Enhancing Autonomous Driving Safety through Bird's Eye View and Lane Changing Maneuvers.

Tejas Pankaj Kalsait

University at Buffalo
tejaspan@buffalo.edu

Athindra Bandi

University at Buffalo
athindra@buffalo.edu

Samarth Gangwal

University at Buffalo
samarthg@buffalo.edu

Lakshmi Ramaswamy Vishwanath

University at Buffalo
lramaswa@buffalo.edu

Abstract

In today's world of automation, safety is of utmost importance, especially with the rise of self-driving cars. These cars must drive responsibly, avoid obstacles and follow traffic rules to ensure the safety of passengers and pedestrians. To achieve this, we use multiple cameras installed in the car to monitor the surroundings. By incorporating a bird's eye view camera, we can combine the images from all the cameras to create a top-down view. The top-down view lets the car detect obstacles and navigate more safely and efficiently.

1 Introduction

Autonomous vehicles have been a futuristic dream for some time now, and while there have been advancements, they are still a ways away from being commonplace on the roads. Safety is a crucial aspect holding them back. Although it has greatly improved from rudimentary, simple sensors, continuous evaluation and improvement are necessary. Multiple cameras are mounted in the car to capture views from all directions. The input from these cameras is used to ensure that the car travels without any hindrance to the public in the vicinity. In this project, we have implemented a method that enhances the experience of autonomy. The images from the onboard cameras are retrieved to make a complete sense of the surroundings. We perform homography on the images to obtain a complete bird's eye view. Bird's eye view is the literal sight viewed by a bird above the scene. Once we obtain the bird's eye view, training the car to perform said actions becomes relatively more straightforward since we have to deal with just one image that incorporates the entire surrounding rather than multiple images. We can then use annotation methods to supervise its movements. In the end, after training, it will work as an independent model that will detect any obstacles and change lanes to avoid them.

2 Literature Survey

Generative Adversarial Frontal View to Bird View Synthesis[1]: In this work, they have used the BridgeGAN (generative model for bird view synthesis) as there is a large gap between the frontal and the bird view. Homography view is used to bridge the large gap, and the BridgeGAN learns the cross-view translation, i.e., frontal, bird, and homography view, to produce results.
The Right (Angled) Perspective[2]: Improving the Understanding of Road Scenes Using Boosted Inverse Perspective Mapping: In this work, Inverse Perspective Mapping is applied to remap the front-facing images of a vehicle into a 2D domain to provide a top-down view. The adversarial learning approach generates top-down view images that are sharper and have a more homogeneous illumination while all the dynamic objects are removed.
Learning to Map Vehicles into Bird's Eye View[3]: In this work, they have taken a huge synthetic dataset of 1M couples of frames (car dashboard and bird's eye view, respectively) To warp detections from the car dashboard view to bird's eye, a deep-network is trained.
Automatic dense visual semantic mapping from street-level imagery[4]: Semantic mapping or overhead mapping is performed of a region with some objects such as pavement, car, or road labeled. Each image is individually used to model the overhead view of a street. A geometrical function backs a region from the street view image into the overhead ground plane.

3 Bird's Eye View (BEV)

3.1 Methodology

CNN is the primary network used to achieve the bird's eye view image. Convolutional Neural Networks are one of the popular deep learning models used for image and video recognition tasks. It is designed to recognize designs and patterns and process them in a certain way. One of the challenges while dealing with BEV is occlusions. When obstacles like vehicles or any other object are on the road, it gets captured by the cameras. However, when the images are combined for the final BEV, the parts of the scene that was behind the obstacle and not visible to the camera are not available for the BEV. In such a situation, a semantic class is introduced and used on the ground truth images in the preprocessing stage. This inserts virtual rays from the camera to the object along the maximum visible range to show how much environment is captured. All the pixels within this range are processed to determine the level of occlusion. In the BEV image, the occluded part appears like a shadow along the range of the extended virtual rays.

In the paper written by Lennart Rehier [5], to obtain the proper BEV, we incorporate projective transformation between camera frames. These transformations are called homography. The homography matrix depends upon the intrinsic and extrinsic parameters of the cameras.
Consider $x_w \epsilon R^4$ and $x_i \epsilon R^3$ are homogeneous world and image coordinates, respectively. The relation between them is defined as

$$x_i = P x_w$$

The projection matrix encodes the camera's intrinsic parameters (e.g., focal length) in a matrix K and extrinsics (rotation R and translation t w.r.t. the world frame):

$$P = K[R|t]$$

Assuming there exists a transformation M R 4×3 from the road plane $x_r \epsilon R^3$ to the world frame, s.t.

$$x_w = M x_r$$

we obtain a transformation from image coordinates to the road plane:

$$x_r = (PM)^{-}1 x_i$$

This preprocessing part is applied to all images obtained from the camera. Only after this transformation are the images combined into a single image called the Homography image. If the pixels are overlapping after concatenation, they are chosen arbitrarily from either of the original images.

3.2 Model Architecture

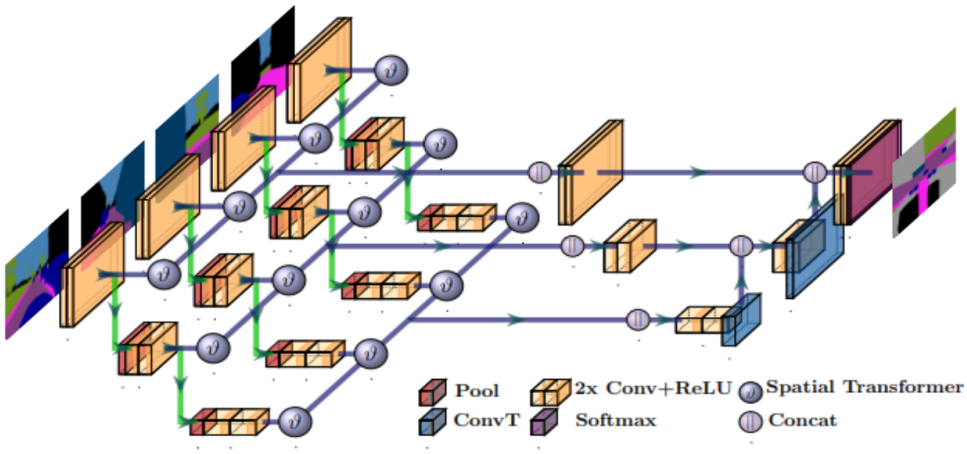


Figure 1: Neural network architecture used for getting the bird's eye view image.[5]

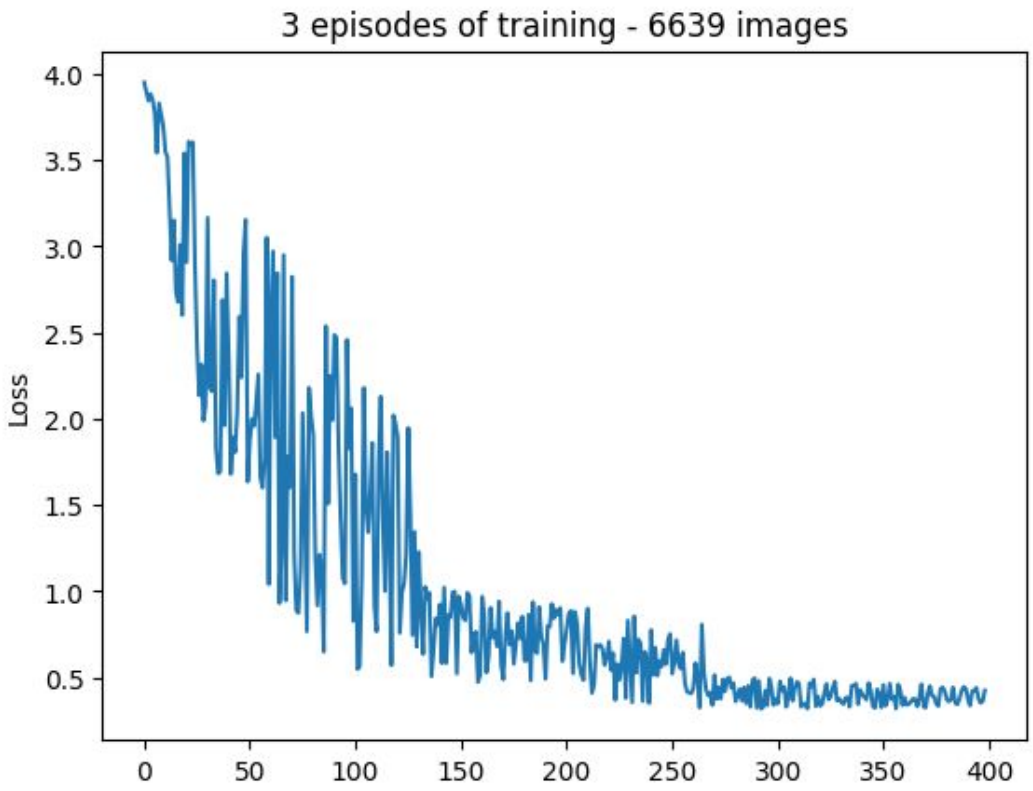
The model architecture used for the bird's eye view is inspired by the paper written by Lennart Reiher[5].
He has spoken about this model in his paper: "The uNetXST architecture has separate encoder paths for each input image (green paths). As part of the skip-connection on each scale level (violet paths), feature maps are projectively transformed (-block), concatenated with the other input streams (||-block), convoluted, and finally concatenated with the upsampled output of the decoder path. This illustration shows a network with only two pooling and two upsampling layers. The actual trained network contains four respectively.

3.3 Data Acquisition

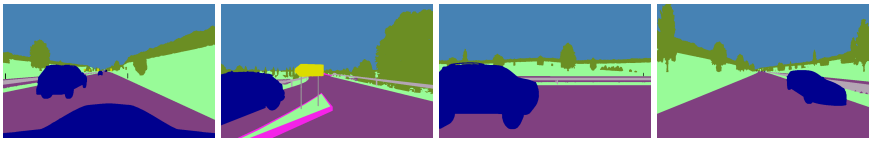
The data used to train and assess our proposed methodology is the same as the one used by Lennart Reiher[5], and is created in the simulation environment Virtual Test Drive (VTD). "The car has four identical virtual wide-angle cameras covering a full 360 degrees surround view. It has an approximate field of view of 70 m × 44 m. Both input and ground truth images are recorded at a resolution of 964 px × 604 px. The cameras are said to have produced both realistic and semantically segmented images. Nine different semantic classes are considered for the visible areas (road, sidewalk, person, car, truck, bus, bike, obstacle, vegetation)." As a whole, the dataset consists of around 33000 samples for training and 3700 for validation. Each sample is said to contain multiple input images and one ground truth label. We have assigned labels:

3.4 Training and Testing

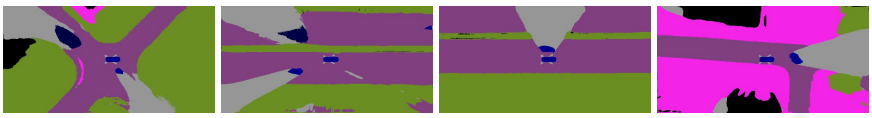
The model was trained for 3 epochs, and the results obtained are as follows:
The final training accuracy was observed to be 94.2%
and the final validation accuracy was 72%
The Loss graph for the model over 3 episodes of training is as follows:



Loss graph for used model



a.Front camera b.Right camera c.Left camera d.rear camera



The Bird's Eye View output by the model

3.5 Birds Eye View Summary

Implementing a bird's eye view system is a valuable tool for improving the safety and accuracy of self-driving cars in the absence of more expensive Lidar sensors. This system can aid with navigation, obstacle detection, and lane keeping by providing a full view of the vehicle's surroundings. The employment of high-resolution cameras, complex image processing algorithms, and 3D mapping technologies enables the system to generate a detailed and accurate representation of the surroundings in real-time. This can lessen the danger of accidents, improve the efficiency of the vehicle's operation, and improve the overall driving experience. As technology advances, it is projected that bird's eye vision systems will become more prevalent in autonomous vehicles, as the industry is moving away from the costlier Lidar sensors in favor of cheaper yet less accurate standalone cameras.

4 Lane Changing using BEV

4.1 Methodology

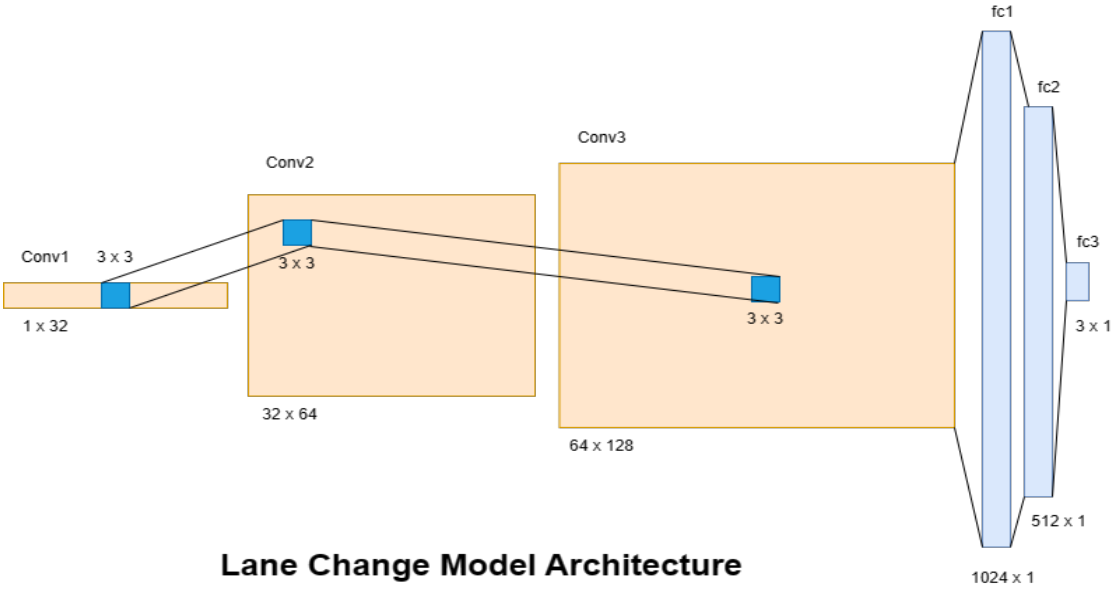
Our methodology is based on a Convolutional Neural Network (architecture given in 4.2).
From the outputs obtained in Bird's Eye View (BEV) in Part-3, we annotate the data and train it through our designed CNN model (single input) to predict whether the car must move left, straight, or right, to avoid any potential obstacles, We have annotated left as '0', straight as '1', and right as '2' denoting the presence of the closest obstacle.
So when there is an obstacle in front of the car, the network gives an output of left or right depending on the position on the road. If there is another coming from the left, the network suggests that the vehicle move right to avoid the oncoming vehicle.

4.2 Model Architecture

The model architecture implemented for "Lane changing using Bird's Eye View (BEV)" is as follows:

Model Architecture		
Layer	Layer Type	Layer Dimensions
Conv1	Conv2D	1 x 32
Conv2	Conv2D	32 x 64
Conv3	Conv2D	64 x 128
fc1	Linear	1024 x 1
fc2	Linear	512 x 1
fc3	Linear	3 x 1

The visual representation of the model we used for Lane Changing to avoid obstacles is:



Lane Change Model Architecture

4.3 Experimental Setup

4.3.1 Data Acquisition

Ground Truth Labels	
Direction	Label
Left	0
Straight	1
Right	2

Our data is taken from the output from the 1st part - Bird's Eye View. Followed by this, we have annotated our data from [6] and [7]. After the data annotation, it was divided into - training, testing, and validation datasets.

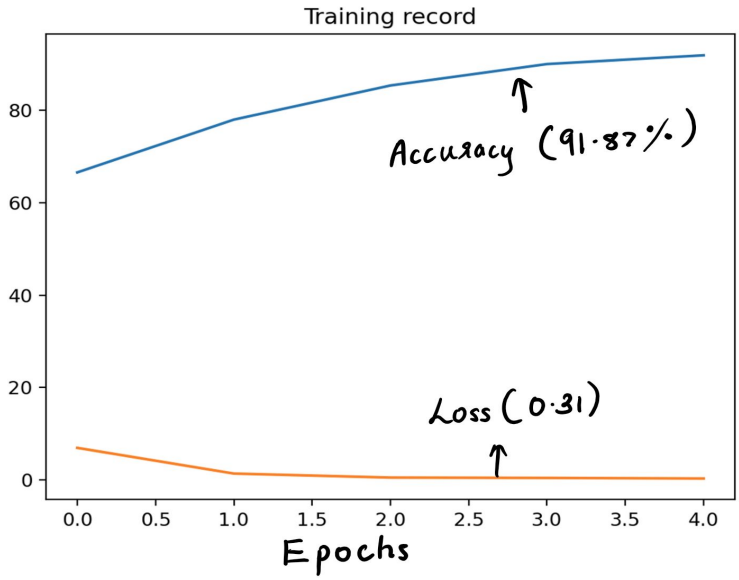
4.3.2 Training Setup

The total number of images used in the training was 2005 images, and for testing, and validation of our model we used 2005 images.
The hyper-parameters used are learning rate - 0.001.
The model was trained for 5 epochs, with Adam optimizer and a cross-entropy loss.

4.3.3 Evaluation Metrics

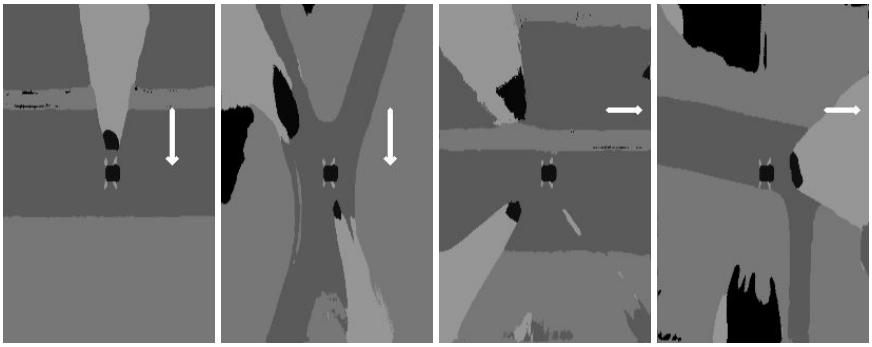
The testing accuracy obtained from our model was 91.87%.
The validation accuracy achieved was 88.43%.

4.3.4 Testing Results



4.4 Conclusion

We successfully trained our model to predict the lane change mechanism. The final precision accuracy that we got was 91.87%. As we can see in the images below, the ego vehicle is directed with a white arrow of the predicted direction to go. This can be expanded further by making accurate direction decisions for specific angles. The run time of forward propagation of both the models combines is less than 1 second on an 11th gen Intel CPU. Therefore, our entire project converts images from 4 cameras to a bird's eye view of the lane change safety mechanism, all in one go.



a.Right Prediction b.Right Prediction c.Straight Prediction d.Straight Prediction

The Bird's Eye View lane change output by our model

References

[1] Lennart Reiher and Bastian Lampe, Lutz Eckstein. A Sim2Real Deep Learning Approach for the Transformation of Images from Multiple Vehicle-Mounted Cameras to a Semantically Segmented Image in Bird's Eye View.

[2] Xinge Zhu, Zhichao Yin, Jianping Shi, Hongsheng Li Dahua Lin (2018). Generative Adversarial Frontal View to Bird View Synthesis. In Proceedings of the IEEE Conference on 3D Vision (3DV) (pp. 454-463)

[3] Liang, M., Yang, B., Chen, Y. (2019). The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2468-2476).

[4] Lee, D. H., Kim, J., Yoon, K. J. (2019). Learning to map vehicles into bird's eye view. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5252-5261).

[5] Cermelli, F., Caputo, B. (2019). Automatic dense visual semantic mapping from street-level imagery. In Proceedings of the IEEE International Conference on Computer Vision (pp. 5415-5424).

[6] https://github.com/heartexlabs/labelImg

[7] https://supervisely.com/