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

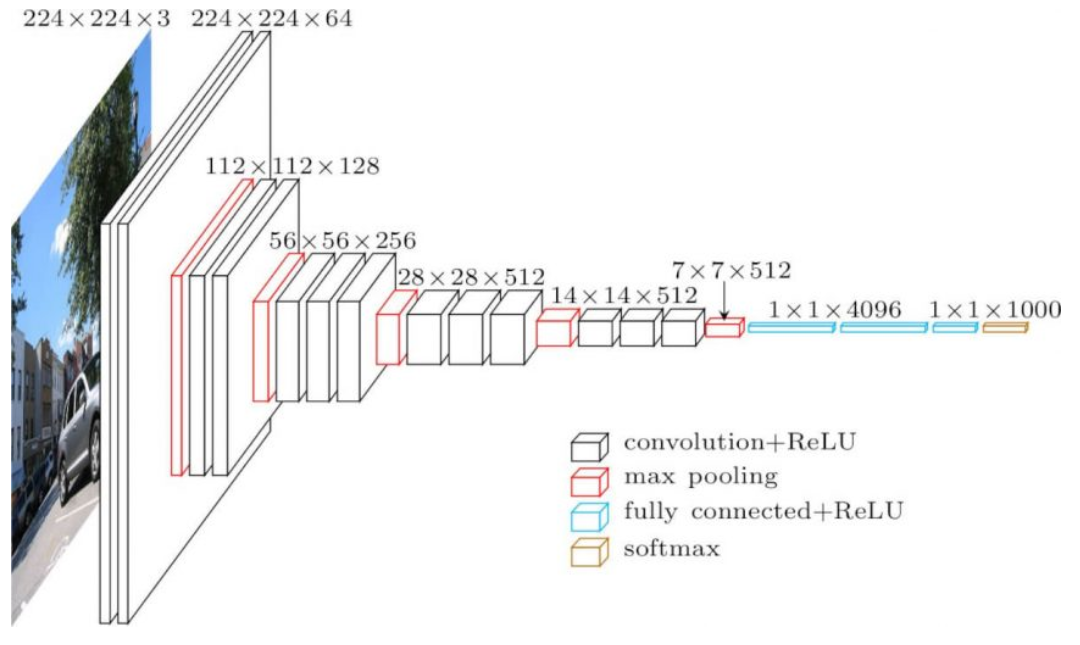
**Feature Extractor (VGG16) :**

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.**

**References:**

**Distance estimation:**

Inverse perspective mapping algorithm [28 , 25]

DisNet algorithm (CNN Based model) [13,14]

Distance detection algorithm for images in deflecting angle (Yifei Feng 2016) (use rectangular pattern to estimate distance but need those patterns to work effectively)

**2D Perception:**

R-CNN boost accuracy and decrease time processing [12,11,24,15]

SSD (Single shot multibox detector)

YOLO algorithm

Depth map prediction [7,9,18,19]

**Technique :**

1. **Feature extractor:**

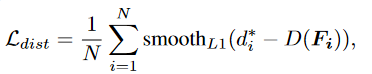
Use of vgg16 (2048, 512, 1) and res50 (1024,512,1)

1. **Distance regressor:**

3 FC Layer (with a softplus activation function on the last layer) and one output (the distance Z)

It has a softplus function activation on the output of the last layer to make sure it’s a positive distance.

Loss function:



N = Number of objects

di = Ground truth distance

D(Fi) = Predicted distance

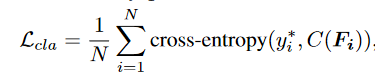
x = di- D(Fi)

**SmoothL1** = if |x| is higher than a given parameters then it’s equal to |x| else it is equal to |x|²

1. **Classifier:**

One Fully connected layer (with a number of neuron equal to the number of categories in the dataset) and a SoftMax function (probabilities for each category to be true)

**Loss function:**



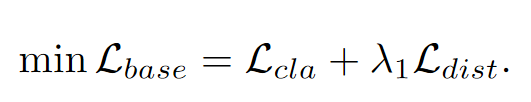
Yi = category label

C(Fi) = Output of the classifier

## Cross-entropy = *L* = Σ−yi\*log(C(Fi))

1. **Learning and training:**

The feature extractor, the distance regressor and the classifier are simultaneously trained with loss.



They also use Adam optimizer (same as DisNet) to obtain the optimal parameters network. (β = 0.5)

The learning rate is initialized as 0.001 and exponentially decayed after 10 epochs. Λ is set to 1.0 when training their framework.

One of the problems with object-specific distance estimation task is the lack of dataset with ground truth. So, in this study they constructed ground truth dataset with KITTI and nuScenes dataset (depth evaluation dataset) and used the depth information to link their images coordinates with the distance.

They split KITTI and nuScenes in half for validation and training. (8 classes)

**Evaluation de la technique:**

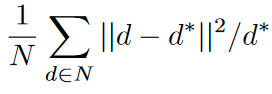
They compare their algorithm with IPM from matlab toolkit (classic method for distance estimation for automobile) and SVR algorithm with their base model and their enhanced model.

Threshold

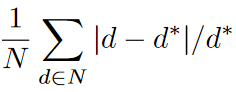
The computed error are (check **[7]**):

RMSE (linear and log): Tells you how close you are from fitting the data (lower is better)

Squared relative difference:



absolute relative difference:



**Résultats de la technique :**

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.

**Limites :**

- Manque de Dataset

-dépend énormément du dataset

-encore beaucoup d’erreur

**Lexique :**

Bouding box (Boite d’ancrâge) : image location

Inverse perspective mapping

End-to-end learning

Bird-eye view

Keypoint regressor

Region of interest pooling

Kitty and nuScenes dataset

Softplus activation function

Batch and Epoch

Adam optimization algorithm

SVR