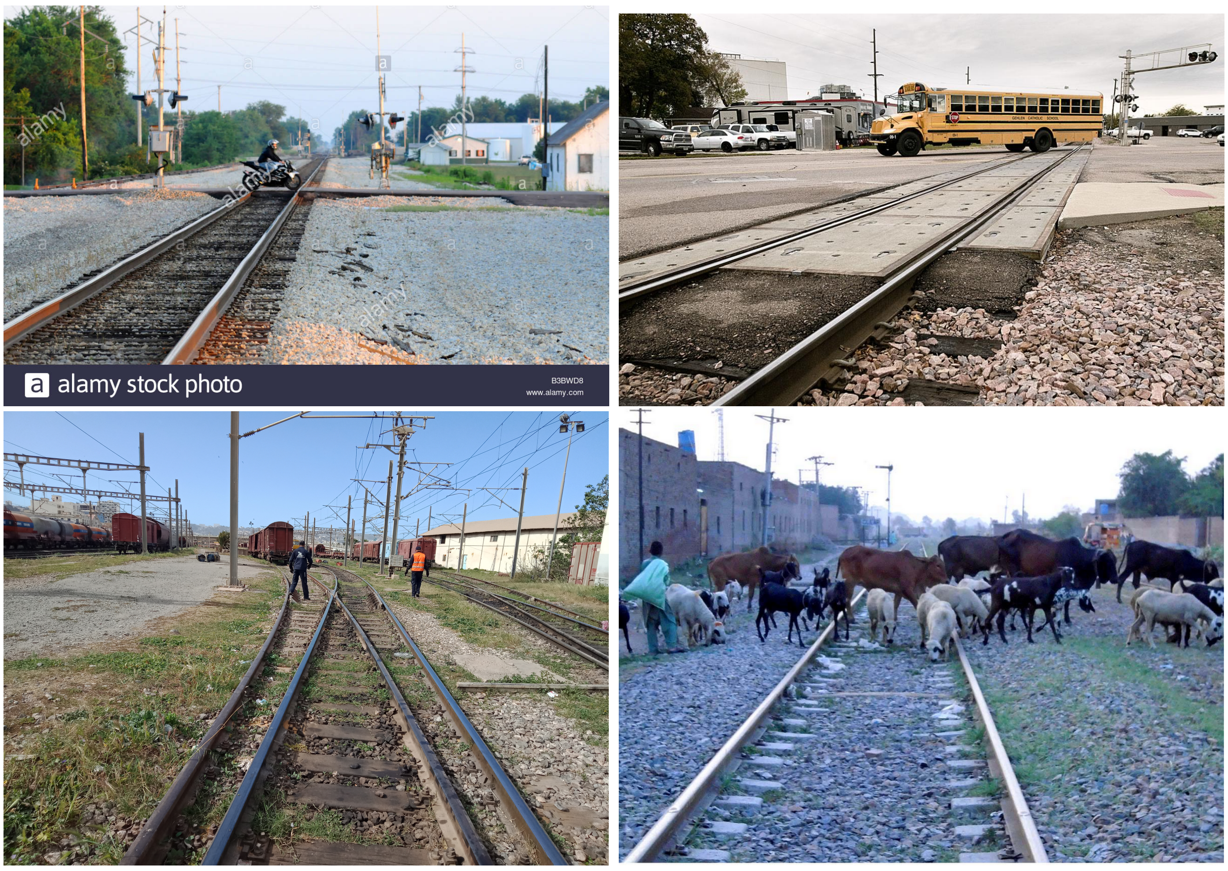


Figure 2.4: Exemples of images from the collected test data-set.

2.2.2 Data-set labeling

After collecting images, the next step is the image labeling and annotation. This task requires a lot of manual work.

There are different types of image annotation such as bounding boxes, 3D cuboids, polygonal and semantic segmentation. The type of annotation data that has been used here is bounding boxes, which is the most commonly used type of annotation in computer vision. It is generally used in object detection and localisation tasks.
Bounding boxes are rectangular boxes used to define the location of the target object. They can be determined by the x and y axis coordinates in the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle.

Some tools used for image annotating:

- MakeSense.AI [44].
- LabelImg [45].
- VGG image annotator [46].
- LabelMe [47].

The online tool used in this work for image annotation is Makesense [44]. It is an open source and free tool under GPLV3 license. It does not require any advanced installation. Just a web browser is needed to run it. Furthermore, it has a simple and easy user interface, that is the main reason for choosing this tool.
This tool takes images as input and after delimiting manually each object present in the image with a rectangular bounding box as shown in Figure 2.5 and selecting the class of these objects. It gives as output a file for each image containing the class of the detected object and its bounding boxes information in Yolo format as follow:

<class name> <left> <top> <right> <bottom>,

where left, top, right, and bottom are the bounding box coordinates.

Figure 2.5 shows an example of image labeling with makeSense.

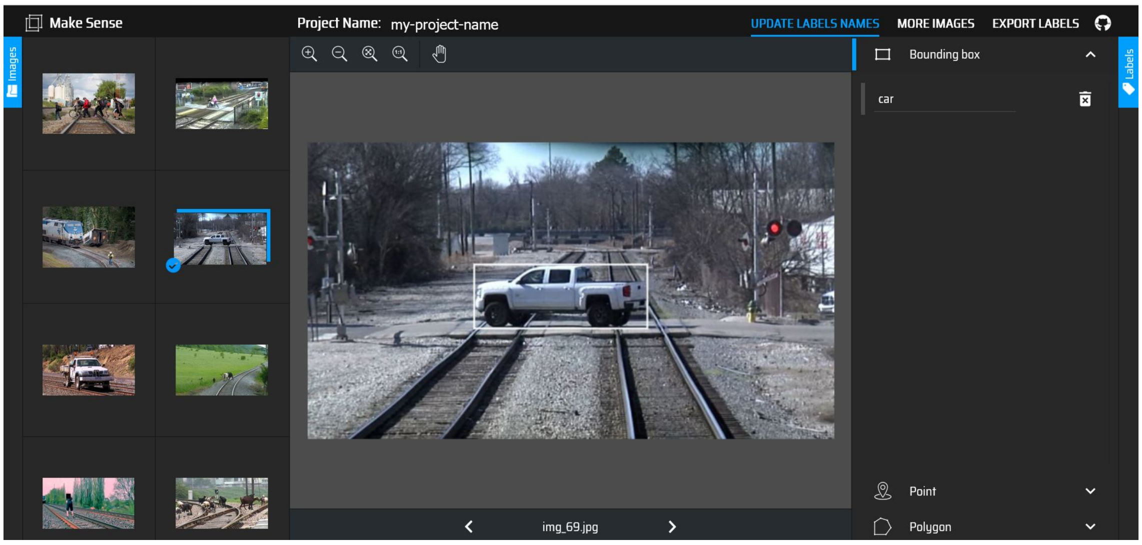


Figure 2.5: Data labeling with makeSense [44] tool.

2.3 Evaluation of object detection algorithms

In this section, the object detection part is studied by comparing together the effectiveness and the efficiency of the common object detection algorithms.

2.3.1 Effectivness comparison of obstacle detection algorithms

To fulfill the application requirements and create a reliable system, the most common 3 object detection algorithms (Yolo V3 [17], Faster RCNN [14], SDD [15]) have been evaluated. The best one which meets the needs of our application is then chosen. A system with a minimum number of false alarms is required to avoid stopping the train every time even in absence of obstacles. A maximum number of detection rate is needed, in order to detect all the obstacles on the railways. For that mAP, Precision, Recall, Detection rate, and False Alarm Rate have been calculated. The mAP is measured with the PASCAL VOC testing set.

Where the false alarm rate (FAR) is defined as follow:

$$FAR = \frac{FP}{FP + TP} \tag{2.1}$$

The parameters already cited (mAP, FAR, Recall, Precision) are influenced by two important values; a threshold of intersection over Union (IoU) in the function non maximum-suppression (NMS) and the confidence score of classes. In order to obtain a good comparison, we have tried to take different values of them.
The results are presented in the tables below, followed by interpretations.

YOLO V3 Evaluation

Both confidence score and IoU thresholds have been fixed to 3 values: 0.25, 0.5, and 0.7.
An evaluation has been performed to the algorithm for each combination by calculating the mAP, FAR, P, R, and F-score, as shown in Table 2.6. F-score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score. The weight of Yolo V3 used in this work is trained by [17], using the COCO data-set [12].

Confidence threshold	0.25			0.5			0.7		
IoU threshold	0.25	0.5	0.7	0.25	0.5	0.7	0.25	0.5	0.7
map(%)	92	91	84	84	83	79	74	74	70
FAR(%)	19	22	43	05	06	22	02	03	13
P(%)	81	78	57	95	94	78	98	97	87
R = Dr(%)	92	94	94	84	85	85	73	73	74
F-score	0.86	0.85	0.71	0.891	0.892	0.81	0.84	0.83	0.78

Table 2.6: Yolo V3 evaluation.

SSD Evaluation

The same procedure used for Yolo V3, has been applied to SSD (Table 2.7) and Faster RCNN(Table 2.8) algorithms.
The weight of SSD used is trained by [15], using Pascal VOC data-set [13].

Confidence threshold	0.25			0.5			0.7		
IoU threshold	0.25	0.5	0.7	0.25	0.5	0.7	0.25	0.5	0.7
map(%)	61	56	48	52	50	44	47	45	40
FAR(%)	44	57	68	32	46	61	23	38	53
P(%)	56	43	32	68	54	39	77	62	47
R = DR(%)	69	74	87	58	62	63	49	53	53
F-score	0.61	0.54	0.47	0.63	0.58	0.48	0.60	0.57	0.50

Table 2.7: SSD evaluation.

Faster R-CNN Evaluation

The weight of Faster R-CNN used in this work is trained by [48], using the COCO data-set [12].

Confidence threshold	0.25			0.5			0.7		
IoU threshold	0.25	0.5	0.7	0.25	0.5	0.7	0.25	0.5	0.7
map(%)	84	83	80	83	83	80	79	79	77
FAR(%)	38	47	65	35	44	63	23	28	44
P(%)	62	53	34	65	56	37	77	72	56
R = DR (%)	92	93	93	90	91	91	85	86	86
F-score	0.74	0.68	0.50	0.75	0.69	0.53	0.80	0.78	0.68

Table 2.8: Faster R-CNN Evaluation.

Interpretation

We summarize the variation of the metrics versus the thresholds of IoU and score confidence in the table below.

Thresholds	mAP	FAR	P	R
IoU $\nearrow$	$\searrow$	$\nearrow$	$\searrow$	$\approx$
confidence$\nearrow$	$\searrow$	$\searrow$	$\nearrow$	$\searrow$

Table 2.9: Metrics variation vs. IoU and confidence score thresholds.

When choosing a high confidence threshold, the model becomes robust to positive examples (i.e. boxes containing an object). Hence, there will be less positive predictions. As a result, false negatives increase, and false positives decrease. This reduces the mAP, the FAR, and the detection rate or the recall (the denominator increases in both FAR and recall formulas) and improves precision (denominator decreases in the precision formulas).

Similarly, further lowering the confidence threshold causes the precision to decrease and the recall or the detection rate with both mAP and FAR values to increase.

For that, a medium value of confidence score threshold is recommended.

The higher the IoU threshold, the higher the double detections (i.e. many boxes for one object). Hence, there will be more false negatives. As a result, both of the mAP and precision decrease (the denominator increases in the precision formula) and FAR increases. When considering double detection as FP, the TP values are not affected, so the Recall value (or DR) stays approximately the same (for a fixed value of confidence only).

Similarly, further lowering the IoU threshold causes the precision and mAP to increase while the FAR value decreases. So a small value of IoU is recommended.

High recall (DR) but low precision implies that all ground-truth objects have been detected, but most detections are incorrect (many false positives). That causes the augmentation of false alarm rate and the diminution of the mAP.

Low recall (DR) but high precision implies that all predicted boxes are correct, but most ground-truth objects have been missed (many false negatives). That causes the diminution of DR, FAR, and mAP also.

High values of precision, recall or detection rate, and mAP with low FAR is the ideal detector which has most ground-truth objects detected correctly.

For each algorithm, the thresholds which give the best results (low FAR, High DR & mAP) have been selected and the final comparison is in the table 2.11.

2.3.2 Efficiency comparison

For time execution, the comparison of the number of processed frames per second by the algorithms, and the total processing time is shown in the table 2.10. Approximately the same image resolution have been used.
The environment used for the inference is *Colaboratory* [49], which is a Jupyter notebook environment that runs entirely in the cloud.
Environment specifications:

- CPU : Intel(R) Xeon(R) CPU @ 2.30GHz.
- RAM : 12 Gb.
- GPU: Nvidia Telsa K80.

The properties of the video used to calculate the number of processed frames per second:

- Duration: 00:05:29.
- Size of frame: 1280 × 720.
- Number of total frames: 4935 frames.

Algorithm	Yolo V3	SSD	Faster RCNN
Resolution	608 × 608	512 × 512	800 × 800
Number of processed frames per second	12	6	4
Total processing time (min)	7	14	20

Table 2.10: Number of processed frames per second by the three object detection algorithms.

Interpretation

Faster R-CNN is very slow. Its speed is about 4 processed frames per second due to the use of a pipeline, where first object proposals are generated by the RPN (Regional Proposal Network) and then they are sent to classification. These two neurons can not process at the same time, because the outputs of RPN (Regional Proposal Network) are the inputs of the classifier. Thus, this model cannot run for real-time applications. Regarding the remaining approaches, the SSD can attain on average 7 processed frames per second, but still slower than Yolo V3. This one uses a pyramidal hierarchy, which consists of several convolutional layers of different sized default boxes that allow the detection of objects at different scales which is computationally more expensive. On the other hand, Yolo V3 has shown a very fast processing time and can achieve 12 processed frames per second due to the use of one network to treat the complete image.

Final comparison

The results of the comparison between the top three configurations are presented in Table 2.11, based on highest rate detection and mAP with a minimum of false alarm rate, and the number of processed frames per second:

Algorithm	Yolo V3	SSD	Faster RCNN
IoU threshold	0.5	0.25	0.25
Confidence threshold	0.5	0.7	0.7
MAP (%)	84	47	33
FAR(%)	6	23	13
P(%)	94	77	87
DR =R(%)	85	49	46
Number of processed frames per second	12	6	4
Total processing time (min)	7	14	20

Table 2.11: Final comparison of the 3 object detection algorithms.

Conclusion:

As we can see from Table 2.11, Yolo V3 gives the lowest false alarm rate with the highest values of mAP and detection rate. It also achieves the best execution time. All these characteristics makes of Yolo V3, the best choice for obstacle detection algorithm.

2.4 The ML algorithm for obstacle detection and distance estimation

In this section, we detail the second part of the algorithm to use in the application, which is the estimation of object distances. The training data-set used, the structure of the network, and an evaluation of the performance of the distance estimation algorithm.

2.4.1 DisNet

The algorithm chosen for the estimation of the distance between the obstacle and the train is DisNet [40]. The main reasons are: The data-set used for the training has been captured on

railroad of long range. The results (distances) are independents of the input image size.
As described in Section 1.3, the camera image is the input to the Object Classifier which is based on a state-of-the-art computer vision object detector Yolo V3 trained with the COCO data-set [15]. The object bounding boxes resulted from Yolo V3 are then processed to calculate the features, and bounding boxes parameters. After that, based on the input features, the trained DisNet [40] gives as outputs the estimated distance of the object to the camera sensor. In the system architecture illustrated in Figure 2.6, an example of the estimation of distances of two persons on the rail tracks is shown.

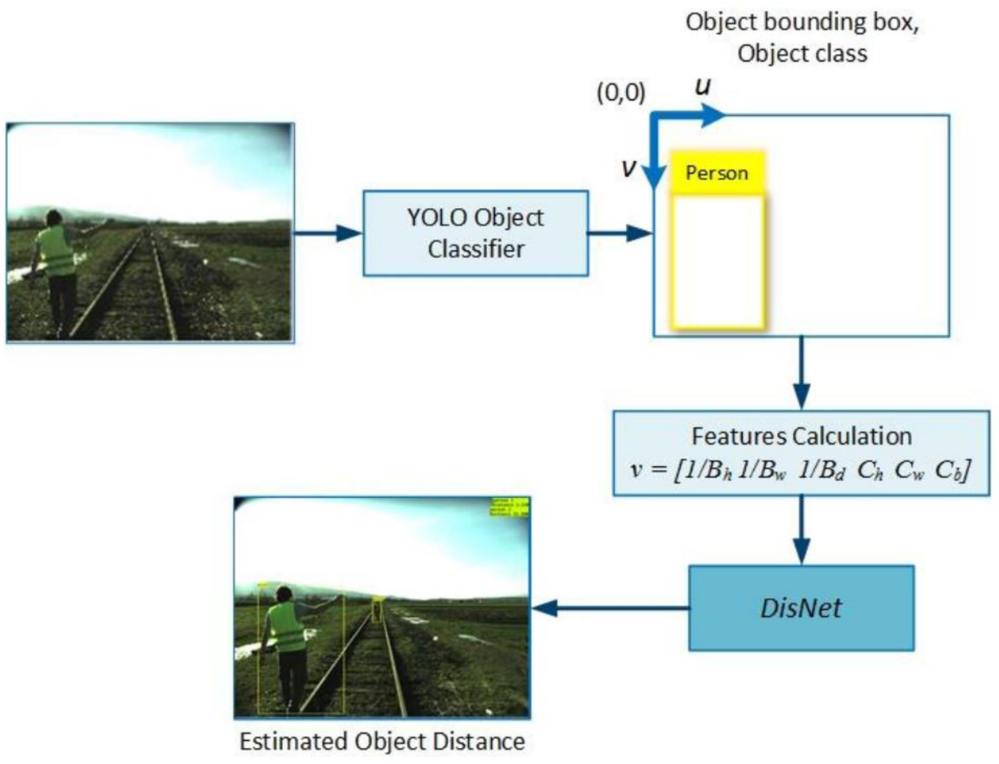


Figure 2.6: DisNet based system for object distance estimation [40].

Where B_h, B_w, B_d being height, width and diagonal of bounding box, and C_h, C_w, C_d are average width, height and breadth of an object of a particular class.

Training data-set

The model used in this work have been trained on 2000 feature vectors created manually from bounding boxes of different objects in the images recorded with RGB cameras by [40]. The distances of those objects from the camera have been measured with the accurate laser scanner.
For each extracted object bounding box, a six-dimensional feature vector ν is calculated:

$$\nu = \begin{bmatrix} \frac{1}{B_h} & \frac{1}{B_w} & \frac{1}{B_d} & C_h & C_w & C_b \end{bmatrix}, \tag{2.2}$$

where the coordinates of the vector ν, features, are:

- Height B_h: height of the object bounding box in pixels/image height in pixels.
- Width B_w: width of the object bounding box in pixels/image width in pixels.
- Width B_w: width of the object bounding box in pixels/image width in pixels.

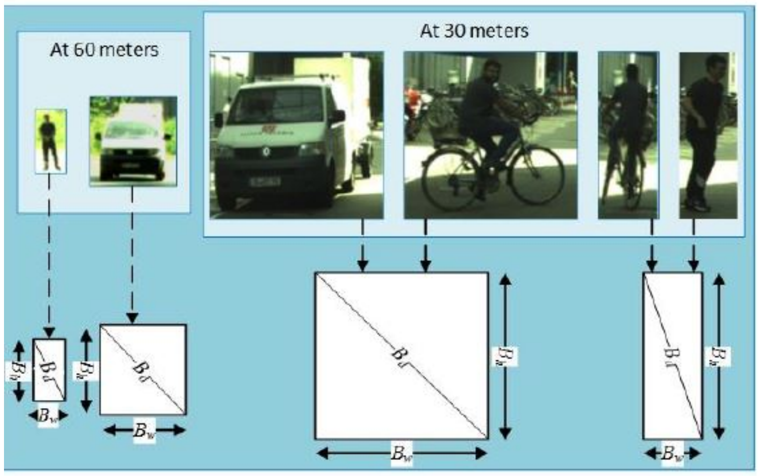


Figure 2.7: Examples from the DisNet dataset of different object bounding boxes in the RGB images [40].

The ratios of the object bounding box dimensions to the image dimensions B_h, B_w, and B_d enable the reusability of DisNet trained model with a variety of cameras independent of image resolution. C_h, C_w, and C_b in Equation (2.2) are the values of average height, width, and breadth of an object of a particular class. For example, for the class $person$ C_h, C_w, and, C_b are respectively 175 cm, 55 cm, and 30 cm, and for the class "car" 160 cm, 180 cm, and 400. The features C_h, C_w, and C_b are assigned to objects labeled by the Yolo V3 classifier as belonging to the particular class. They complement 2D information on object bounding boxes and give more information to distinguish different objects [40].

For training the network, the input data-set has been firstly randomly split into a training, validation, and test set respectively 80% ,10% and 10% of the data.
The DisNet has been trained using a supervised learning technique which required a collected data-set including both inputs and outputs, i.e. the ground truth. This method is the back-propagation method with Adam optimizer [50].

DisNet structure

DisNet is multi-layer neural network for distance estimation. In order to determine the number of hidden layers and hidden neurons per layer, several tests have been performed by [51].

Assuming that each hidden layer has 100 neurons. The accuracy and Mean absolute error of distance estimation over 1000 epochs for different number of hidden layers have been calculated. The results show that DisNet with one hidden layer achieves the lowest distance

estimation accuracy. It is also obvious that there is no significant difference in distance estimation accuracy achieved with 2, 3, 5 and 10 hidden layers.
For this analysis, a reduced data-set has been used [51]. The network has been trained on the 80% data-set and the estimation accuracy reported is on the 10% validation set.

The smallest values of Mean Absolute Error and also the accuracy of the distance have been achieved for 10 hidden layers. So a trade-off has been made between the computational time and accuracy/error and finally, DisNet with 3 hidden layers has been chosen.

After the determination of the number of hidden layers, experiments have been performed [51] to find the appropriate number of neurons in each hidden layers. The results show that the distance estimation accuracy achieved with 10 hidden neurons is very low, much lower than the one achieved with 30, 100 and 200 hidden neurons. The distance estimation accuracy with 30 hidden neurons is also lower than with 100 and 200 neurons. There is no significant difference in distance accuracy estimation with 100 and 200 hidden neurons, in order to reduce the complexity of DisNet 100 neurons per hidden layer have been chosen.
DisNet has 3 hidden layers with 100 hidden neurons per layer as shown in Figure 2.8 [51].

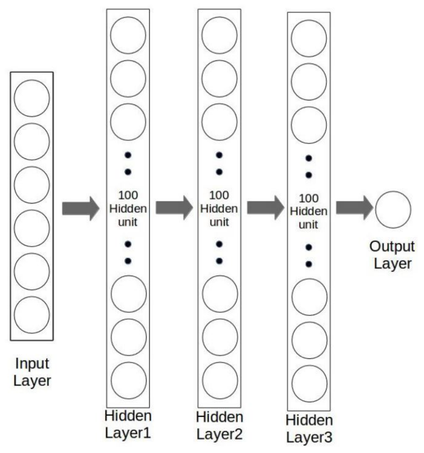


Figure 2.8: The structure of DisNet used for object distance predictions [40].

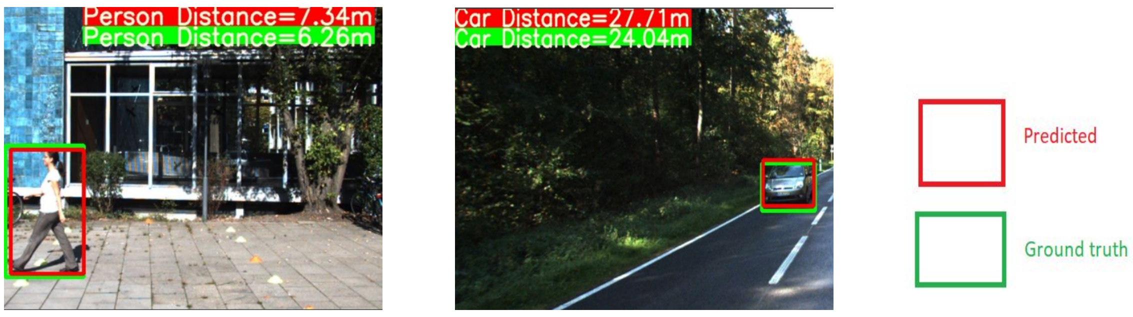
DisNet input layer consists of 6 neurons corresponding to 6 features, parameters of output layer consists of only one single neuron. The output of this node is the estimated distance between the camera and the object viewed with the camera.

2.4.2 DisNet evaluation

To evaluate the DisNet algorithm, ground-truth distances data-set is required for the comparison with DisNet distances results. Because of the lack of data-set for railway applications, a data-set dedicated to autonomous car application called "Kitti" [52] has been used. The maximum range is about 100m.
A set of 200 objects have been collected from Kitti data-set with ground-truth lidar distances [52]. The data-set contains the following objects: Car, bus, truck, bicycles, and persons.

Two methods for distance evaluation have been used. The first one is the RMSE metric. The second one consists of setting a threshold value for a tolerated error and then calculating the error rate.
Noting that the model used for distance estimation "DisNet" takes as input, a vector of 6 parameters calculated from Bboxes and image size (as describes in Section 2.3.1).
The evaluations have been done for the predicted distances using the bboxes predicted from Yolo V3 and the ground-truth bboxes. These two evaluations are done in order to verify the impact of the predicted bboxes from Yolo V3 on the estimated distance from DisNet.

In Figure 2.9, the two bboxes of Yolo V3 and the ground-truth are drawn. Green ones are ground truth, and red ones are predicted from Yolo V3.



(a) Pedestrian example (b) Car example

Figure 2.9: Distance estimation with Yolo V3 and ground-truth Bboxes.

Table 2.12 shows the impact of the precision of detecting bbox on the precision of distance estimation using *DisNet*. Hence, both the RMSE and error average values have been calculated between:

- The actual distance (ground-truth) and the predicted from DisNet with Yolo V3 bboxes.
- The actual distance and the predicted from DisNet with ground truth bboxes.

Where the average error is calculated as follow:

$$AverageError = \frac{\sum_{i=1}^{N} |distance_{actual} - distance_{predicted}|}{N}, \tag{2.3}$$

with:

$distance_{actual}$ is the ground-truth distance,

$distance_{predicted}$ is the estimated distance from DisNet,

N is equal to the number of object.

The evaluation of the estimated distance using ground-truth bboxes and predicted bboxes from Yolo v3 is shown in Table 2.12 and Table 2.13.

Evaluated BBox	ground-truth	Yolo V3
RMSE (m)	7.82	8.53
Average Error (m)	5.09	5.87

Table 2.12: Distance evaluation using RMSE and average error.

For the second method, different values of tolerated error between 5m to 10m have been chosen to precise whether the estimated distance is acceptable or not. If the difference between the ground truth distances from "Kitti" and the estimated distances from DisNet is higher than this value, the detection is true. Otherwise, it is considered as an erroneous value. Table 2.13 presents the error of the estimated distance with DisNet for the bounding boxes of Yolo V3 and the bounding Boxes of the ground-truth for different values of the tolerated error.

Evaluated BBoxes / Tolerated error (m)	ground-truth (%)	Yolo V3(%)
5	34.74	44.60
6	31.92	35.21
8	21.13	27.70
10	17.37	18.30

Table 2.13: Error of the estimated distance for the different values of the tolerated error.

Interpretation

The RMSE value is about 8.5m for Yolo V3 bounding boxes. This means that for the range 0-100m, the error between actual and predicted distance using Yolo V3 bounding boxes is about +/- 8.53 meters. However,using ground-truth bounding boxes, the RMSE is about +/- 7.82 meters.
For the average error, it is about +/- 5.87m using the predicted bounding boxes from Yolo V3, and +/-5.09 using ground truth bounding boxes as shown in the table 2.12.

Table 2.13 shows the distance error rates when setting different thresholds of tolerated error from 5m to 10m. As we can see, for 10m tolerated error, the error rate on the estimated distance is about 17.37 % using the ground truth bboxes. However, it is about 18.30 % using the predicted bboxes from Yolo V3 which is acceptable in our application.

From the presented results, we conclude that the prediction of bboxes does not affect so much the accuracy of the estimated distances (the distance errors using Yolo V3 and ground-truth bboxes are roughly similar).

Chapter 3

Results and Experiments

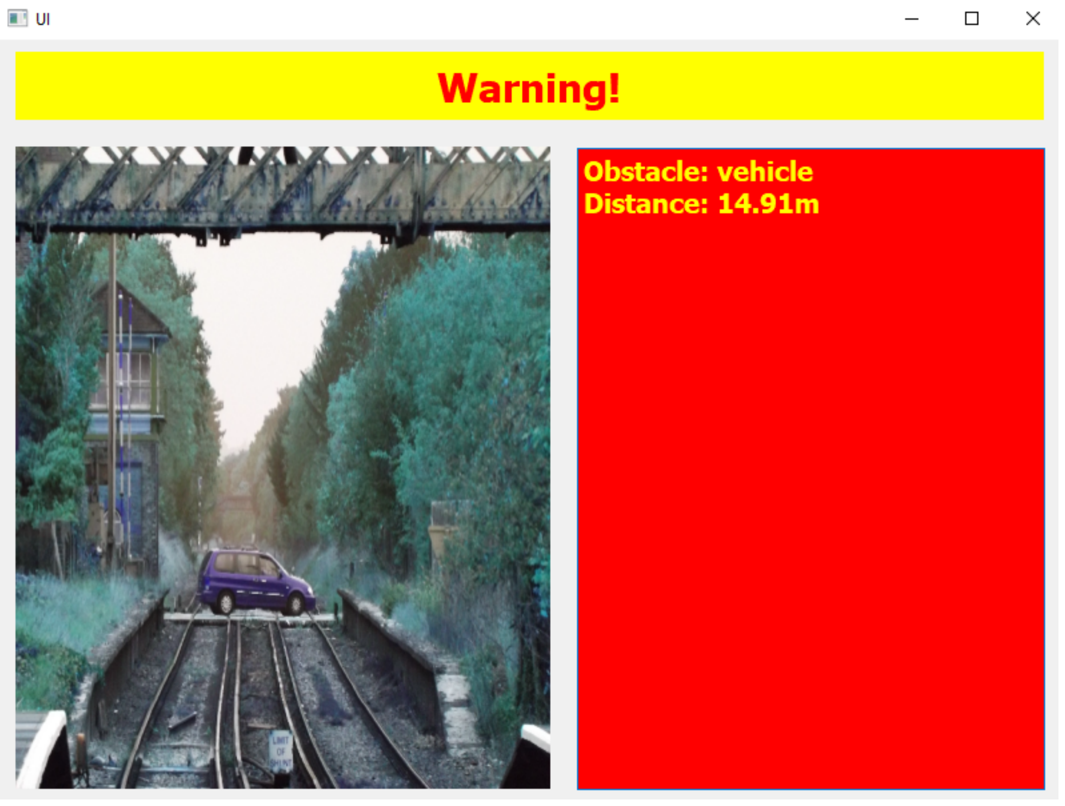
In this chapter, the application development and user interface are described. The results of the different tests made on the chosen algorithm to determine the best resolution of the processed frames are then presented.

3.1 Application development

This application has been developed on a computer with i5 CPU and 8Gb Ram on Python. As described in the previous chapter, two neural networks are used: Yolo V3 for obstacle detection, and DisNet for distance estimation.
For Yolo V3 the weight trained by [17] on the COCO data-set [12] with 80 classes is used. These classes have been filtered to detect only the categories present on railways.
For DisNet the weight trained by [40] on 2000 obstacles captured on railways, with their ground-truth distances is used. This distances are measured with Lidar sensor.
Both of the pre-trained weights have been built using Keras [53], which is TensorFlow's [54] high-level API for building and training deep learning models. The frames are processed with OpenCV [42] library. OpenCV is a cross-platform, used for real-time computer vision and image processing.

3.2 User Interface of the Application

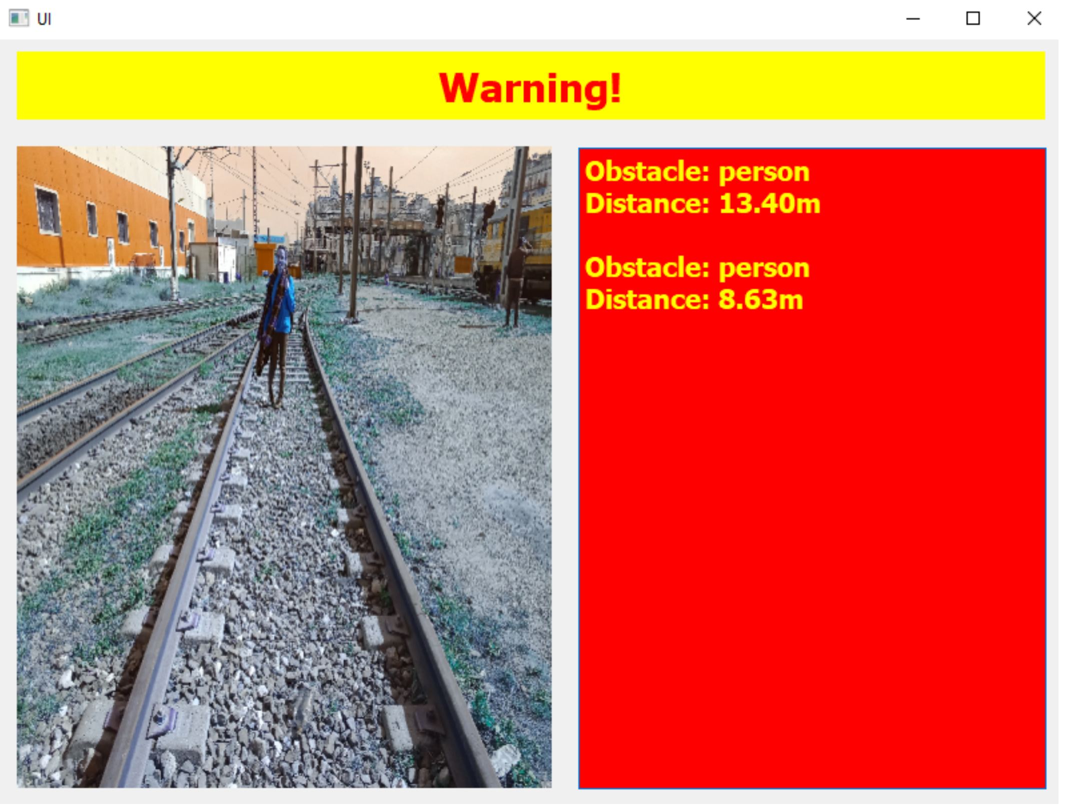
After the processing of the frames captured on the road, the next step is creating a system to warn the train conductor. For this reason, a visual alarm which is a user interface has been developed with PYQT5.
Whenever an obstacle is detected, a warning appears on the screen with the scene specifying the class of the object detected and its distance to camera estimated value as seen in the Figures.



Warning!

Obstacle: vehicle
Distance: 14.91m

Figure 3.1: UI in presence of a car.

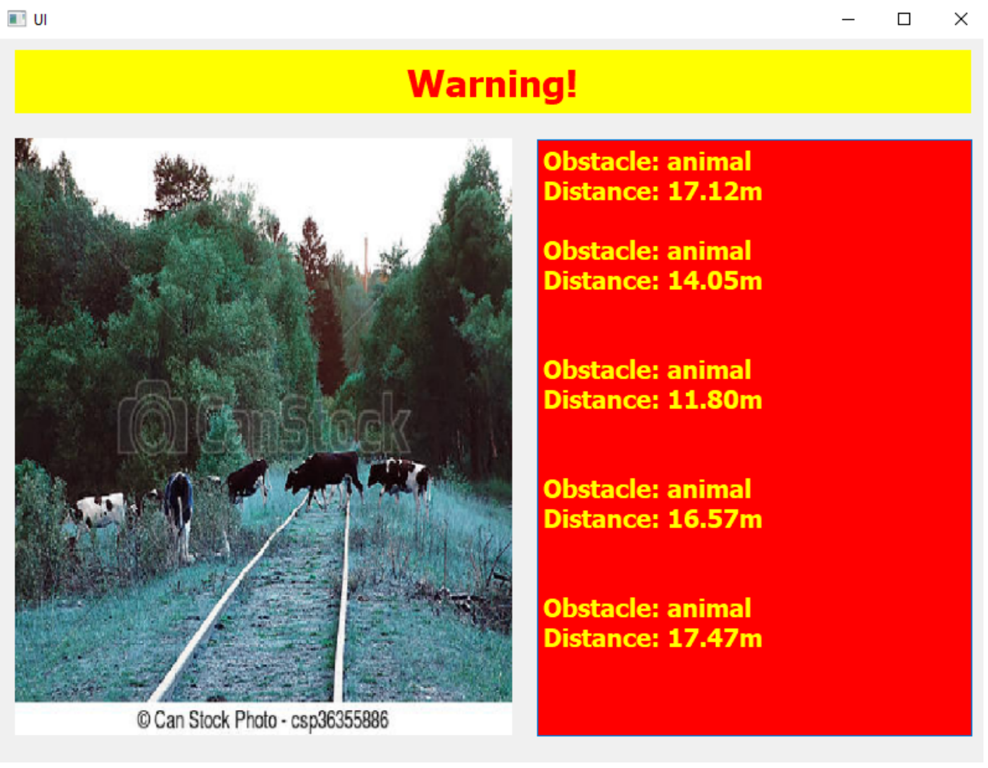


Warning!

Obstacle: person
Distance: 13.40m

Obstacle: person
Distance: 8.63m

Figure 3.2: UI in presence of a person.



Obstacle: animal
Distance: 17.12m

Obstacle: animal
Distance: 14.05m

Obstacle: animal
Distance: 11.80m

Obstacle: animal
Distance: 16.57m

Obstacle: animal
Distance: 17.47m

Figure 3.3: UI in presence of animals.

If there are no obstacles this window is shown.

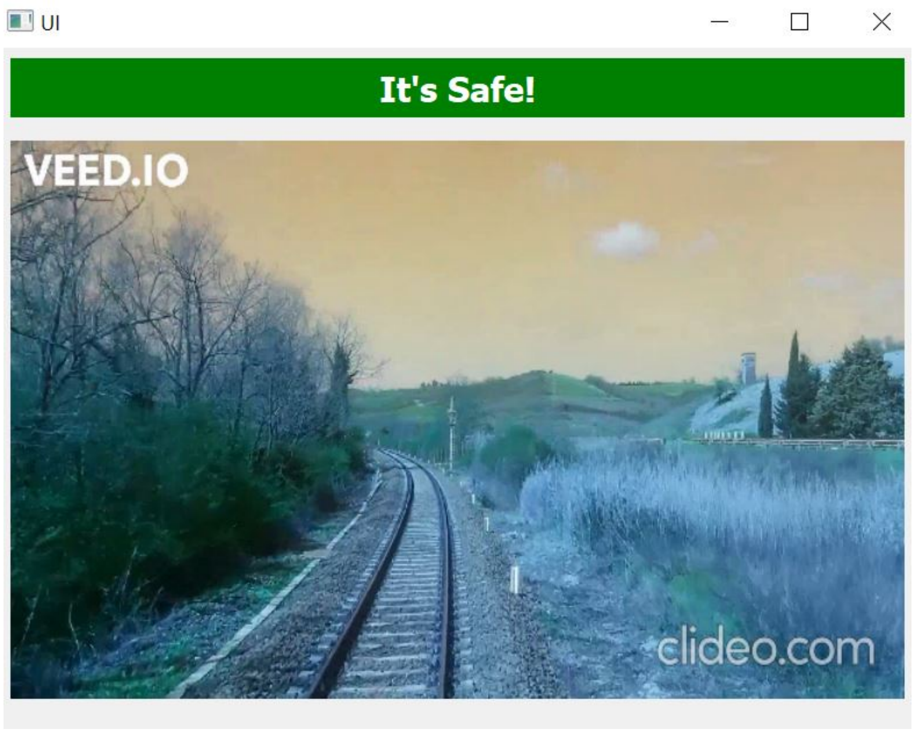


Figure 3.4: UI in absence of obstacles.

3.3 Results

Since the Yolo V3 presented the best results with respect to the other algorithms in the same conditions, there is still one important parameter to evaluate its influence on efficiency and effectiveness. This parameter is the resolution of input images to Yolo V3.

Reducing the resolution can increase the time of frame processing, but it may affect negatively the accuracy of the algorithm. In this section, both efficiency and effectiveness are evaluated with respect to the size of input images.

Efficiency

For time execution, the number of frames processed per second is calculated for different input image resolutions in three different environments using GPU and without GPU. The test data-set described in section 2.2.1 has been used.

The properties of the video used to calculate the number of processed frames per second:

- Duration: 00:05:29.
- Size of frame: 1280 × 720.
- The total number of frames: 4935 frames.

Environment 1 (without GPU)

Device specifications:

- Processor: i3 CPU 1.80 GHz.
- RAM: 4.00 Gb.

Algorithm	YOLO V3			
Resolution	320 × 320	416 × 416	608 × 608	960 × 960
Number of processed frames per second	0.39	0.20	0.10	0.04

Table 3.1: Number of processed frames per second achieved by Yolo V3 vs. image resolution.

Environment 2 (without GPU)

Device specification:

- Processor: i5 CPU 2.4 GHz.

- RAM: 8.00 Gb.

Algorithm	YOLO V3			
Resolution	320 × 320	416 × 416	608 × 608	960 × 960
Number of processed frames per second	0.83	0.38	0.26	0.10

Table 3.2: Number of processed frames per second achieved by Yolo V3 vs. image resolution.

Environment 3: Colaboratory (with GPU)

Specifications:

- CPU : Intel(R) Xeon(R) CPU @ 2.30GHz.
- RAM : 12 Gb.
- GPU: Nvidia Telsa K80.

Algorithm	YOLO V3			
Resolution	320 × 320	416 × 416	608 × 608	960 × 960
Number of processed frames per second	21.04	18.08	12.30	7.39

Table 3.3: Number of processed frames per second achieved by Yolo V3 vs. image resolution.

Interpretation

The obtained results show that the higher the resolution, the lower the number of processed frames per second.

Reducing the image size means that the input of the network will be smaller. This allows the processing to be less heavy, which reduces the total execution time, so the number of the processed frames per second increases.

The device specification also influences the speed of the algorithm as illustrated. Using the GPU extremely increases the number of processed frames per second.

The fact that GPU is intended for the long and heavier computations with much bigger data. So, the execution of Yolo V3 will be way faster on GPU.

Effectiveness

For the evaluation of the effectiveness, the metrics defined previously were calculated for different images resolution as shown in the table below.
The data-set used here is the same one used in chapter 2.

Algorithm	YOLO V3				
Resolution	320 × 320	416 × 416	608 × 608	960 × 960	1024 × 1024
mAP(%)	55.98	66.73	72.03	85.61	84.17
FAR(%)	8.5	8.7	8.3	6.2	8.9
DR=R(%)	63	70	76	85	83
P(%)	91.4	91.2	91.6	93.7	91.0
F-score	0.745	0.792	0.830	0.891	0.868

Table 3.4: Effect of the resolution on the effectiveness of Yolo V3.

Interpretation

Table 3.4 shows that resizing images affects the effectiveness of the algorithm.

Reducing the image size means that the input of the network will be smaller with some lost data. So the accuracy of the detection will be reduced.

We note that the map value for the resolution of 960 is 30 % bigger than the map of 320 resolution.

The best results that present the highest mAP and DR, with a minimum of FAR is the resolution 960 × 960.

Conclusion

A decision should be made to balance accuracy and speed. In this work object detection with short processing time and good accuracy are required.
The resolution which provides the best effectiveness is (960 × 960) but it takes a large time for processing. Since the computer chosen to process the data is efficient, and compose an accelerator for video processing, the time of execution for this image resolution will be compensated with the accelerator.

The final configuration of the algorithm is chosen as follow:

- Confidence threshold of 0.5.
- IoU threshold of 0.5.

- Image resolution of 960 × 960.

The results of the chosen configuration, using the collected data-set is presented below.

mAP

The figure below show the AP (average precision) values for each category, and the final mAP value. It illustrates how precise is the class detection and the bboxes localisation.

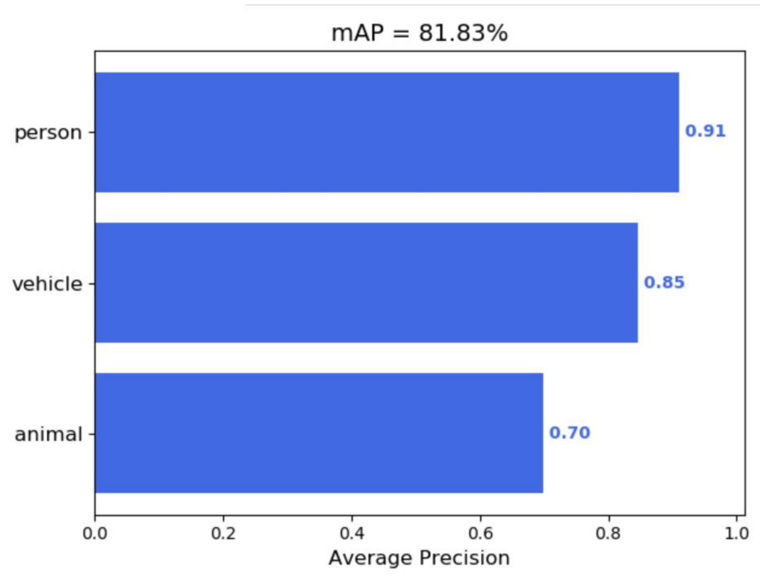
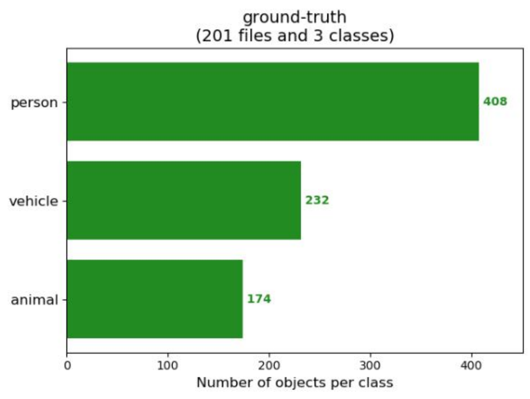


Figure 3.5: Mean Average precision with AP values for each category.

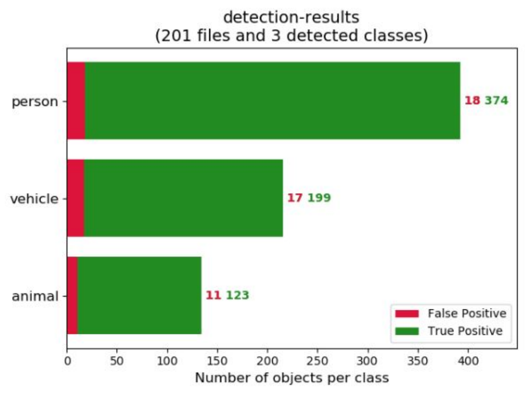
As it shown in Figure 3.5 the AP values for the classes person, vehicle, and animals are 0.91, 0.85, and 0.7 respectively, which gives a mAP of 81.83%.

Ground-truth and detection results

In Figure 3.6, the left image (a) shows the number of objects in each class on the collected data-set, while the right image (b) shows the number of detected objects in each class. TP detections are marked in red. FP detections are marked in green.



(a) Number of ground-truth per category



(b) Detection results for each category

Figure 3.6: Number of ground-truth objects and detection results in each.

Using these results, the DR, FAR, and P are calculated. The detection rate is about 85%, the FAR is about 6%, and the precision value is 95%.

Precision recall curves

The precision-recall curve shows the trade-off between precision and recall for different confidence score threshold. Here, we represent the Precision-Recall curves for each class to show the classifier output quality.
Noting that high recall relates to a low FN, while high precision relates to a low FP.

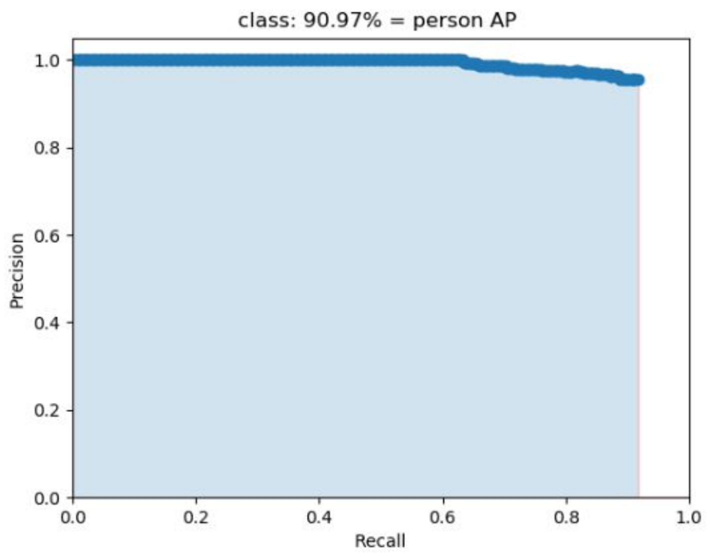


Figure 3.7: Precision-Recall curve for the category:Person.

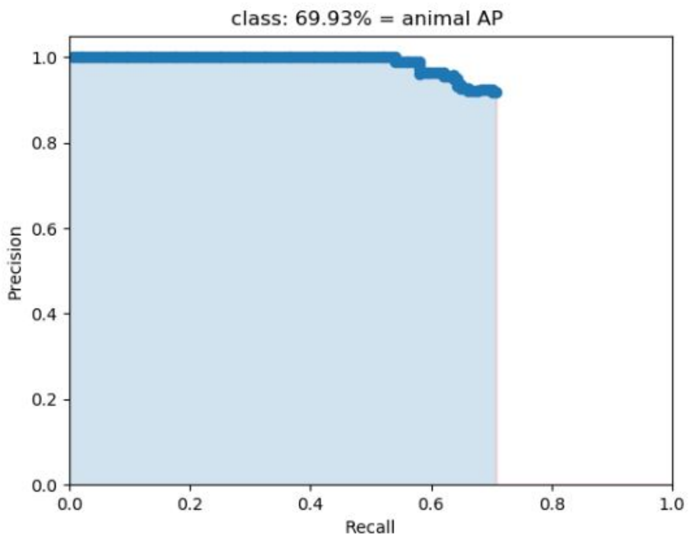


Figure 3.8: Precision-Recall curve for the category:Animal.

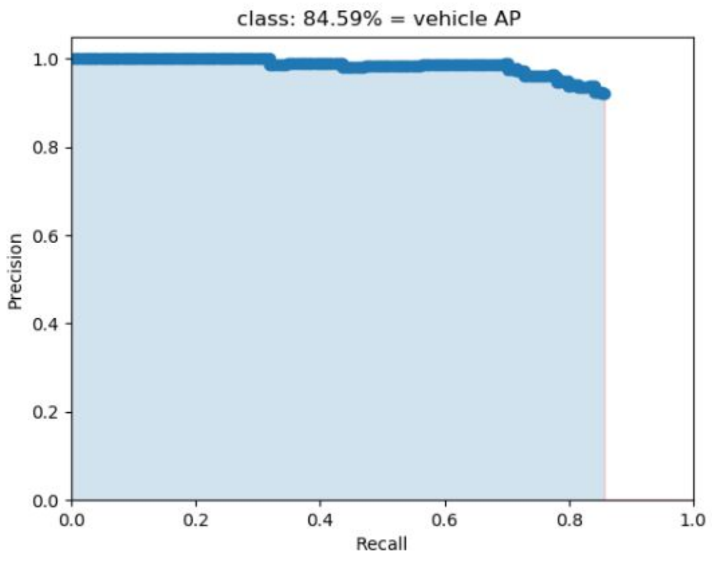


Figure 3.9: Precision-Recall curve for the category:Vehicle.

As shown in Figures 3.7, 3.8, and 3.9, the area under the Precision-Recall curves are high, which means high recall and precision values. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

Noting that high recall relates to a low FN, while high precision relates to a low FP.

Conclusion

The proposed solution to prevent accidents on railways is a machine learning method based on a combination of object detection algorithm "Yolo V3" and the estimation of the object camera distance algorithm "DisNet", with an evaluation for these two algorithms using two different data-sets.

For obstacle detection, three object detection algorithms have been evaluated (Yolo v3, SSD, Faster R-CNN) by calculating the different metrics such as mAP, detection rate, false alarm rate,...for different values of IoU and confidence score threshold. To choose the best of them which, fulfills the requirement of the application. This evaluation could not be done without a data-set. This data-set has been manually collected and labelled with MakeSense tool. It consists of 200 images captured on railroads containing the different obstacles that can be found on railways.

After choosing the Yolo v3 algorithm for object detection, another evaluation has been done for the distance estimation algorithm. For this purpose, another data-set consisting of selecting 200 objects from Kitti data-set with their ground-truth bounding boxes and actual distances measured with Lidar. For distance evaluation, a tolerated error threshold has been chosen to determine the acceptable error.

Obtained results illustrate reliable object detection and distance estimation from a monocular camera in static railway scenes collected from the internet. We have achieved a detection rate of 85% with a false alarm of 6% for the object detection and an error rate for distance estimation of 18%.

Future Works

Since our tests have been realized on static images of obstacles on railways, reproducing results in scenes captured with the two types of cameras mounted on a moving train is a reasonable next step. Creating a data-set for ground-truth distances up to 1000 m is also needed in order to evaluate the algorithm on a long-range.

The movement of the train reduces the quality of the images taken on the railway and makes them blurry. For this, a system can be designed to limit this distortion. It should be installed with the housing of the cameras mounted on the train.

The user interface has been designed to warn the driver. In order to create an autonomous system an improvement should be added by triggering the train stop automatically.

The data used contains common objects, one definite improvement is to add an algorithm based on classical methods such as background subtraction or segmentation could be added to treat the unknown objects present on rail-road that can be dangerous.

To improve this system, an alternative solution could also be added to predict the existence of obstacles on the curvy railroad, since this structure of railways has not been treated in our work.

Bibliography

[1] Armin B.Cremers Shanshan Zhang, Christian Bauckhage. Informed haar-like features improve pedestrian detection. https://www.researchgate.net/publication/263654079_Informed_Haar-Like_Features_Improve_Pedestrian_Detection, 2014. [Conference: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)].

[2] Cordelia Schmid Krystian Mikolajczyk. A performance evaluation of local descriptors. https://www.researchgate.net/publication/7528498_A_Performance_Evaluation_of_Local_Descriptors, 2005. [IEEE Transactions on Pattern Analysis and Machine Intelligence].

[3] DahyotRozenn Dahyot Donghoon KimRozenn. Face components detection using surf descriptors and svms. https://www.researchgate.net/publication/4374607_Face_components_detection_using_SURF_descriptors_and_SVMS, 2008. [Conference: Machine Vision and Image Processing Conference, 2008. IMVIP '08. International].

[4] Tim Cheng K.-T Tim Cheng Shai Avidan Qiang ZhuMei-Chen, YehMei-Chen YehK.-T. Fast human detection using a cascade of histograms of oriented gradients. https://www.researchgate.net/publication/4246340_Fast_Human_Detection_Using_a_Cascade_of_Histograms_of_Oriented_Gradients, 2006. [Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference].

[5] SANJEEVANI K. SHAH SHRUTI S. MANE. Object tracking and analysis for detecting various situations at level crossings of railway-road. <https://sci-hub.tw/10.1109/iccubea.2017.8463973>, 2017. [Third International Conference on Computing, Communication, Control And Automation (ICCUBEA) 2017].

[6] Nilanjan et al Dey. A comparative study between moravec and harris corner detection of noisy images using adaptive wavelet thresholding technique. https://arxiv.org/abs/1209.1558>, 2012. [International Journal of Engineering Research and Applications (IJERA).

[7] Danijela Ristic-Durrant Vlastimir Nikolić Milos Simonovic Milica Ciric Milan S Banic Milan Pavlović, Ivan Ćirić. Advanced thermal camera based system for object detection on rail tracks. <https://www.researchgate.net/publication/330387029_Advanced_thermal_camera_based_system_for_object_detection_on_rail_tracks>, 2018. [Thermal Science 22].

[8] Jorgen Ahlberg-Michael Felsberg Amanda Berg, Kristoffer Ofjall. Detecting rails and obstacles using a train-mounted thermal camera. <https://link.springer.com/content/pdf/10.1007%2F978-3-319-19665-7_42.pdf>, 2015. [Springer International Publishing Switzerland 2015].

[9] Deguchi D. Kawanishi Y.-Ide I. Murase H. Ukai M. Nakasone R Mukojima, H. Moving camera background-subtraction for obstacle detection on railway tracks. <sci-hub.tw/10.1109/icip.2016.7533104>, 2016. [IEEE International Conference on Image Processing (ICIP)].

[10] Neural networks: setting up the architecture. https://cs231n.github.io/neural-networks-1/. [Course Website].

[11] Convolutional neural networks. https://cs231n.github.io/convolutional-networks/. [Course Website].

[12] Serge Belongie Lubomir Bourdev Ross Girshick James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollar Tsung-Yi Lin, Michael Maire. Microsoft coco: Common objects in context. https://arxiv.org/pdf/1405.0312.pdf, 2015.

[13] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2015.

[14] Ross Girshick Shaoqing Ren, Kaiming He and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks . <https://arxiv.org/pdf/1506.01497.pdf>, 2016.

[15] Dumitru Erhan Christian Szegedy Scott Reed Cheng-Yang Fu Alexander C. Berg Wei Liu, Dragomir Anguelov. Ssd: Single shot multibox detector. <https://arxiv.org/pdf/1512.02325.pdf>, 2016.

[16] Mounir Frikha Yehia Massoud Mehdi Masmoudi, Hakim Ghazzai. Object detection learning techniques for autonomous vehicle applications., 2019. [IEEE International Conference on Vehicular Electronics and Safety (ICVES)].

[17] Ross Girshick Ali Farhadi Joseph Redmon, Santosh Divvala. You only look once: Unified, real-time object detection., 2016.

[18] Evaluation metrics for object detection and segmentation: map. <https://kharshit.github.io/blog/2019/09/20/evaluation-metrics-for-object-detection-and-segmentation>, 2019. [website].

[19] Mark Everingham·Luc Van Gool·Christopher K. I. Williams·John Winn·Andrew Zisserman. The pascalvisual object classes (voc) challenge. http://host.robots.ox.ac.uk/pascal/VOC/pubs/everingham10.pdf, 2009. [Int J Comput Vis (2010) 88: 303–338].

[20] Precision and recall. <https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel?fbclid=IwAR0OoPz51wmVUouLoOwvSYXC9JdhRpGjh-0MUyTmT4FptjyAF0XkfBxUJ-k>. [Wikipedia].

[21] Coco evaluation. https://cocodataset.org/#detection-eval. [Course Website].

[22] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty, 2019.

[23] Hao Zeng Yi Zeng Shuaicheng Liu Bing Zeng Qingdong He, Zhengning Wang. Svga-net: Sparse voxel-graph attention networkfor 3d object detection from point clouds. <https://arxiv.org/pdf/2006.04043v1.pdf>, 2020.

[24] George Tzelepis Tiago Cortinhal and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentationof lidar point clouds for autonomous driving. <https://arxiv.org/pdf/2003.03653v2.pdf>, 2020.

[25] Simultaneous localization and mapping "slam". <https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping>. WebSite.

[26] U. Wandinger. Introduction to lidar. [Lidar . pp. 1–18].

[27] J Carter et al. Lidar101: An introduction to lidar technology, data, and applications. [National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center. 2012].

[28] QR. C. Gonzalez and R. E. Swoods, digital image processing, global editions, 2017. Pearson Higher Education.

[29] Point cloud library. <https://pointclouds.org/>. WebSite.

[30] Lidar data (point-cloud) collected from vehicle. <https://jp.reuters.com/article/autos-autonomous-lidar-idJPKBN1QS083>. WebSite.

[31] How lidar technology enables autonomous cars to operate safely - velodyne lidar. <https://velodynelidar.com/newsroom/how-lidar-technology-enables-autonomous-cars-tooperate-safely>. [Velodyne Lidar , 20-Sep-2018. [Online].].

[32] D. Marr and T. Poggio. A computational theory of human stereo vision, 1979. [Proc. R. Soc. Lond. B Biol. Sci. , vol. 204, no. 1156, pp. 301–328].

[33] T. Takeuchi M. Hariyama and M. Kameyama. Reliable stereo matching for highly-safe intelligent vehicles and its vlsi implementation, 2000. [Proceedings of the IEEE Intelligent Vehicles Symposium].

[34] Gabriel J. Brostow Clément Godard, Oisin Mac Aodha. Unsupervised monocular depth estimation with left-right consistency. <https://sci-hub.tw/10.1109/cvpr.2017.699>, 2017. [IEEE Conference on Computer Vision and Pattern Recognition (CVPR).].

[35] Khemmar R. Decoux B. Atahouet A. Ertaud J.Y. Chen, Z. Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility. <https://sci-hub.tw/10.1109/est.2019.8806222>, 2019. [Eighth International Conference on Emerging Security Technologies (EST).].

[36] O. Choi M. Hansard, S. Lee and R. P. Horaud. Time-of-flight cameras: Principles,methods and applications., 2012. [Springer Science & Business Media.].

[37] B. De Brabandere W. Van Gansbeke, D. Neven and L. Van Gool. Rmse: Root mean square error - statistics how to. <https://www.statisticshowto.datasciencecentral.com/rmse/>, 2019. Statistics How To , 25-Oct-2016. [Online]].

[38] Wang B. Song P. Li J. Ye, T. Automatic railway traffic object detection system using feature fusion refine neural network under shunting mode. <https://sci-hub.tw/10.3390/s18061916>, 2018. [Sensors, 18(6), 1916.].

[39] Emami Damon Gräser Axel Nikolić Vlastimir Ćirić Ivan Banić Milan Brindić Branislav Nikolić Dragan Radovanović Dušan Eßer Florian Schindler Christian. Ristić-Durrant Danijela, Haseeb Muhammad Abdul. Smart concept of an integrated multi-sensory on-board system for obstacle recognition. <http://doi.org/10.5281/zenodo.1446119>, 2018. [Transport Research Arena TRA.].

[40] A. Gräser. M. A. Haseeb, D. Ristić-Durrant. Long-range obstacle detection from a monocular camera. <https://wp.mpi-inf.mpg.de/cscs/files/2018/09/06-Long-range-obstacle-detection-from-a-monocular-camera.pdf>, 2018. [CSCS18.].

[41] Thermal imaging: how far can you see with it? https://www.flirmedia.com/MMC/CVS/Tech_Notes/TN_0002_EN.pdf. [FLIR Commercial Systems B.V].

[42] Opencv. <https://opencv.org/>. WebSite.

[43] Pyqt. <https://www.qt.io/?fbclid=IwAR1lWLtVRXfhfzwA6Uowhu13dpFKUmHQ9G5dmLxGpQBOWCoFcm

[44] Piotr Skalski. Make Sense. https://github.com/SkalskiP/make-sense/, 2019.

[45] Labelimg. <https://github.com/tzutalin/labelImg>.

[46] Ankush Gupta Abhishek Dutta and Andrew Zisserman. Vgg image annotator (via). <http://www.robots.ox.ac.uk/~vgg/software/via/>.

[47] Labelme. <http://labelme.csail.mit.edu/guidelines.html>.

[48] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[49] Google colaboratory. <https://colab.research.google.com/#create=true>.

[50] Jimmy Lei Ba Diederik P. Kingma. Adam: A method for stochastic optimization. <https://arxiv.org/pdf/1412.6980.pdf>, 2015. ICLR.

[51] Danijela Ristić-Durrant Axel Gräser Muhammad Abdul Haseeb, Jianyu Guan. Disnet: A novel method for distance estimation from monocular camera. <https://project.inria.fr/ppniv18/files/2018/10/paper22.pdf>, 2018. [CSCS18.].

[52] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[53] Keras. <https://keras.io/>. WebSite.

[54] Tensorflow. <https://www.tensorflow.org/?hl=fr>. WebSite.

[55] Athira S. Image processing based real time obstacle detection and alert system for trains. <https://sci-hub.tw/10.1109/ICECA.2019.882181>, 2019. [Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019].

[56] Deguchi D. Kawanishi Y. Ide-I. Murase H. Ukai M. Nakasone R Mukojima, H. Moving camera background-subtraction for obstacle detection on railway tracks. <sci-hub.tw/10.1109/icip.2016.7533104>, 2016. [IEEE International Conference on Image Processing (ICIP)].

[57] Ryan Nash Keiron O'Shea1. An introduction to convolutional neural networks. <https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks>, 2015.

[58] John Winn Mark Everingham. The pascal visual object classes challenge 2012 (voc2012) development kit. https://pjreddie.com/media/files/VOC2012_doc.pdf, 2012.

[59] Dibyendu Ghoshal Pinaki Pratim Acharjya, Ritaban Das. Study and comparison of different edge detectors for image segmentation. https://core.ac.uk/reader/231151822, 2012. [Global Journal of Computer Science and Technology (GJCST)].