Grenze International Journal of Engineering and Technology, June Issue

Annotation Tool and Urban Dataset for 3D Point Cloud Semantic Segmentation

SureshKumar M[1], Abisha S[2], Tejashree D[3] and Vasanthan P[4]
[1-4]Sri SaiRam Engineering College/Information Technology, Chennai, India

***Abstract*—Deep learning requires a significant amount of annotated training data in order to accurately separate unstructured 3D point clouds using semantics. Nevertheless, there isn't any free specialist software on the market right now that can effectively annotate big 3D point clouds. By providing PC-Annotate, a free annotation tool for 3D point cloud research, we close this gap. The suggested solution provides essential functions of point cloud registration and the creation of volumetric samples that may be readily understood by modern deep learning point cloud models, in addition to enabling systematic annotation with a variety of essential volumetric shapes. A sizable outdoor urban dataset for 3D semantic segmentation is also introduced by us. The Ouster LiDAR was used to capture the required dataset, PC-Urban, which was then PC-Annotate labeled. It has 25 annotated classes, 66K frames, and more than 4.3 billion points. Finally, we present PC-Urban baseline semantic segmentation outcomes for well-liked contemporary techniques.**

***Index Terms*— Annotation tool, Faster R-CNN, AlexNet Architecture, Depth Estimation.**

I. INTRODUCTION

A long-running debate has focused on the use of machine learning and computer vision techniques to scene classification and object localization, notably for automated driving. There have been certain advancements in deep learning architectures in addition to the hardware needed for the same, as the automotive industry is now developing advanced driver assistance systems (ADAS) and autonomous driving at a rapid pace. In this thesis, a neural network is created for use in the real-time object localization and detection in a driving context. The CNN architecture, which was created specifically for object detection and classification in visual input, is first described. We tackle the recognition time and training data availability issues by using a high-capacity Convolutional Networks (CNN) to generate region recommendations in order to localise and split objects. In our challenge, we employ a monitored pre-trained CNN (AlexNet) that was created especially for such an auxiliary task, followed by domain-specific fine tuning, i.e. for target recognition and identification in a driving environment. This contributes to greatly improving the generated CNN's results. We begin to develop our R-CNN for depth estimation after evaluating which R-CNN framework is optimal for real-time object recognition and localization in an image. We begin by briefly describing the methodology of this research to perceive depth information and then continue to discuss why stereo vision is a credible and consistent method for achieving this goal.

II. INTRODUCTION TO NEURAL NETWORKS

This section will cover CNN Architecture, a few well-known CNN designs, and their specifics. We suggest a

Grenze ID: 01.GIJET.9.2.530
© Grenze Scientific Society, 2023

technique for object detection that makes use of deep convolutional networks, particularly for detecting typical things found on roads (Car Pedestrians, Traffic Signs etc.). We use a straightforward method by applying the rich feature hierarchies of CNN learned from a sizable image dataset to a particular position in this example, items on a road.

A. Architecture

Convolutional Neural Networks, figure 1, benefit from the fact that they were created with picture inputs in mind. The neurons of CNN are grouped in three dimensions, namely width, height, and depth, which limits their layout in a more logical manner.

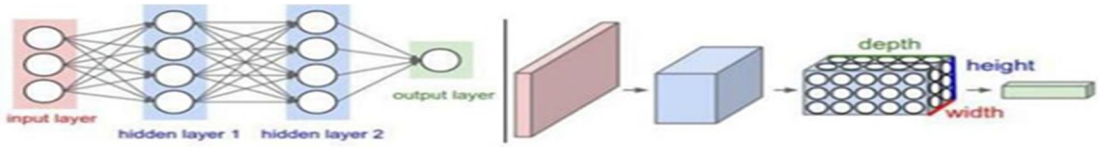

Figure 1. Regular Layers of Neural Network Vs Layers of Convolutional Neural Network

The Convolution layer, which performs much of the computational labour, is the foundational building component of a Convolutional Neural Network. A group of learnable filters make up the CONV layer's parameters. Each filter is miniscule in terms of width and height but encompasses the entire depth of the input volume. On higher levels of the network, the network will eventually learn whole honeycomb or wheel-like patterns. The depth, stride, and zero-padding are the three hyper-parameters that regulate the size of the output volume. The convolution layer can be summarized as: Accepts a volume of size $W1 \times H1 \times D1$. Requires four hyper-parameters: Number of filters K, Spatial extent F, Stride S, Amount of zero padding P. Produces a volume of size $W2 \times H2 \times D2$ where: $W2 = (W1 - F + 2P)/S + 1$,$2 = (H1 - F + 2P)/S + 1$ (i.e., width and height are computed equally by symmetry) ,$D2$ = K.

II. Region based methods for object detection and identification

We will discuss region-based approaches for object detection in this chapter. Specifically, R-CNN (Regional CNN), the first CNN to be used to this problem, as well as Fast R-CNN and Faster R-CNN. By identifying the single object that is the focus of the image, classification aims to identify the image. The work we do when we observe the environment, however, is significantly more challenging. When we observe complex scenes with several overlapping objects and varied backdrops, we classify the various objects as well as recognize their limits, distinctions, and relationships to one another. A region-based CNN's objective is to detect and locate an object from an image input.

A. R- CNN

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik originally proposed R-CNN in 2014. Using two concepts, the region-based convolutional network (R-CNN): (1) In order to localise and segment objects, one can use high-capacity Convolutional Networks (CNN'S) to bottom-up region proposals, and (2) When labelling data is performance is considerably improved by sparse, supervised pretraining for an auxiliary task followed by domain-specific fine adjustment.

A model called an R-CNN, figure 2, or region-based convolutional network, is produced by merging CNNs and region proposals.

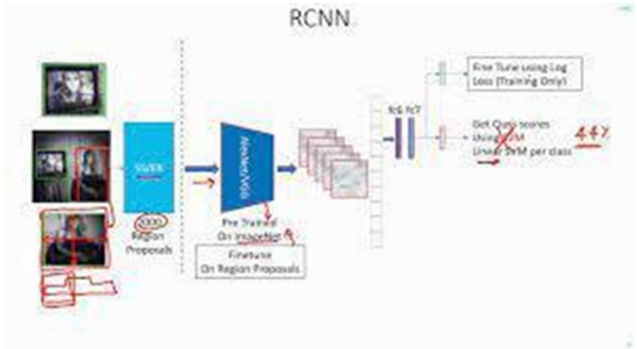

Figure 2. RCNN Architecture

To categorise area suggestions, we constructed the R-CNN architecture using AlexNet as the basic CNN. With the label categories of Bus, Cars, Pedestrians, and Traffic Signs, we offered 5969 labelled photos. The MATLAB R2017b Ground Truth Labeler tool was used to complete the labelling.
We discovered that processing each image took between 19 to 40 seconds. We also conducted a time study using five different RCNNs to see if the number of categories of items of interest could affect detection time. The results are shown in the table below.

TABLE I. TIME STUDY FOR RCNN'S

	1 Object of Interest	2 Objects of Interest	2 Different Objects of Interest	3 Objects of Interest
RCNN 1	15.586	20.25	20.19	22.03
RCNN 2	14.565	18.85	19.10	21.30
RCNN 3	17.432	19.59	18.90	23.35
RCNN 4	14.4031	19.85	20.25	22.15
RCNN 5	15.70	19.50	19.04	23.035

Although RCNN works quite well, it is rather slow to implement in real-time object detection. These are the causes, every area proposal for every image requires a CNN (AlexNet) forward pass, which amounts to about 2000 forward passes per image, The CNN model, which creates picture features, the classifier, which predicts the class, and the regression model, which tightens the bounding boxes, must all be trained individually. Because of this, training the pipeline is very challenging.

B. Fast R-CNN

In this framework, after entering an image, it is passed through a CNN (AlexNet) to create a visual feature map, from from which we obtain our region proposals. RoI pooling is done in the following step. The forward pass of a CNN image is shared by RoIPool among its subregions. Then, each region's attributes are combined (usually through max pooling).

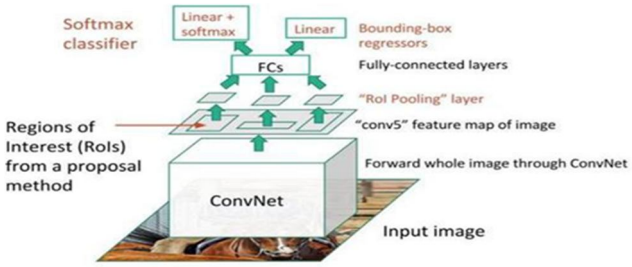

Figure 3. Fast RCNN Architecture

TABLE II. TIME STUDY FOR FAST RCNN

	1 Object of Interest	2 Objects of Interest	2 Different Objects of Interest	3 Objects of Interest
Fast RCNN 1	.85	2.01	2.84	3.32
Fast RCNN 2	1.565	2.90	1.98	2.96
Fast RCNN 3	1.93	3.55	2.89	2.30
Fast RCNN 4	1.61	2.94	3.85	3.15
Fast RCNN 5	1.87	1.97	2.79	3.38

According to the time study, Fast R-CNNs perform comparably better than R-CNNs in terms of speed. However, if we pay close attention and in accordance with, the computation of region proposals takes the majority of the

runtime of the Fast R-CNN. Faster R-CNN, a new method that was put up to minimize that, has made it possible to deploy area-based object detection techniques in real time as well.

C. Faster R-CNN

They fixed the region proposer, the last remaining snag in the Fast R-CNN process. As we saw, creating a large number of potential bounding boxes or regions of interest to test is the first stage in finding the locations of objects. Selective search, a somewhat slow technique that was discovered to be the process bottleneck of Fast R-CNN, was used to develop these ideas. These region ideas are produced more quickly by R-CNN using CNN features. Faster R-CNN creates the Neural Network Approach by adding a Convolutional Neural Network on atop of the CNN's characteristics. The CNN feature map is traversed by a moving window used by the Region Proposal Network, which outputs k probable anchor boxes and ratings for how excellent each one of those regions is anticipated to be at every frame

We intuitively understand that items in a picture should conform to standard aspect ratios and sizes. In doing so, we produce k such typical aspect ratios that we refer to as anchor boxes. We produce one bounding box and a score for each place in the image for each such anchor box. After RoI pooling, the image is passed via the Fully Connected convolution layer to match the CNN's inputs.

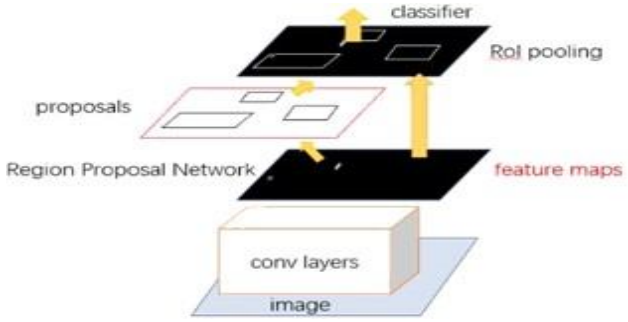

Figure 4. Faster RCNN Architecture

We executed the data analysis diversely than we performed in RCNN and Fast RCNN since we found that the number of elements of attention in a picture does not significantly affect efficiency. We came to the conclusion that the average detection time for an image is 0.2 seconds. The results of our comparison with Girshik et al. were similar. The following figure displays the relative times for detection for the region-based methods for object detection.

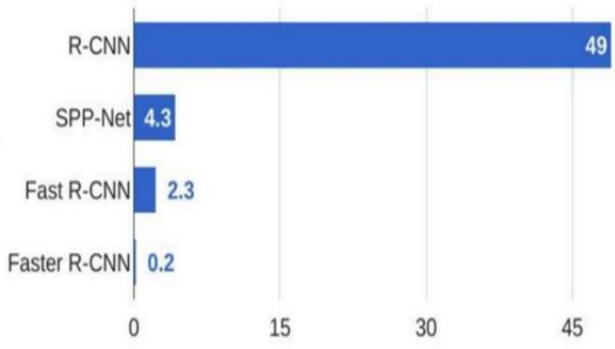

Figure 5. Comparison to other CNN

With the help of the time research that was conducted, we were able to arrive at the reasoning that the Faster R-CNN can be employed for context of physical recognition and localization. We also determined the Mean Average Precision (mAP) of our Faster R-CNN, and the result was around 70%, which is acceptable given the volume of training data offered.

We then used our trained Faster R-CNN to calculate the separation between entities, in this example automobiles, pedestrians, and traffic signs, from the sensor (camera) mounted on the vehicle. This illustrates how computer vision-based object identification paves the way for ADAS applications and is essential for the automation of moving vehicles.

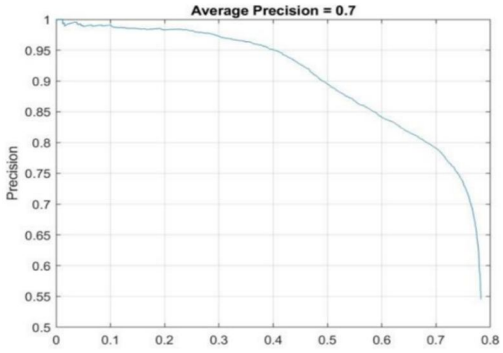

Figure 9. Mean Average Precision (mAP) of our Faster RCNN

III. IMPLEMENTATION OF FASTER RCNN FOR DEPTH ESTIMATION

A. Stereo Vision

Like human eyes, stereo vision has the ability to determine the distance between two images collected from different perspectives. Disparity is the appearance of the identical thing appearing to be in two different places when viewed from two different angles. The depth particulars is then derived using this discrepancy and the stereo camera specifications. Two cameras with parallel optical axes that are spaced apart by a certain amount. The baseline is the line that unites the centres of the camera lenses.

- Both cameras have a focal length of f.
- This system's origin O should be situated halfway inbetween the lens centers.
- The 3-D world coordinate system's x axis should parallel to the starting point.
- Consider a point (x ,y ,z) in 3-D world coordinates on an object.
- Assume that the point (x, y, z) has (x' , y') image coordinates in the left sector.
- Assume that the point (x, y, z) has right-plane image coordinates (x' , y').

In this study, feature set is inferred from a combination of image sequences using stereo vision. When two cameras are placed a certain distance apart, they each record a slightly distorted picture of the same scene in reality. If we can accurately correlate which characteristic qualities in the photo from the left lens correspond with which features in the photo again from right lens, and if we understand the stereo imaging topology and screen depth of field, we can reassemble the depth details.

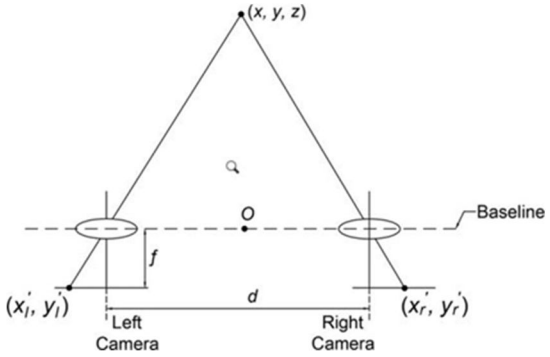

Figure 10. Stereo Vision

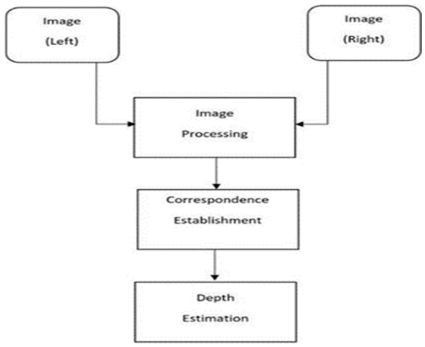

Figure 11. Depth Estimation Technique

The three basic steps in stereoscopic vision are typically pre-processing data to obtain matching features, reconstructing the disparity in between frames to use a suitable stereo technique, while using geometries to retrieve the stereo depth.

B. Disparity Mapping

A stereo camera, or two horizontally positioned cameras, makes up a stereo vision system (i.e., one on the left and the other on the right). These cameras simultaneously take two images, which are subsequently processed to recover the visual depth data. A sequence of stereo images can be utilized to determine depth data by first determining the difference in pixels in between format's position in one photo and its placement in the second

picture. The block identifying algorithm requires a local strategy that places restrictions on a small selection of pixels close to the desired pixel. By contrasting a small area close to a destination with congruent areas extrapolated from an adjacent image, the disparity at that place is estimated. Block matching looks for the best area in one image that matches a template in the other image. In our approach, we calculated the discrepancy using block matching.

C. Camera Calibaration

The lens and image sensor attributes of the camera to take a picture or a video are approximated via geometric camera calibration. These variables can be used to assess object size, adjust for lens distortions, or locate the camera in a scene. This makes it possible for us to use machine vision for 3D scene reconstruction, robot navigation, and object detection and quantification.

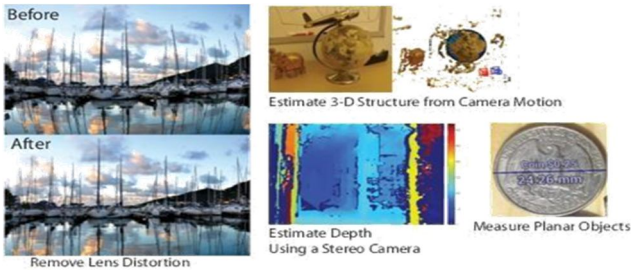

Figure 12. Camera Calibration to remove distortion

The intrinsic, extrinsic, and distortion factors make up the camera parameters. The extrinsic parameters signify a hard transition from the cartesian coordinates of the 3-D world to that of the 3-D camera. The intrinsic parameters show a geometric transition from the values of the 3-D camera to the dimensions of the 2-D image. Let [X, Y, Z] represent the 3D world coordinates, [Xc, Yc, Zc] represent the camera waypoints, and [x, y] indicate the 2D picture coordinates.

D. 3D Reconstruction and Depth Estimation

With just two images from the same scene, stereo vision can estimate the geometry of a three-dimensional scene. Rectification significantly reduces the complexity of the stereo correspondence issue, allowing for the simple production of large disparity maps, which serve as the foundation for dense 3D restoration.

Every element in the disparity mapping can indeed be re-projected to a 3D point in relation to the lens coordinate frame. Every physical location in a stereo camera system that captures two images produces a pair of 2D projections on the two images, referred to as m1 and m2, called m1 and m2. The intrinsic and extrinsic stereo system variables, m1 and m2, can be used to reconstruct the 3D placement of the point M assuming that we are familiar of them both.

Figure 13. Result of Depth Estimation

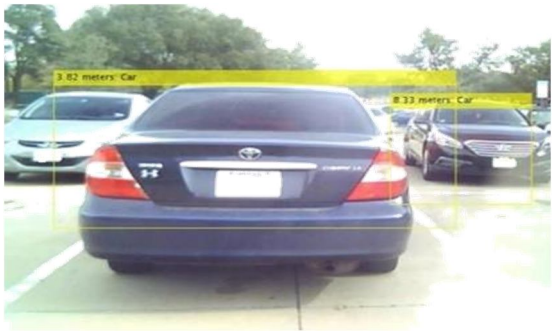

Figure 14. Result of Depth Estimation 2

IV. CONCLUSION

In this study, we introduced a faster region-based convolutional neural network architecture (Faster RCNN) that is effective in detecting objects that are present on the road in typical driving situations. As part of creating and evaluating this method, we discussed best practices for training, analyzed a number of network architecture options, and illustrated the value of tuning, proposal generation, and how the CNN model's depth affects

detection performance. We also came to the conclusion that the networks' mean average precision can be influenced by the quantity and quality of training data (mAP).

A. Future Scope

OEMs have made significant expenditures in the creation of ADAS systems utilising computer vision, which is speeding the pace of research in this area and the creation of infrastructure (Stereo Cameras) that is suited for ADAS applications. Let's sum up by saying that object recognition and detection using computer vision is among the most hotly contested topics in the present decade, and there is still a lot to be done. Fully self-driving cars will soon be a reality, and computer vision and deep learning will play a significant role in advancing that innovation.

REFERENCES

[1] W. Tan, N. Qin, L. Ma, Y. Li, J. Du, G. Cai, et al., "Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), pp. 202-203, Jun. 2020.

[2] M. Ibrahim, N. Akhtar, M. Wise and A. Mian, "Annotation Tool and Urban Dataset for 3D Point Cloud Semantic Segmentation," in IEEE Access, vol. 9, pp. 35984-35996, 2021, doi:.

[3] Ross Girshick, Jeff Donahue, Student Member, IEEE, Trevor Darrell, Member, IEEE, and Jitendra Malik, Fellow, IEEE. "Region-Based Convolutional Networks for Accurate Object Detection and segmentation" Copyright © 2016, IEEE.

[4] Rostam Affendi Hamzah and Haidi Ibrahim, "Literature Survey on Stereo Vision Disparity Map Algorithms," Journal of Sensors, vol. 2016, Article ID 8742920, 23 pages, 2016. doi:10.1155/2016/8742920

[5] Jeff Salton, "Volvo S60 details revealed ahead of Geneva premiere", Published on February 10th, 2010, (last accessed on 11/27/2017).

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, ''PointNet++: Deep hierarchical feature learning on point sets in a metric space,'' in Proc. Adv. Neural Inf. Process. Syst., 2017, pp.5099–5108.

[7] Chen, Xinlei, Ross B. Girshick, Kaiming He and Piotr Dollár. "TensorMask: A Foundation for Dense Object Segmentation." 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019): 2061-2069.

[8] X. Xiang, M. Zhang, G. Li, Y. He, and Z. Pan, "Real-time stereo matching based on fast belief propagation" , achine Vision and Applications, vol. 23, no. 6, pp. 1219–1227, 2012.

[9] L. Nalpantidis and A. Gasteratos, "Stereovision-based fuzzy obstacle avoidance method," International Journal of Humanoid Robotics, vol. 8, no. 1, pp. 169–183, 2011.

[10] Xu, Y., Arai, S., Tokuda, F., and Kosuge, K., "A convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color," IEEE Access 8, 70262–70269 (2020). https://doi.org/10.1109/Access.6287639 Google Scholar.

[11] Nguyen, A. and Le, B., "3d point cloud segmentation: A survey," in [RAM 2013], 225–230, IEEE, [Piscat-away, N.J.] (2013).

[12] Lyu, Y., Huang, X., and Zhang, Z., "Learning to segment 3d point clouds in 2d image space," 12255–12264 (2020).