**PRESENTATION DES ETUDES :**

Learning Object-Specific Distance from a Monocular Image - 2019

Jing Zhu, Yi Fang, Husam Abu-Haimed, Kuo-Chin Lien,Dongdong Fu,Junli Gu

**Introduction:**

Most of the current robotic self-driving systems employ inverse perspective mapping which consist in projecting object into a bird eye view (in order to get an image with no perspective issue).

After seeing that IPM method performed poorly for objects far away (fail over 40 meters or curved road) or on the side of the camera, researchers decided to develop an algorithm able to predict distance of obstacles in images (based on NN)

In this paper they also developed an enhanced version of their own algorithm thanks to the addition of a keypoint regressor (Used to predict 3D coordinate of the obstacle)

**Hypotheses:**

* There is no efficient and accurate object-specific distance estimation deep learning algorithm…
* … especially for autonomous driving
* Their application can outperform Inverse Perspective Mapping

**Method:**

Extract feature from the image (vgg16 and res50 are tested as feature extractor algorithm)

Use ROI pooling to create fixed-size vector for all the feature of the image.

Feed the ROI features into a distance regressor and a classifier.

Use keypoint regressor to get X and Y coordinate of the object and combine them with the distance to get the full 3D coordinates. (hence a better precision)

Use KITTI object detection and nuScenes dataset for validation and depth prediction as the evaluation metrics

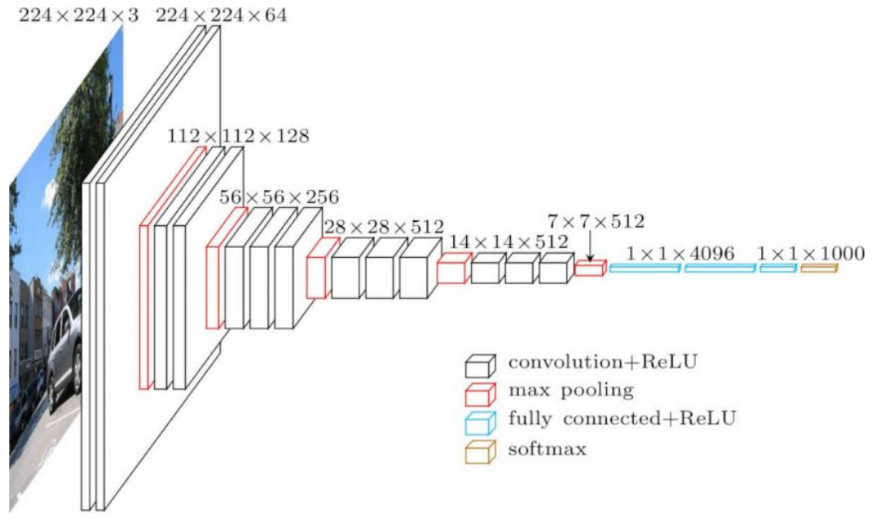
**Feature Extractor (VGG16) :**

VGG16 is a convolutional neural network model for Image Recognition. It has a 92,7% accuracy rate (for 1000 classes using ImageNet dataset).

The VGG16 consists in a pattern made up of multiple 3x3 convolutional layer followed by a maxpool layer, this pattern is reproduced 5 times. It is then followed by 3 fully connected layer.

The first layer have a fixed size of 224x224 and this number is divided by 2 at each max pooling layer, which means that it has a size of 7x7 after the last max pooling layer.

On the contrary the number of filter is doubled after each pattern, it begin with a depth of 64 at first and finish with a depth of 512.



Since the depth of this algorithm is large it takes a lot of time to train it and takes a lot of space.

**There is lighter algorithm like SqueezeNet or GoogleNet.**

**Results :**

For the KITTI dataset :

We can see that on average the enhanced model using vgg16 as the feature extractor is the best whereas the one with the worst result is the SVR method.

We can also note that the IPM method has similar result to their model (for the threshold)

**For the car :**

The method having the best results on average is the base model (vgg16) but the enhanced method (vgg16) has very similar result.

All the model of the study has really good result whereas de SVR and IPM have really inconsistent results.

For example the RMSE is around 3.5 for the 4 model of the study but the IPM has a RMSE of 237 and the SVR has 19.

For the threshold the IPM method has similar result to the study model.

**For the pedestrian:**

The method having the best results on average is the Enhanced model (vgg16) and with the pedestrians we can see a clear difference between both the enhanced model and the base model.

The SVR method has really bad result but the IPM has result similar to the base model.

**On average:**

The enhanced model has best results on average than the base model but it depends on the obstacle. The enhanced model is better with pedestrian and cyclist but has similar result for car.

It is also important to note that their model has been trained with part of the KITTI and nuScenes dataset.

DisNet: A novel method for distance estimation from monocular camera – 2018

Muhammad AbdulHaseeb, Jianyu Guan, Danijela Ristić-Durrant, Axel Gräser, Member, IEEE

**Introduction :**

Novel method for object distance estimation from a single image with no prior knowledge of the camera parameters or the scene.

**Hypothesis :**

* Reliable and accurate object detection is still a problem nowadays.

**Method :**

Utilisation de YOLO (base de données COCO)

YOLO 🡪 calcul de la boite d’ancrage (bounding box) des objets

* En fonction de la géométrie on classifie l’objet et on détermine sa distance

DisNet 🡪 Calcul distance par rapport à la caméra (relation inverse entre dimension et distance)

**Références :**

[11]: Stereo vision method for collision warning

[3]: Advanced driver assistance system using computer vision and laser scan

[1]: Obstacle detection technique based on stereo vision (using monocular cues)

[13]: 3-D Depth Reconstruction from a single still image

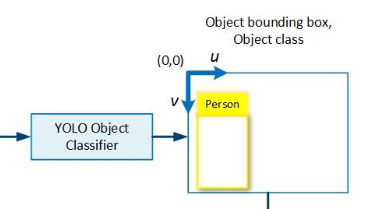
[16]: Stochastic optimization method Adam (appropriate for non-stationary objectives and problems with very noisy and sparse gradient)

**Technique:**

We first train the YOLO Object Classifier with COCO dataset.

The camera image is sent as an input to the YOLO object detector.

This same Object classifier gives the bounding box of each detected object in the images as an output and label each one of them.



For each extracted bouding-box, they calculate a six-dimensional feature vector like this:



Where :

Bh = height of the object bounding box in pixels/image height in pixels)

Bw = width of the object bounding box in pixels/image width in pixels

Bd diagonal of the object bounding box in pixels/image diagonal in pixels

Ch = Average height

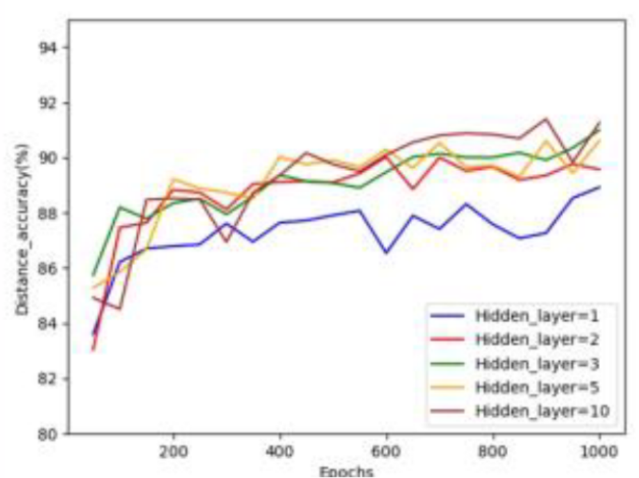
Cw = Average width

Cd = Average diagonal

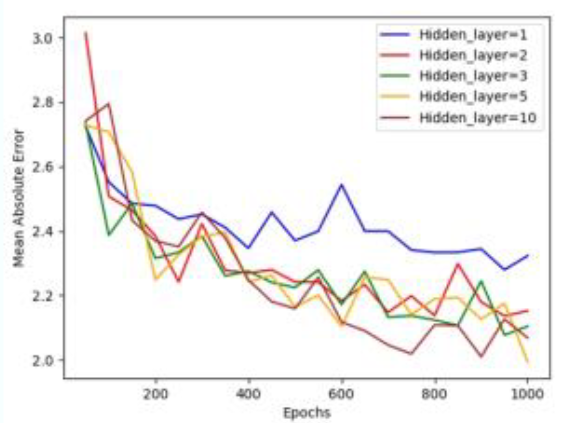
These bounding are then processed by the trained DisNet Multi Hidden-layer Neural Network.

**Disnet Structure :**

The DisNet algorithm has been tested with 1,2,3 ,5 and 10 Hidden Layer at first, after thorough test it has been shown that the 3 hidden layer structure was the best one because it has approximatively the same Mean Absolute Error value as the 5 and 10 hidden layers.



**Figure : Distance estimation accuracy achieved for different number of hidden layer**



**Figure : Mean Absolute error achieved for different number of hidden layer**

We can see that only one hidden layer is not nearly enough because it has a bad mean absolute error and a bad accuracy rate.

We can also see that 10 Hidden layer has the best results in both test but a trade off has been made between computational time and the final result.

And then to decide the number of hidden neurons in each layer they also tested the DisNet over 1000 epoch for 10, 30, 100 and 200 neurons.

Input (6 entrées)

Output (1 sortie)

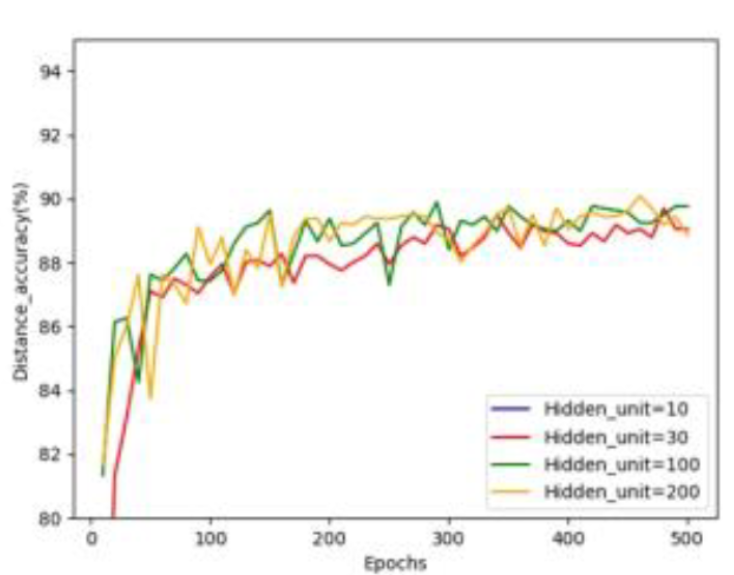
Hidden Layer (100 neurones)

Hidden Layer (100 neurones)

Hidden Layer (100 neurons)

**Input**: 6-dimensional feature vector

**Output**: Distance



**Figure : Distance estimation accuracy achieved for different number of hidden neurons**

In this graph we can see that there is a clear difference between 30 hidden neurons and 100/200 neurons, but not much of a difference can be seen between 100 and 200 neurons, so 100 hidden neurons have been chosen.

**Training :**

There is a relationship between the calculated dimension of the bounding boxes and the distance of the obstacle. Geometrically, by the projective transformations, the object bounding box size is expected to get smaller the further away the object is, so the inverse of the bounding box size is expected to increase as the distance increase.

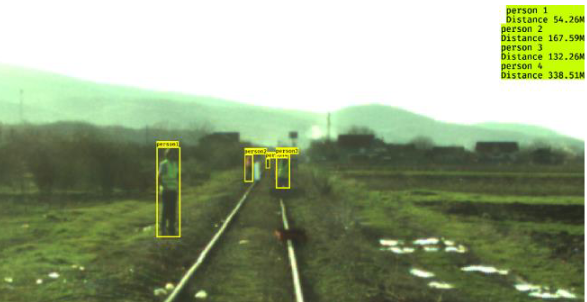
For training the network the input dataset was firstly randomly split into a training (80% of the data), validation (10% of the data) and test set (10% of the data). The DisNet was trained using the backpropagation method with the **Adam optimizer** backpropagation method on the dataset collected

But the algorithm was never trained with real railway track scene.

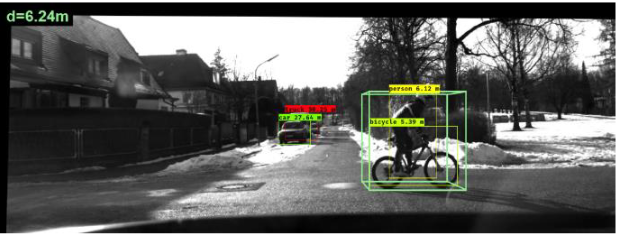
**Evaluation de la technique:**

The DisNet method is made for Railway situation so to evaluate the network they used straight rail tracks images of day and night location. They used a laser for the ground truth.

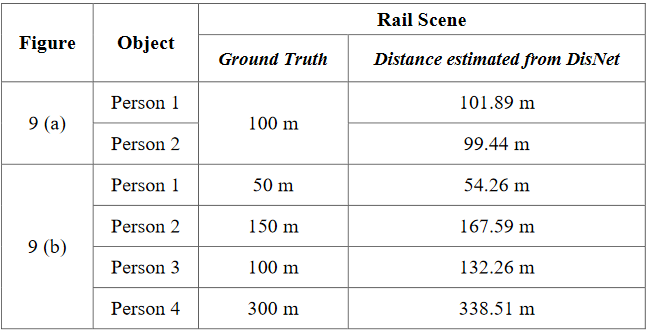
(9.a)

 (9.B)

The algorithm was then tested with road scene against a HiSpe3D-Vision which is a collision warning system for automobile.



**Résultats de la technique :**

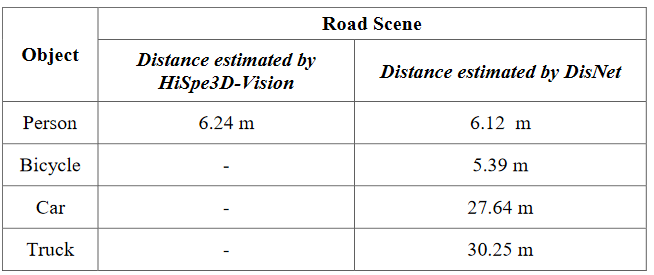


**Figure: Test on rail scene (*9a and 9b*)**

We can see with those first results that the Disnet is very accurate when the distance is less than 100m and there are just a few obstacles. Whereas in a situation with a lot of people we can see that the accuracy of the Disnet is really bad.

It is also due to the fact that the DisNet algorithm was not trained with real rail track scene which can lead to those inaccuracy.

It was then tested on road scene:



**Figure: Test on road scene**

The Disnet Algorithm gives actual result and similar result to HiSpe3D for the person but since we don’t have ground truth we can’t say for sure if DisNet works accurately.

**Conclusion :**

The DisNet algorithm recognized a lot more of obstacle but there is still a problem with the bounding boxes extraction

DisNet is really interesting because firstly it talks about the YOLO algorithm which is easy to implement and shows that it has a really good success rate, secondly the idea of using the bounding box dimension for distance estimation and having usable result can be useful for our project.

The training set used in this algorithm is really important because the results will vary a lot, for a usage in amateur video this can be a problem.

DeepCalib: A Deep learning approach for automatic intrinsic calibration of wide field-of-view cameras – 2018

***Oleksandr Bogdan, Viktor Eckstein, François Rameau, Jean-Charles Bazin***

Lexique :

Unified spherical model

# Introduction :

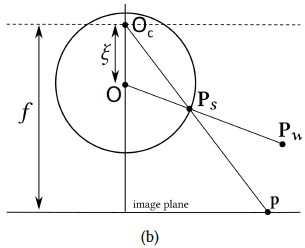
Calibration of wide field-of-view cameras is a fundamental step for numerous visual media production applications, such as 3Dreconstruction, image undistortion, augmented reality and camera motion estimation. However, existing calibration methods require multiple images of a calibration pattern, assume the presence of lines, require manual interaction and/or need an image sequence

# Objectif :

They present a novel fully automatic deep learning-based approach that works with a single image of general scenes. It performs explicit camera calibration and runs automatically, and thus enables several applications, such as image undistortion and SfM (Structure from Motion).

# Method :

Generate own dataset of images with different distortion and focal length using 360° images. For this they are using unified spherical model (stereographic projection).



| **PW** : 3D world point | **PS** : point projected on the sphere | **ξ** : camera distortion | **p** : point on the image plane |

We first project a point Pw from the 3D world onto a sphere (Ps) by tracing a line going from PW to the center of the sphere. This point is then projected on the image plane by tracing a line passing through OC (distorted center) and PS.

