# Spatial Detection of Vehicles in Images using Convolutional Neural Networks and Stereo Matching

Jeremy Pinto, University of Waterloo, ON, Canada
Nolan Lunscher, University of Waterloo, ON, Canada
Georges Younes, University of Waterloo, ON, Canada
David Abou Chacra, University of Waterloo, ON, Canada
Henry Leopold, University of Waterloo, ON, Canada
John Zelek, University of Waterloo, ON, Canada

## Abstract

Convolutional Neural Networks combined with a state of the art stereo-matching method are used to find and estimate the 3D position of vehicles in pairs of stereo images. Pixel positions of vehicles are first estimated separately in pairs of stereo images using a Convolutional Neural Network for regression. These coordinates are then combined with a state-of-art stereo-matching method to determine the depth, and thus the 3D location, of the vehicles. We show in this paper that cars can be detected with a combined accuracy of approximately 90% with a tolerated radius error of 5%, and a Mean Absolute Error of 5.25m on depth estimation for cars up to 50m away.

## 1 Introduction

Image recognition is of central importance in autonomous driving. When designing a system that can navigate vehicles, a machine must analyze a scene and identify what surrounding objects are present and where they are located. This information can then be used to adjust driving commands accordingly [6]. Chenyi *et al* [1] have proposed a novel paradigm, *Direct Perception Approach*, in which a Convolutional Neural Network (ConvNet) is used on images to detect key affordance indicators necessary for driving, such as the closest cars in the vicinity of a host car, lane markings, and vehicle angles. They demonstrate in [1] that using this paradigm, a virtual vehicle can be driven relatively smoothly.

Most of the work presented in Chenyi *et al* [1] consists in training a ConvNet based on videogame data. A section of their work focuses on the KITTI Dataset, a publicly available dataset, consisting of footage from a camera mounted on a vehicle driving around European cities and complemented with ground truth of positions of objects surrounding the host vehicle [2]. Their results were promising and their methods have inspired the work presented in this paper. Particularly, we use a convolutional neural network in order to locate the position of vehicles in stereo pairs of images, and use those predicted positions as a starting point for performing stereo matching using state-of-the-art stereo-matching methods [7], [8]. Using ConvNets to determine starting points for stereo matching has previously been proposed by Zbontar and Lecun [8], in which they use ConvNets to evaluate degrees of similarity between stereo images before using a stereo-matching pipeline. This paper seeks to gap the bridge between Chenyi *et al*'s Direct Perception approach [1] and Zbontar and Lecun's stereo-matching methods [8], combining ideas of both papers into one comprehensive pipeline.

In the context of this paper, only objects labelled as *car* or *van* are considered from the KITTI Dataset. Others, such as *pedestrian*, *cyclist* are ignored. Each labelled object is accompanied by its tracklet information, which consists of information related to its spatial position and degree of occlusion, and by coordinates for 2D and 3D bounding boxes. The dataset used in this project consists of approximately 6000 stereo pairs of images, of which only one side (the left side) is labelled.

In this paper, a slightly modified version of the ConvNet known as AlexNet, will be used [5]. It consists of 5 convolutional layers, followed by 3 fully connected layers. An L2-norm *Euclidean Loss* function is used to evaluate the output loss from the network in a process known as regression. *Euclidean Loss* is defined as the average distance squared between the output vector and the ground truth vector as shown in equation (1). The *Caffe* [4] framework was used to train our ConvNet and the original AlexNet parameters were used for training.

$$E = \frac{1}{2N}\sum_{n=1}^{N} ||\hat{y}_n - y_n||^2 \quad (1)$$

## 2 Background

### 2.1 DeepDriving

In DeepDriving, Chenyi *et al* [1] introduce the concept of *Direct Perception Approach* in detail. A section of their work focuses on analyzing images from the KITTI dataset. A ConvNet is trained to look for the 3 closest vehicles to the host vehicle, by learning their (*x,z*) coordinates as defined in Figure 1. To do so, labelled images from the KITTI dataset are used for training their ConvNet from scratch and regression is used to estimate the vehicle coordinates as outputs.

### 2.2 Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches

Zbontar and Lecun have recently shown that ConvNets can be used to determine the matching cost between images, which is typically the first stage in many stereo matching algorithms [8]. It consists in quantifying the degree of similarity between image patches in order to determine which patches to compare for depth estimation. To do so, they train a ConvNet to determine a similarity score. Disparity and depth maps are then calculated using state-of-the-art stereo matching methods as shown by Mei *et al* on rectified stereo image pairs [7]. Their methods have proven to be successful, particularly on images from the KITTI dataset.

## 3 Methodology

The first step of detection involves training a ConvNet to estimate the (*x,y*) coordinates of vehicles as shown in Figure 1, or in other words, an approximation for the pixel location of the cars present the original KITTI images. The rationale in estimating (*x,y*) instead of (*x,z*) as presented by Chenyi *et al* [1] is that stereo information is available and using stereo matching on properly identified vehicles to determine the depth could benefit from higher accuracy, as a shift of few pixels could affect the output in the former case. Calculating depth through stereo matching also allows to have all (*x,y,z*) coordinates of vehicles in images, as opposed to DeepDriving which looks only for (*x,z*) coordinates.

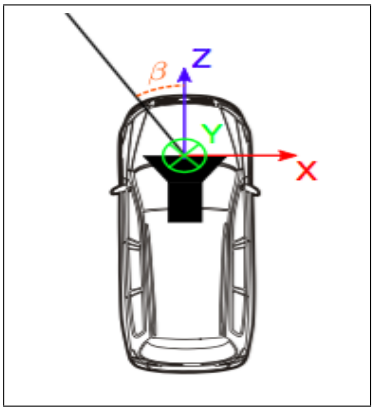


*Fig. 1: x,y and z coordinates in the reference frame of a host car in the KITTI dataset [2]*

Each KITTI image is segmented in to 3 distinct images, as shown by the blue lines in Figure 2 and in each segment the (*x,y*) coordinates of 3 vehicles are estimated. Thus, it could be possible, in a given KITTI image, to find up to 9 vehicles using our method. Also, by segmenting the images into 3 distinct images, they can be

rescaled to approximate the input image sizes used by AlexNet in ILSVRC 2012, making fine-tuning a lot more appropriate. Since images are being cropped, it is possible for a vehicle to only be slightly present in an image. We determined empirically that ground truth for the vehicle should be kept only if at least 20% of the bounding box of the vehicle was contained within a crop. Finally, seeing as ConvNets are very data intensive, splitting the images into 3 separate images results in 3 times more training data. Mirroring each image also allowed to double the dataset, resulting in just over 48 000 labelled images. About 10% of the dataset was used for testing purposes, and the rest for training purposes. Dropout was also used through training to avoid overfitting. Once the 2D position of vehicles are estimated, they are combined with methods presented by Zbontar and Lecun [8] to estimate the depth of each vehicle.

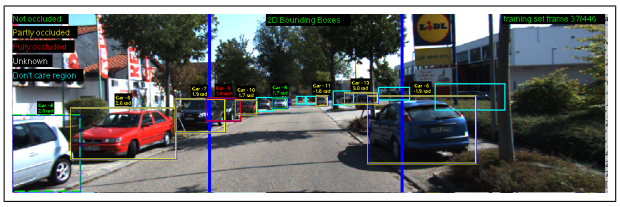


*Fig. 2:* Example of an image from the KITTI dataset with labels. The blue lines represent the segmentation boundaries used for training.

One limitation of ConvNets is that they must always output a fixed-sized vector of information. Since the images used to train the network don't always contain 3 vehicles, a special workaround is used. When cars are not present in an image, the ground truth is set to being the top-centre pixel of the image, since it is very rare for cars to be present there. If they ever were (in the case of an overhead bridge, for example), they would not be very important in the context of vehicle awareness. Figure 5 shows examples of positions being estimated in the top centre to indicate that no vehicles were found. A single, centred point at the top of the image was chosen to avoid noise or bias towards a given side.

## 4 Results

Fine-tuning was performed with the weights from the AlexNet structure used in the ILSVRC 2012 classification problem [5]. One reason these weights were chosen was because many cars are contained within that dataset and thus the appropriate car filters could be activated upon learning on a relatively small training set.

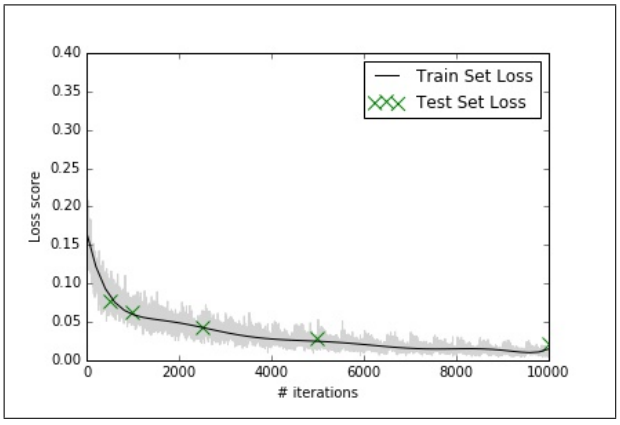


*Fig. 3:* Train and test loss as a function of iterations. Decay in both sets indicate convergence towards an appropriate solution without overfit.

Upon training, convergence towards an acceptable model was observed. Indeed, the loss function seemed to be decreasing and converging towards a steady value, as opposed to the erratic behaviour observed when training from scratch. The loss function of the training and testing sets as a function of iterations is shown in Figure 3.

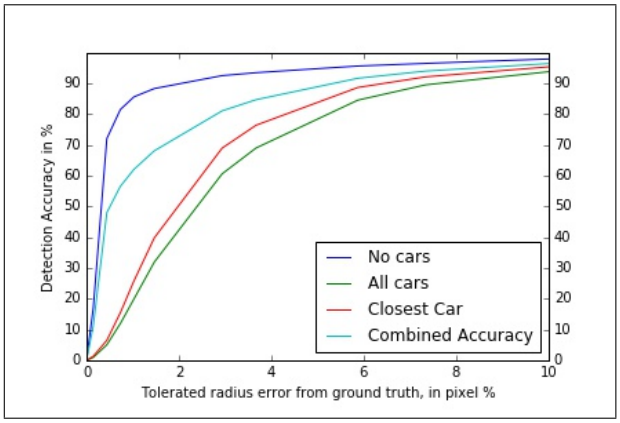


*Fig. 4:* Detection Accuracy for different scenarios, as a function of tolerated radius error. When no cars are present, the system must predict a position determined as the "no car" position.

Figure 4 gives a more intuitive look at the results for the testing set (for which ground truth is provided). The detection accuracy is measured, in the case of *no cars*, as the ratio of proper detection of no cars present over the total instances of no cars predicted by the ground truth, and in the case of *closest car* (or *all cars*), as the proper detection of the closest car in an image compared to all closest cars in images (or all cars in the image in the case of all cars). The combined accuracy is the combination of proper detection of no cars, and all cars, over all car positions predicted by the ground truth. *Tolerated radius error* is defined as the radius of a circle, centred about the ConvNet prediction, needed to overlap the ground truth position coordinate. Figure 4 demonstrates that as we increase the *Tolerated radius error*, detection accuracy increases. The combined accuracy, within a tolerated radius error of 5%, yields a combined accuracy of approximately 90%.

The next step is to determine the depth of vehicles, using state-of-the-art stereo matching algorithms. In the context of autonomous driving, the 2D pixel coordinate of a car in a given image is not particularly useful, as depth is missing. However, the idea is that for a given ConvNet, the predicted pixel position of a vehicle should be similar in a pair of stereo images. These positions could be used as a similarity measure to find regions of interest to compute depth from disparity of vehicles. Thus, the output of this neural network is used as the starting point to estimate the depth of the vehicles. This allows for the estimation of all $(x,y,z)$ coordinates of vehicles. Figure 5 shows two images, right and left of the stereo pair, that were both not part of the training set. In each, the cross symbolizes the ground truth positions from the KITTI dataset, the dot represents the ConvNet estimation, and the circle shows a radius error of 2.5% relative to the entire KITTI image. When no car is present, ground truth is set to be the top centre of the image. We see that the predictions lie very close of each other, and serve as a good starting point for stereo matching. We see as well in Figure 5 a disparity map (computed in this case for the entire image) and how this information can be used to determine the depth of vehicles.

In terms of precision, it was noticed that our network better estimated depth for vehicles that were closer to the host car, as opposed to vehicles that were further away. When looking at vehicles properly detected in the testing set which were at most 25m away from the host vehicle, depth was approximated with a Mean Absolute Error of approximately 3.97m, as shown in Table 1, and of 5.25m for vehicles up to 50m away. This compares to the reported values of 5.83m presented by DeepDriving for vehicles up to 50m away. However, this is not a direct comparison, since their results were not reproduced and tested on the exact same conditions at the time of publishing this paper, but are rather metrics to give the reader an appreciation of the practicality of this application, and how it can generally compare to other state-of-art methods.

There are many directions to focus future work on. The dataset can still be augmented by a few factors, mainly by cropping more within images and by using the stereo information and camera calibration to effectively double the training set. This could potentially lead to a much more accurate prediction for vehicles. We noticed that predictions of images from the right side of the stereo pairs were less precise. This could be explained by the fact that training

| Method | 50 m | 25 m |
|---|---|---|
| DeepDriving | 5.83m | - |
| CNN + StrMat | 5.25m | 3.97m |

*Table 1:* Results for our test set, compared to those reported by DeepDriving. These results were not tested on the same sets, and serve only as a means of comparison

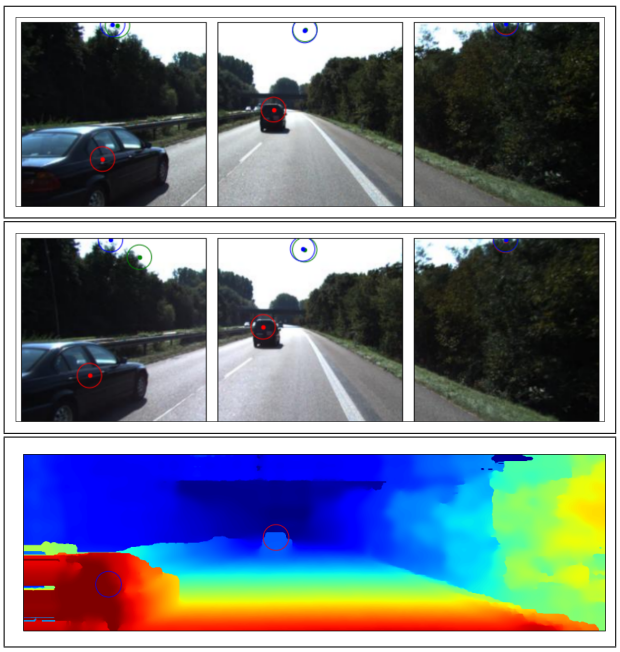

*Fig. 5:* Two stereo images from the KITTI dataset that were not part of the training set. No ground truth is provided for these images, but the network does a seemingly good job at finding the vehicles in the image, and the 2D positions are used to determine depth by stereo-matching (bottom image).

images only came from the left side of the dataset.

Another path to explore would be the use of smaller ConvNets, like SqueezeNet, which boasts similar accuracy as AlexNet using a network with 50 times fewer parameters [3]. This could make computational times faster, and network deployment more efficient.

Methods in DeepDriving also suggest training two separate networks, one with the original KITTI images as input and another with a zoomed in KITTI image as input, in order to achieve higher precision when looking a vehicles further away. This method should be explored further, adapted to the methods presented in this paper, in order to improve accuracy. It would also be a good idea to use transfer learning on similar datasets to see how robust this method truly is.

Finally, benchmarking against other methods is a natural next step for this paper. It would be necessary to reproduce and quantify results of other methods using similar metrics and testing conditions and compare precision, accuracy, and complexity of each.

### 4.1 Conclusion

We show in this paper how a ConvNet can be used for regression to estimate the pixel positions of vehicles in pairs of stereo images. These coordinates are then used as starting points for depth estimation of identified vehicles using state-of-the-art stereo matching methods. Thus, 3D positions for vehicles in images can be found using this method. We show that we can detect vehicles with approximately 90% detection accuracy given a tolerated radius error of 5%. Based on our test set, we show that depth can be estimated with a Mean Absolute Error of 5.25m on cars up to 50m away. This system still has a lot of room for improvement, but shows how ideas from multiple papers can be combined to offer alternative approaches to depth estimation of vehicles in pairs of stereo images.

## References


[1] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[3] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 1mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 163–168. IEEE, 2011.

[7] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 467–474. IEEE, 2011.

[8] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.