HOW MANY FEATURES IS AN IMAGE WORTH? MULTI-CHANNEL CNN FOR STEERING ANGLE PREDICTION IN AUTONOMOUS VEHICLES

Jason Munger and Carlos W. Morato

Department of Robotics Engineering, Worcester Polytechnic Institute,
Worcester, MA, USA

ABSTRACT

This project explores how raw image data obtained from AV cameras can provide a model with more spatial information than can be learned from simple RGB images alone. This paper leverages the advances of deep neural networks to demonstrate steering angle predictions of autonomous vehicles through an end-to-end multi-channel CNN model using only the image data provided from an onboard camera. Image data is processed through existing neural networks to provide pixel segmentation and depth estimates and input to a new neural network along with the raw input image to provide enhanced feature signals from the environment. Various input combinations of Multi-Channel CNNs are evaluated, and their effectiveness is compared to single CNN networks using the individual data inputs. The model with the most accurate steering predictions is identified and performance compared to previous neural networks.

KEYWORDS

Autonomous Vehicles, Convolutional Neural Network, Deep Learning, Perception, Self-Driving Cars.

1. INTRODUCTION

Though the general problem of autonomous steering is understood, more specific issues arise that prevent AI models from being deployed to a 4 or 5 level of autonomy. Steering angle predictions are directly related to external factors in the AV's surrounding environment: The path of the road (or lack thereof), surrounding vehicles, pedestrians, or objects in the immediate vicinity, etc. Even if an AV has self-steering capability, it requires stability and accuracy to drive in a wide variety of environments and react quickly to changes. While CNN's have allowed for advances in steering angle predictions by automatically learning features from RGB images, it is not enough to address the issues above. Humans steer cars using our eyes: We identify important features ahead, determine the location of the road, and discern relative distances to navigate traffic or in complex urban environments. Driving a car is not an innate skill that humans are born with, but rather a learned skill obtained through a multi-faceted "sensor suite" of our bodies and the experience of training in a variety of driving scenarios. Eventually, driving becomes habitual, almost instinctual, and isn't affected by never-before-seen environments. For vehicles to have the level of autonomy necessary to drive without human involvement, they will require a human level of situational awareness and driving skill. They must interact with the surrounding

David C. Wyld et al. (Eds): AIMLA, DBDM, CCNET - 2021
pp. 01-19, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111301

environment delineating between various scene objects in 3D space and deciphering which are important for steering and navigation decisions. There currently exists a debate between major automotive companies regarding the best method of sensor data AVs should rely on to achieve autonomy. On one side, companies like Tesla believe cameras should be the primary method an AV should sense its surroundings given the advances in camera technology and AI, particularly with image recognition [1]. Most other companies working on autonomous vehicles believe LIDAR is necessary to incorporate necessary depth features. However, there are significant downsides that include stability, cost, volume, and resources for visual recognition [2].

In parallel with the advances of steering angle prediction, deep learning has also made vast strides in other areas of computer vision including image segmentation and depth estimation. Image segmentation allows for the discrete detection and/or classification of objects within an image down to the level of individual pixels. This not only allows the model to detect the presence of a particular object, but also provides an accurate mapping of the location and boundaries of the object in an image. Depth estimations of the environment can be achieved through the use of stereo vision cameras or through LIDAR, however, deep learning has been able to achieve similar depth predictions using only 2D RGB images as demonstrated in [20] and [21].

With these considerations and advances in deep learning in mind, questions arose regarding the amount of useful information 2D mononuclear RBG images can provide for steering angle predictions:

1. Can image segmentation and depth estimates provide enhanced signalling features to a model to improve the accuracy and robustness of steering angle predictions?
2. Can the segmentation and depth estimates generated from RGB images *alone* provide sufficiently significant signalling power to an end-to-end steering angle prediction network?
3. Is it possible to extract and synthesize the outputs of independently developed pre-trained (off-the-shelf) models to use as inputs to another network?
4. What architecture provides the best performance with this extended dataset?
5. What impact do each of the additional features have on the overall steering prediction performance?

These questions are explored in this paper using a proof-of-concept neural network and evidence is provided demonstrating 2D RGB monocular camera images alone can provide sufficient signalling power to perceive the driving environment and provide accurate end-to-end steering angle predictions.

This paper is organized in the following manner: Section 2 provides a literature review of steering angle prediction methods, Section 3 provides a concise description of the steering angle problem this paper addresses; Section 4 details the proposed solution of using Multi-Channel CNNs to provide additional signals from RGB images for the prediction of steering angles; Section 5 presents and discusses the results of the proposed solution; and Section 6 draws conclusions of the work and provides potential future research areas that could expand and improve on this current work.

2. LITERATURE REVIEW

2.1. Computer Vision

Various methods of steering angle predictions of autonomous vehicles have been researched in recent years that include the use of computer vision and deep neural networks [23]. Though the end goal is the same, the approaches differ dramatically. Computer vision techniques have been applied to raw image data to manually extract relevant features from the frame, for example, road boundaries, and fitting curves or points that estimate the deviation of the vehicle orientation concerning the road as described in [3] or in [4]. While computationally light compared to deep learning techniques, these methods do not provide the robustness or accuracy necessary to provide steering commands to an autonomous vehicle in a multitude of environments and driving conditions.

2.2. Convolutional Neural Networks

Deep neural networks, particularly Convolutional Neural Networks, have provided breakthroughs in this area to automatically extract the features required from input images and map them to the steering angle. As early as 1989, researchers at CMU demonstrated the ability of their vehicle, ALVINN, to determine directions of travel using only a 3-layer neural network with artificially simulated road images [5]. Recent history has further demonstrated the power of CNNs as steering angle predictions have vastly improved such as in NVIDIA Corporation's creation of PilotNet. Here, researchers not only showed CNNs can learn pertinent road features from training data automatically, but they also demonstrated the use of images in an end-to-end system for AV steering [6]. This model has been used as the basis for other researchers looking to replicate or enhance the model such as in [7], who recreated the NVIDIA model architecture and trained it on augmented image data for use on a virtual driving simulator, and [8] who trained the same model himself using image data collected from a webcam taped to his car and reading CAN-BUS data into an Arduino microcontroller. Others, as in [9], utilize different architectures such as a 3D CNN with LSTM layers to include temporal data and use the concept of transfer learning to leverage high performing pre-trained models (i.e. ResNet50) for use in the new application of prediction steering angles with great success. Other variations of spatial and temporal type models have produced many of the state-of-the-art (SOTA) steering angle predictions. [27] Implements a combined CNN/LSTM/FC network with two sets of input images. One image sequence is provided by the Ego vehicle and another sequence is shared from a second vehicle ahead of it over a vehicle-to-vehicle (V2V) communication system. Both image sequences are used as inputs to the network to predict the steering angle of the Ego vehicle. [25] Uses spatio-temporal convolutions (ST-Conv) with ConvLSTMs to extract features at multiple levels of video sequences and inputs the information into an LSTM to predict steering angle, torque, and speed of the vehicle. [26] Implements Event Cameras to obtains asynchronous frames of pixels depicting changes in motion. Sequences of frames over a specified time interval are collected and used as inputs to a CNN for feature extraction and Fully Connected Network for steering angle predictions. [28] combines CNN and Conv-LSTMs to extract spatiotemporal features at varying levels and combines them with future steering angle information during training to predict the steering angle of the current time. [24] Utilizes Hierarchical Reinforcement Learning (HRL) through the use of a manager and worker network known as the Feudal Steering Network. This network uses a CNN+LSTM+FC architecture to obtain steering angle predictions from frames of a video sequence.

2.3. Multi-Modal Networks

While steering angle prediction neural networks produced unprecedented results, they still suffered from robustness and optimal accuracy issues due to varying driving conditions, illuminations, shadows, and road geometries that inhibit the ability of the neural network to extract necessary features. Many researchers have begun to utilize the concept of multi-modal end-to-end networks to leverage more information from the surrounding environment from cameras and onboard sensors to bring the AV closer to the context in which it is driving. [10] and [11] utilize auxiliary tasks or networks to include additional side models that perform tasks such as image segmentation and optical flow to be used to input features into a network similar to [9] with a 3D CNN + LSTM. [11] transfers low, middle, and high-level features of each auxiliary branch at the same level as the primary CNN model while [10] combines the optical flow and segmentation information as additional inputs with the original image. Both methods yielded better results than the baseline CNN architectures with raw images as the only inputs.

Alternative methods have been proposed, such as [12], where image data is fused with depth information from LIDAR sensors. [13] uses two separate CNN streams to extract spatial information from a processed image and temporal information from pre-calculated optical flow features and merges them before passing through an MLP regression network. [14] utilizes a multitask network using image inputs as well as speed sequences to predict the steering angle and speed of the AV accurately. Researchers have utilized multi-modal networks for other autonomous vehicle applications such as in [15] where LIDAR front view, bird view, and raw front camera image data are processed and fused for 3D object detection. [29] Uses a combination of ConvLSTMs to generate future frame predictions, thereby obtaining future steering angle estimates, and combines auxiliary data in the form of image segmentation to predict steering angles.

All of the techniques presented have shown that while CNNs have been a powerful tool in end-to-end steering angle predictions, including more contextual information of the AV's environment is crucial to improving the robustness and accuracy of these predictions.

3. PROBLEM DESCRIPTION

A vital problem for autonomous vehicles is the ability to accurately and reliably predict steering angles in any number of different driving situations. A complex task in and of itself, it must also accomplish this with affordable technology for fully autonomous cars to become mainstream. Current technology has enabled the areas of image feature extraction, object detection and recognition, and sensor fusion that can combine to create autonomous systems. Given the expensive nature of this technology in terms of computational resources and price, it is difficult to assemble into a single package ready for level 4 or 5 deployments. While steering angle predictions have become accurate over time, current models still suffer failures due to high vehicle speeds, sub-optimal obstacle avoidance, and the inability to use RGB input as the sole signalling feature as described in [16]. A solution is presented to the steering angle prediction problem to improve accuracy and robustness and enable a reduction in the amount of sensor hardware, data, and software computation necessary to incorporate into an autonomous vehicle. The vehicle will have more capability for fewer resources and cost.

4. SOLUTION

4.1. Strategy

To accurately and robustly predict steering angles, several deep learning models are utilized. By incorporating additional information with the raw RGB image, a better model can be trained by increasing the relevant features of the surrounding environment for the model to use. Humans can identify areas of the environment that constitute a driving surface and determine the distance between ourselves and other objects due to our stereo vision system. To create an artificial model with similar capabilities, several neural networks are stacked together, each performing the function it was trained for, and pass the processed information into a combined "super" network that learns features from the outputs of each individual network. In this case, we want to simulate the ability to identify the driving path and perceive the depth of objects using only the data from an onboard camera system. We obtain this supplemental data using semantic segmentation and depth maps. Neural networks have been developed for each task and leveraged in this model. Figure 1 shows the topology of the proposed full model.

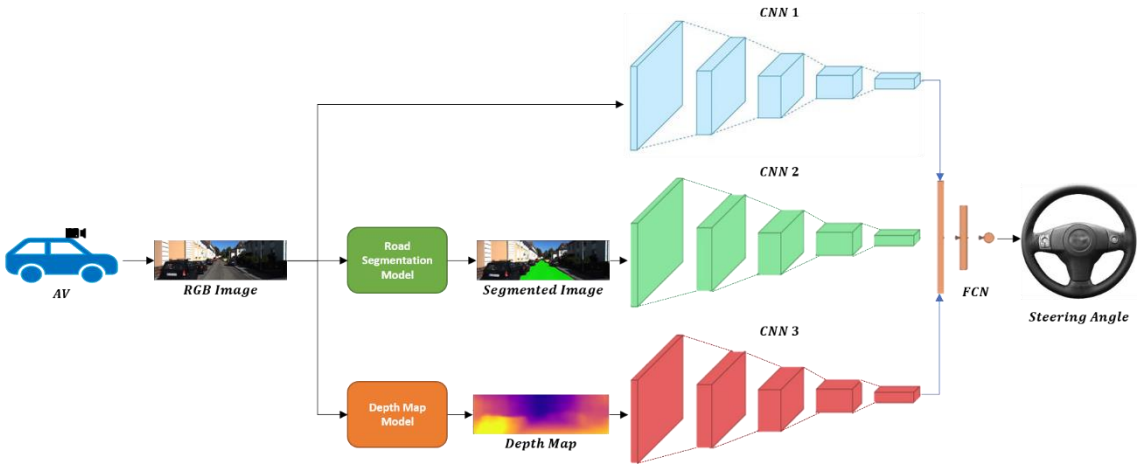

Figure 1. Topology of End-to-End Steering Angle Prediction Neural Network

4.2. Leveraged Pre-Trained Models

Input RGB images taken from a front-view camera are pre-processed through a segmentation model and depth model before passing into individual CNN towers of the primary network. Semantic segmentation allows for pixel-wise object detection and recognition of an image, enabling an autonomous system to gain more information about its environment and allows the system to find only objects for the specific context where the system is used. In the self-driving steering angle case, the AV needs to identify the road and how the driving path is changing. A pre-trained network called RoadSeg parsed through the entire RGB dataset producing another set that includes segmentation arrays in a probabilistic form. Each pixel is assigned a probability as a label of either "road" or "not road." The ability to delineate the road boundaries from the rest of the scene will provide additional signals to the model to learn how the road is changing at different steering angles. A pre-trained depth model trained from monocular camera video data described in [17] allowed for single image frames as input and resulted in an output dataset of depth predictions of objects in the image in the field of view of the camera. The output of this model is an array of values corresponding to the depth predicted for each pixel in the image. This data will require additional processing to scale the data for use as a depth map in the main networks as discussed in 4.4.3. For the multi-channel CNN model demonstration, these models

are assumed deployable and capable of generating data at a high enough speed to be utilized in real time and considered "black boxes" that perform a specific function whose details are not the subject of this project. Details of how they work can be reviewed in [18] and [17]. The outputs of the pre-trained models are shown in Figure 2. The top image shows the original input, the middle shows the road segmentation output, and bottom image shows the depth output. Note the image representations are not what the model actually "sees." The outputs of each have been processed to allow the human eye to understand them as an image. The output of the segmentation model consists of an array of probabilities each pixel belongs to the road class or not. The depth model outputs an array of values representing a distance map of the objects in the image. The values are scaled to a reference distance the model was trained with.

4.3. Full Network Architecture

4.3.1. RGB Image CNN

With the RGB images, semantic segmentation images, and depth map datasets created, a CNN model for each type of data is created and optimized to predict steering angles as accurately as possible. The initial inspiration for the models is the PilotNet architecture created by NVIDIA. For the RGB image CNN, a modified version of PilotNet was implemented. The architecture was largely similar, however, given the slight image size difference, a model with less parameters was constructed. The model consists of a 3 channel RGB input, a normalization layer (not part of the main architecture), 5 convolutional layers with decreasing kernel size (5x5 to 3x3) and stride (2x2 to 1x1) and increasing filters (24, 36, 48 and 64), a flattening layer, followed by a fully connected network with 80, 40, 10 and 1 neurons in each layer, respectively. Each layer used the ReLu activation function. The model can be seen in Figure 3 on the left and compared to the PilotNet model on the right.

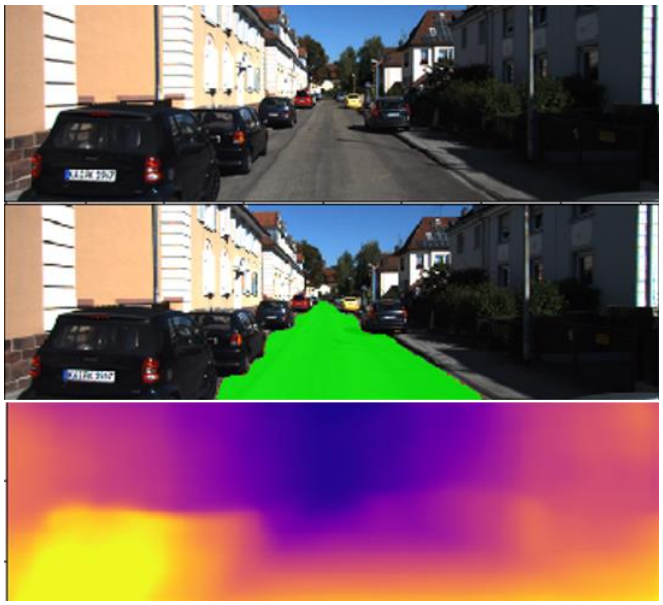

Figure 2. **Top:** Raw RGB Image, **Middle:** Road Segmentation Image, **Bottom:** Depth Map (image representation)

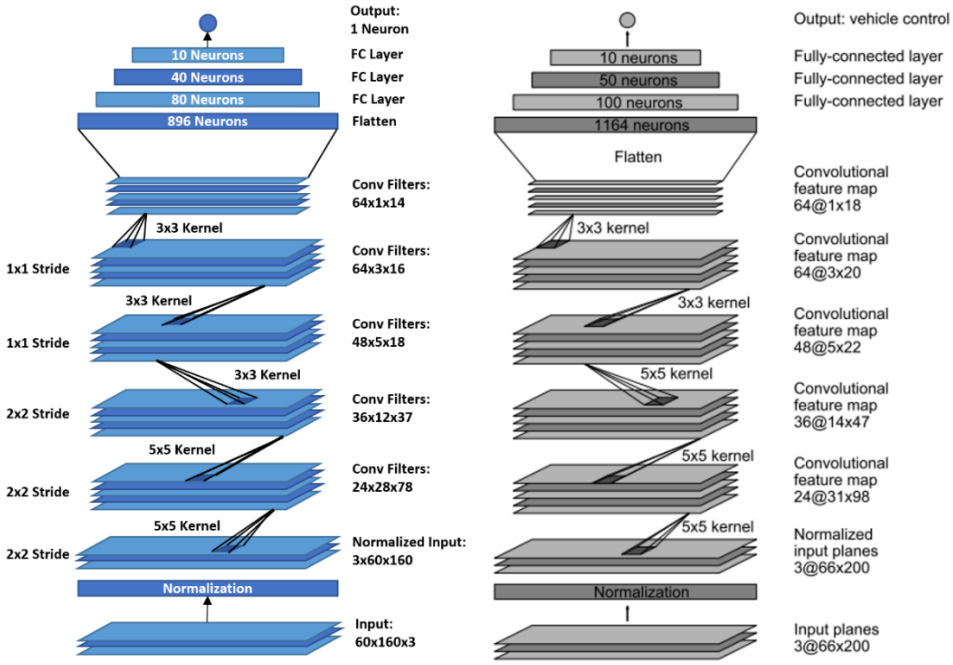

Figure 3. **Left:** Model Architecture for RGB input Multi-Channel CNN, **Right:** NVIDIA PilotNet Model Used as Inspiration and Comparison

4.3.2. Road Segmentation CNN

A CNN for the segmentation input was created following a similar layout as the RGB/PilotNet models though this time a VGG style layer block was followed. This model only required a single normalized channel input, followed by 2-layer blocks consisting of 2 Convolution layers with a kernel size of 3x3 and stride of 1x1 and a Max Pooling layer of size 2x2 and stride of 2x2. The number of filters increased in each subsequent layer from 24, 36, 48, and 64. Again the feature extractor outputs were flattened and input into a small 2 layer fully-connected network of 10 neurons and 1 output neuron. A small dropout of 0.1 was applied to the first FC layer and ReLu activation function was used for each layer. The segmentation CNN model can be seen on the Left of Figure 4.

4.3.3. Depth Map CNN

Finally, a CNN for the depth map input was created in the style of the previous models. Again, this model used a single normalized channel input into 2 VGG style blocks, flattened, and passed into a similar fully-connected network as the RGB model. The primary difference in this model is the number of filters, which range from 32 to 48, and the Dropout rates of 0.5, 0.4 and 0.4 applied to the first 3 fully-connected layers, respectively. ReLu was implemented as the activation function. The depth map CNN model can be seen on the Right of Figure 4.

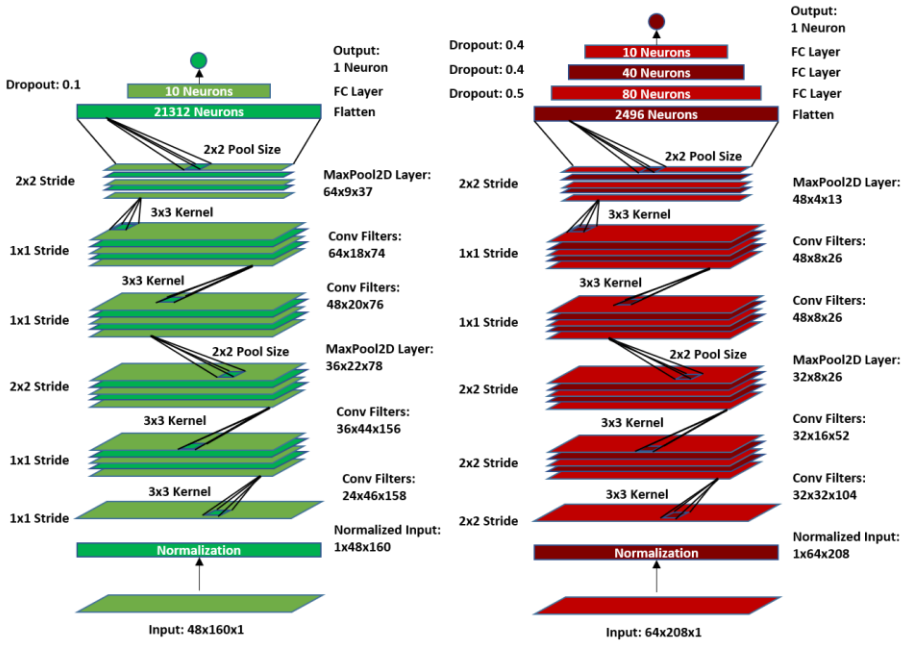

Figure 4. **Left:** Model Architecture for Seginput Multi-Channel CNN, **Right:** Model Architecture for Depth Input of Multi-Channel CNN

4.3.4. Full RGB + Segmentation + Depth Multi-Channel CNN

With the individual CNNs created and trained, a full Multi-Channel CNN can be built. To accomplish this, the fully-connected layers of each CNN was removed leaving only the flattened layers as the outputs. These outputs were merged into a single layer through concatenation. A new fully-connected network was installed consisting of 3 dense layers of 600, 300, and 60 neurons and a single neuron for the steering angle output. A dropout rate of 0.3 was added to each dense layer prior to the output. ReLu activation function was implemented. The weights of the individual networks were left unfrozen to allow for new weights to be computed. The model architecture for the full RGB + Segmentation + Depth CNN network can be seen in Figure 5.

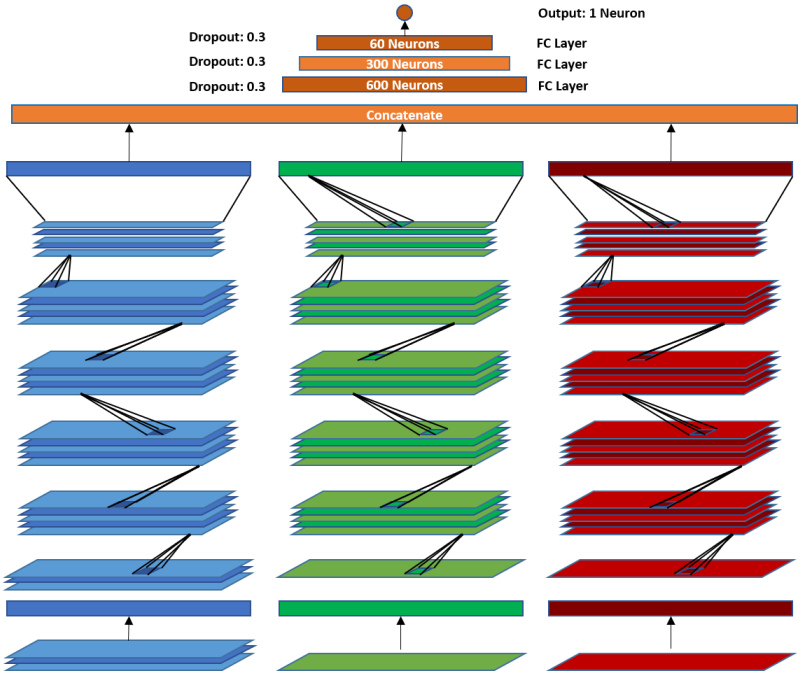

Figure 5. Model Architecture for Full RGB + Seg + Depth Multi-Channel CNN

4.3.5. RGB + Segmentation Multi-Channel CNN

After the full Multi-Channel CNN is built, it is simple to create different combinations of the model to explore how different image representations pair with another. An RGB Image and Segmentation Image Multi-Channel CNN was built nearly identical to the full network only with the depth channel removed. The dropout rate was reduced to 0.2 for this implementation. The model can be seen in Figure 6.

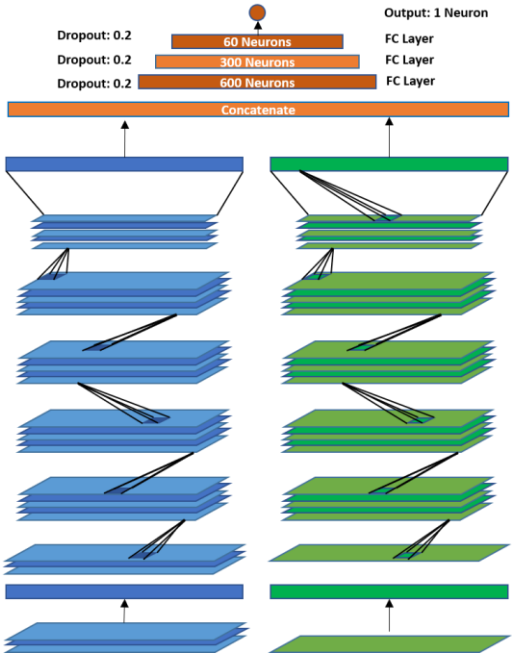

Figure 6. Model Architecture for RGB + Seg Multi-Channel CNN

4.3.6. RGB + Depth Multi-Channel CNN

An RGB + Depth Multi-Channel CNN was built next by removed the Segmentation channel from the full network. All else is identical to the RGB + Segmentation model. The architecture can be seen in Figure 7.

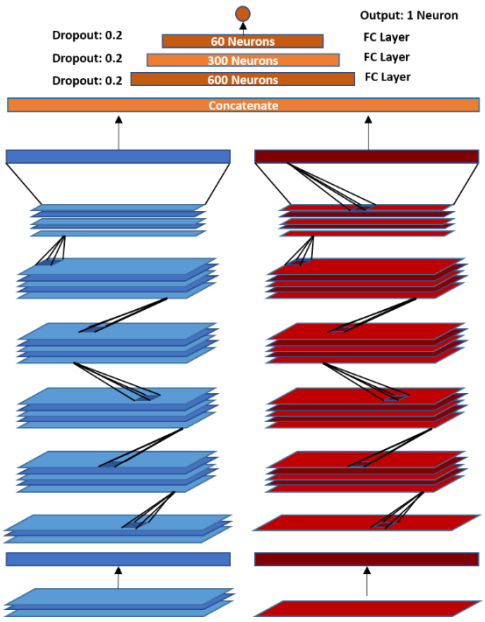

Figure 7. Model Architecture for RGB + Depth Multi-Channel CNN

4.3.7. Segmentation + Depth Multi-Channel CNN

Finally, a Segmentation + Depth Multi-Channel CNN was created by removing the RGB channel from the full model. This time the fully-connected network had less neurons: 80, 40, 10, and 1, respectively. The dropout rate was increased to 0.3 and all else was identical to the full Multi-Channel Model. The architecture can be seen in Figure 8.

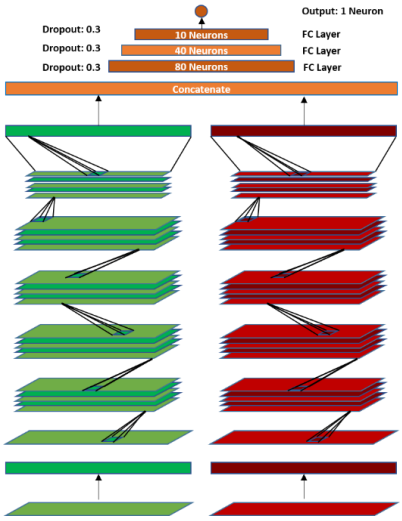

Figure 8. Model Architecture for Seg + Depth Multi-Channel CNN

4.4. Dataset & Data Pre-Processing

All networks trained using the Udacity self-driving dataset used in their self-driving steering angle prediction competitions consisting of 33,808 RGB images of size 640x480. Using this dataset provides a basis for comparison when evaluating model performance against available benchmark performances. The model used an 80/20 training/test dataset split. Each dataset for the individual models required various pre-processing steps given each model was trained on different size images and required their own normalization procedure.

4.4.1. RGB Images

The raw RGB images of the Udacity dataset were cropped the remove the top half to focus only on the road and remove background features that were not pertinent to all images. These images were then down-sampled by a factor of 16 to a final size of 160x60. The pixel values of the images were normalized to value between 0 and 1 by dividing each pixel by 255.

4.4.2. Segmentation Images

For the RoadSeg model described in detail in [18], the required image input size was 640x192 while the depth map model required an image size of 416x128. On top of this, the images needed to be prepared in a manner that resembled the dataset used in each model. Both the segmentation and depth map models were trained using the KITTI dataset [22]. Given this, the Udacity images were cropped to have a similar FOV and aspect ratio of the KITTI images to ensure the inputs were as close the original as possible. Once the cropping operation was complete on the RGB set, a new set was created for the segmentation and depth maps by resizing to the required input sizes. The output of the segmentation model was down-sampled to a final size of 160x48. The segmentation images were normalized such that the pixel values ranged between 0 and 1 representing the probability a pixel belonged to the class of “not road” or “road”, respectively.

4.4.3. Depth Maps

The output depth maps of the depth model were normalized in a similar manner used in by [17] to bound each value between 0 and 1 with values closer to 1 being nearer to the camera and values closer to 0 being farther away. This provides a uniform measurement of relative distances across all input images and provides the model with a synthesized depth signal that can be used to conjunction with the segmentations and RBG images to provide a full spatial representation of the driving scene.

4.5. Training Process

4.5.1. Framework

The Keras API with Tensorflow GPU-supported Backend was used to build, train, and evaluate all models in pursuit of this task.

4.5.2. Loss Function

The loss function chosen for optimization is the Mean Squared Error (MSE) in Equation 1. The MSE function is a measure of the average errors between the ground truth and model predictions. By minimizing this function, model performance is improved. This function also penalizes higher errors more severely due to the squaring term. This is advantageous when low errors are

important and high errors are undesirable, exactly as required in the steering angle prediction task.

$$MSE = \frac{1}{N}\sum(y_i - \hat{y})^2 \quad (1)$$

4.5.3. Evaluation Metrics

While the model is optimized on the MSE function, other metrics are used to evaluate performance. Root Mean Squared Error (RMSE) as shown in Equation 2 is almost identical to MSE with the only difference being the square root being applied. RMSE provides the same advantages as MSE, however, it also provides a metric in the units we care about (steering angle) and serves as the standard deviation of the data. The spread of predictions indicates how closely the data surrounds the regression line. A lower value indicates a more accurate model.

$$RMSE = \sqrt{\frac{1}{N}\sum(y_i - \hat{y})^2} \quad (2)$$

Mean Absolute Error (MAE), shown in Equation 3, is also used as a performance metric. MAE provides the average error the model is predicting giving each prediction equal weight. Using the MAE together with RMSE, the model performance can be better understood. While MAE will always be smaller than RMSE, the magnitude of the difference informs how high or low the variance is in a set of predictions. Higher magnitudes indicate more variance of the errors, while lower magnitudes (MAE and RMSE are closer to each other) indicate less variance of the errors.

$$MAE = \frac{1}{N}\sum|y_i - \hat{y}| \quad (3)$$

4.5.4. Optimizer

For training, the Adam optimizer was used. Adam is commonly used for training neural networks due to its adaptive learning ability and computational efficiency. It was found that the default parameters of the Adam optimizer were sufficient for training with the exception of the learning rate which was set to 0.0001 for all models.

4.5.5. Epochs & Batch Size

The models were trained for a range of 20 to 50 epochs depending on the performance. Some models converged earlier while others needed more time and exposure to the data. Each model was trained using a batch size of 50 samples which worked well regardless of the input data or model architecture.

4.5.6. Callbacks

A checkpoint callback was implemented to save best model and weights as they were achieved. This was done by continuously looking at the validation loss after each epoch to determine if model performance improved. This allowed the best model and weights during a training run to always be obtained regardless if the training process began to plateau or degrade by the end. These saved models also served as the individual channel CNN models in the main network. It

should be noted that while the best model and weights were always saved, the models were fine-tuned to ensure no underfitting or overfitting conditions occurred.

5. RESULTS

All seven CNN models were trained and evaluated with their performance documented in Table 1. For the individual models, the RGB Network performed the best while the RGB+Depth Network performed the best of the multi-channel networks and had the best performance overall. The worse performing individual network was the Segmentation network which also was involved in the worst performing multi-channel network in the Seg + Depth model. In order to understand the model results more, a plot of the real vs predicted steering angles as well as their errors was created for each network and can be seen in Figure 9.

Table 1. Steering Angle Prediction Results

Network Architecture	MAE (°)	RMSE (°)
RGB	2.15	3.30
Seg	5.97	12.20
Depth	4.01	7.88
RGB + Seg_+ Depth	1.25	2.56
RGB + Seg	1.19	2.43
RGB + Depth	**1.02**	**2.32**
Seg_+ Depth	3.38	7.07

The bias and variance of the predictions can be seen in each model's plot. A perfect correlation plot would show a trendline with a slope of 1 and y-intercept of 0, and a perfect error plot would show a trendline with 0 slope and 0 y-intercept. The "good" performing models can be seen in Figure 9 a, d, e, and f and "bad" performing models can be seen in b, c, and g. These plots illustrate the relationship of the tabular performance values with how the data is distributed around the trendlines. For the "good" models, the predicted values are clustered tighter around the trendline and the dispersion of error magnitudes are smaller indicating lower variance. The RGB + Depth Model plots (f) demonstrate how the MAE and RMSE is the lowest of all with the predictions being the most tightly situated around the trendline as well as having the tightest error plot. The Seg Model on the other hand, shows why it performed the worst. Not only are the predictions severely off and widely dispersed, the trendline has less than half the slope of a well performing model indicating a problem with the data inputs. It was found the pre-trained RoadSeg Model was not performing well on the Udacity dataset even though the images were prepared as close to the KITTI dataset as possible. Upon further inspection of random samples, the segmentation of the road was spotty at best with some images having good results with the road fully segmented, while others had mere blotches of road, and many had no segmentation at all. This would explain the performance of this model as passing an array of zeros would not provide any useful information for steering prediction. In fact, it can be seen a steering angle of zero (or near zero) was predicted often over the entire range of possible steering angles. This indicates more work needs to be done to truly test the impact of the segmentation in multi-channel CNNs and should not be discounted as a means of improving steering performance.

One of the most interesting aspects of applying the multi-channel CNN models to the inputs, is the ability for the models to learn how to make use of this "bad" data and calculate a set of weights that provide better steering angle prediction performance than it could in the individual models. As long as the RGB image inputs were present in a multi-channel CNN, the model performed better as whole than the individual channels alone. This can be seen when the

segmentation and depth models were paired together. The two worst performing individual models created the worst performing multi-channel model, which was expected. However, it was expected the full RGB + Seg + Depth model would provide the best overall performance, yet it was the RGB + Depth model that came out on top. This demonstrates that one, multi-channel CNNs can extract useful features from different types of input data, and two, the introduction of depth information provides another spatial component that assists in predicting steering angles.

Another model performance is evaluated from a different perspective by plotting the predicted steering angles over a trajectory. Predictions from each model were calculated and plotted over a subset of the Udacity test sample data and can be seen in Figure 10. In Figure 10a, a plot of all individual CNNs are overlaid on the ground truth data, b overlays all multi-channel CNN model predictions over ground truth, and c and d overlay the best (RGB+Depth) and worst (Seg) predictions over the ground truth, respectively. Significant performance increases can be seen between the single and multi-channel models again reinforcing the idea different image inputs are useful for the steering prediction task. The contrast between the best and worst model is stark. The best model is able to tightly follow the ground truth data over a wide variety of angles (-50 to +70 degrees) while the worst is unable to reliably follow the trajectory. It should be noted that all models struggled with large steering angles though obviously some are better than others. This may be attributable to a lack of data for larger angles. The dataset is oversampled with smaller angles as that is what most driving conditions require. More data containing larger steering angles will help this problem through upsampling of less frequent angles by adding flipped images of existing data or downsampling more frequent smaller angles to remove their bias. Downsampling may come at the expense of less accurate overall predictions which only emphasizes the need for more training data.

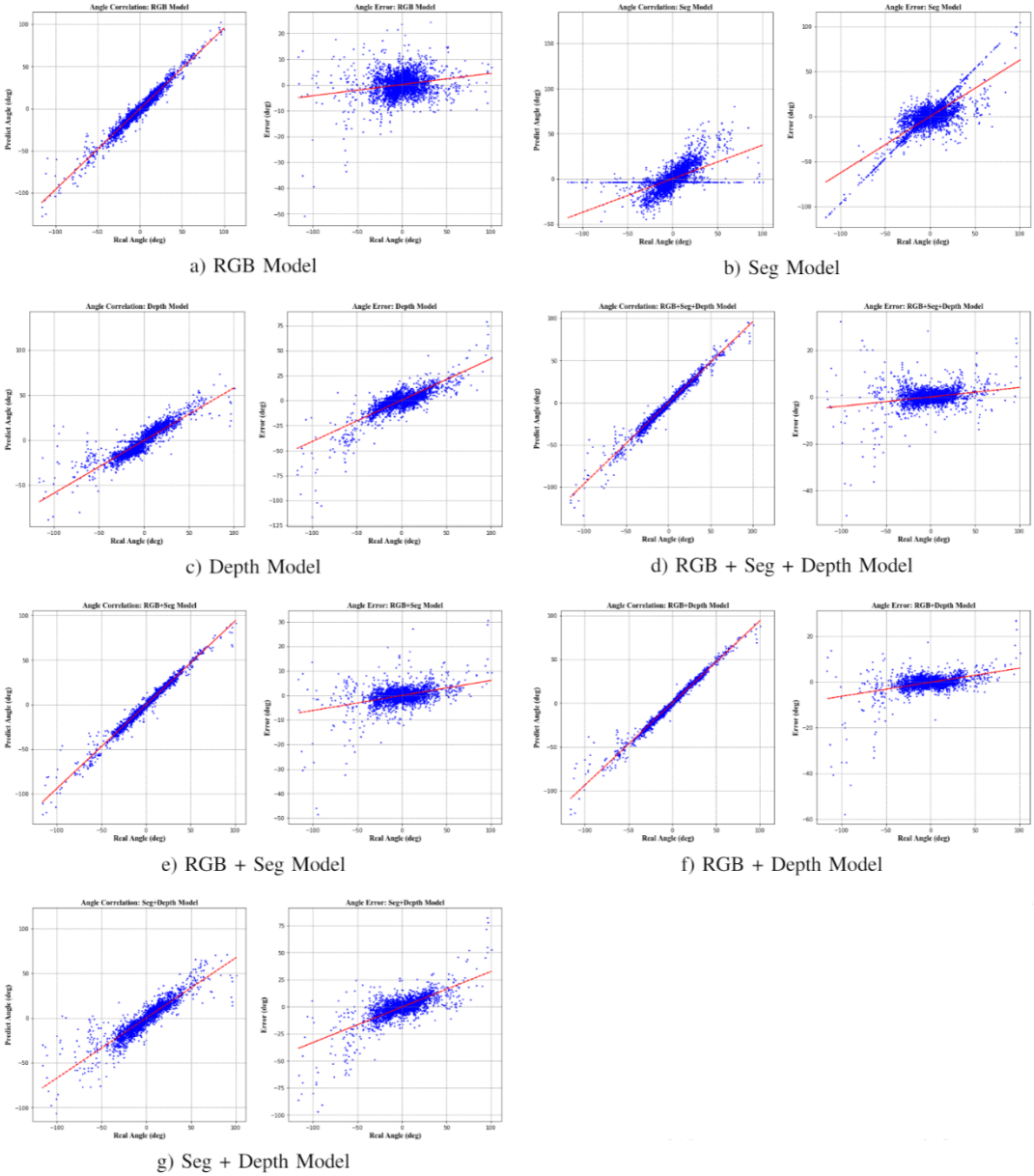

Figure 9. Steering Angle Correlation and Error Between Ground Truth and Predictions

To evaluate the efficacy of multi-channel CNNs for steering angle predictions, the performance of the best model (RGB + Depth) was compared to previous steering angle prediction models trained on the Udacity dataset. Table 2 shows the comparison. Previous models utilized CNN models though some implemented variants. These variants include standard structures as in the NVIDIA PilotNet architecture, transfer learning of pre-trained CNN feature extractors, a temporal aspect using 3D CNN or LSTM models, or a network utilizing auxiliary tasks which is most similar to this current work. The addition of a temporal component increased performance however, the FM-Net combined auxiliary networks to provide feature inputs of segmentation and optical flow using a 3D ResNet and LSTM architecture performed best among them [11]. It was shown the RGB + Depth model outperformed all of these models, many considered SOTA, relying only on single image inputs to a simpler CNN architecture. It is suspected that had the

Segmentation inputs were of higher quality, the full network (RGB + Seg + Depth) likely would have provided even better performance though this needs verification through training.

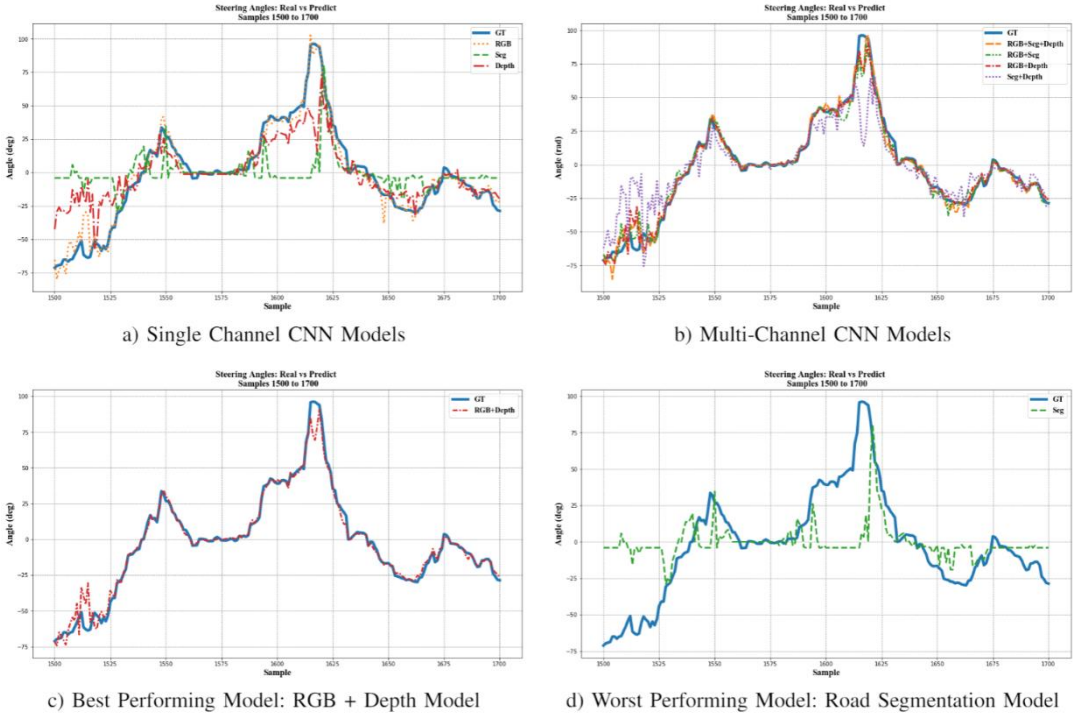

Figure 10. Steering Angle Prediction Results on Udacity Dataset (Sampled from Test Set)

Table 2. Comparison of Methods on Udacity Dataset

Network Architecture	**MAE (°)**	**RMSE (°)**
CNN + FCN (NVIDIA) [19]	4.12	4.83
CNN + LSTM [19]	4.15	4.93
3D LSTM [9]	-	6.44
ResNet50 Transfer [9]	-	4.06
3D CNN [11]	2.56	3.66
MSINet [28]	-	2.81
3D CNN + LSTM [11]	1.86	2.72
Feudal Steering [24]	1.09	2.67
FM-Net [11]	1.62	2.35
RGB + Depth	**1.02**	**2.32**

The results of the models clearly indicate that not only can additional features besides the RGB mononuclear images enhance the signals to the model in the development of a robust and accurate end-to-end steering angle prediction model, but the additional features to enhance the spatial awareness of an autonomous vehicle can be synthesized from the RGB images of an onboard camera alone. This implies the potential for AVs to perceive, learn, and navigate the driving environment from minimal data input. It also demonstrates how individual deep learning models can be trained for separate specific tasks and combined for use in new applications and potentially increasingly complex tasks.

6. CONCLUSIONS

This paper demonstrated the effectiveness of Multi-Channel CNNs using different camera image representations to accurately and reliably make steering angle predictions. Compared to individual CNN models trained on separate inputs, multi-channel CNNs allow for improved performance without the introduction of feature signals other than the images provided from the onboard camera of an autonomous vehicle. It was shown that depth data computed from a pre-trained model in combination with an RGB image provide the best overall steering angle predictions. Networks involving road segmentation provided the worst performance, however, this is most likely due to the pre-trained model's inability to make predictions on different data than the one it was trained on, or the data required an alternate pre-processing procedure to predict effectively. More work is required to investigate segmentation as additional signals and should not be completely dismissed from the results of this paper. The best multi-channel CNN exceeded performance of previous models using various network architectures that included spatial and temporal elements, leveraged transfer learning techniques, and implemented parallel auxiliary networks to feed various levels of features to layers of a single CNN network.

Future work may include refining the multi-channel network to train on other datasets, increasing samples of larger steering angles, or implementing better performing pre-trained models to obtain accurate data inputs. A temporal aspect was not considered for this architecture; however, it is possible to implement data from a series of video frames rather than individual images and still provide a valid model that supports the goal of using camera image data alone to predict steering angles.

ACKNOWLEDGEMENTS

Thank you to WPI for providing a curriculum that enables the study of artificial intelligence and applications to real-world problems.

REFERENCES

[1] "'anyone relying on lidar is doomed,' elon musk says — TechCrunch,"https://techcrunch.com/2019/04/22/anyone-relying-on-lidar-is-doomed-elon-musk-says/, (Accessed on 04/25/2021).

[2] "Lidar vs. camera — which is the best for self-driving cars? —by Vincent Tabora— 0xmachina—medium,"https://medium.com/0xmachina/lidar-vs-camera-which-is-the-best-for-self-driving-cars-9335b684f8d, (Accessed on 04/25/2021).

[3] R. Meganathan, A. A. Kasi, and S. Jagannath, "Computer vision based novel steering angle calculation for autonomous vehicles,"*2018 Second IEEE International Conference on Robotic Computing (IRC)*, pp. 143–146, 2018.

[4] U. Venkatasubramanian, S. Amarjyoti, T. Bakshi, and A. Singh, "Steering angle estimation for autonomous vehicle navigation usinghough and euclidean transform,"*2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, SPICES 2015*, 04 2015.

[5] D. Pomerleau, "Alvinn: An autonomous land vehicle in a neuralnetwork," in*NIPS*, 1988.

[6] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang,J. Zhao, and K. Zieba, "End to end learning for self-driving cars,"2016.

[7] V. Singhal, S. Gugale, R. Agarwal, P. Dhake, and U. Kalshetti, "Steering angle prediction in autonomous vehicles using deep learning," in*2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, 2019, pp. 1–6.

[8] "How a high school junior made a self-driving car — by sullychen — towards data science," https://towardsdatascience.com/how-a-high-school-junior-made-a-self-driving-car-705fa9b6e860, (Accessed on 03/14/2021).

[9] S. Du, H. Guo, and A. Simpson, “Self-driving car steering angle prediction based on image recognition,” 2019.
[10] Y. Chen, P. Palanisamy, P. Mudalige, K. Muelling, and J. M. Dolan, “Learning on-road visual control for self-driving vehicles with auxiliary tasks,” 2018.
[11] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning to steer by mimicking features from heterogeneous auxiliary networks,” 2018.
[12] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. López, “Multimodal end-to-end autonomous driving,”ArXiv, vol. abs/1906.03199, 2019.
[13] N. Fernandez, “Two-stream convolutional networks for end-to-end learning of self-driving cars,” 2018.
[14] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, “End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions,”*2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2289–2294, 2018.
[15] X.Chen, H.Ma, J.Wan, B.Li, and T.Xia, “Multi-view 3d object detection network for autonomous driving,”*CoRR*, vol. abs/1611.07759, 2016. [Online]. Available:http://arxiv.org/abs/1611.07759
[16] U. M. Gidado, H. Chiroma, N. Aljojo, S. Abubakar, S. I. Popoola, and M. A. Al-Garadi, “A survey on deep learning for steering angle prediction in autonomous vehicles,”*IEEE Access*, vol. 8, pp. 163 797–163 817, 2020.
[17] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in *CVPR*, 2017.
[18] “Github - robertklee/kitti-roadseg: A course project for road segmentation using a u-net convolutional neural network on the kitti road2013 dataset,” https://github.com/robertklee/KITTI-RoadSeg, (Accessed on 04/25/2021).
[19] J. Kim and J. Canny, “Interpretable learning for self-driving cars by visualizing causal attention,” 2017.
[20] K. Cantrell., C. Miller., and C. Morato., “Practical depth estimation with image segmentation and serial u-nets,” in Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS, INSTICC. SciTePress, 2020, pp. 406–414.
[21] V. John, “Vision-based steering angle prediction by the fusion of depth and intensity deep features,” in *CVPR*, 2018.
[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” International Journal of Robotics Research (IJRR), 2013.
[23] H. Saleem, F. Riaz, L. Mostarda, M. A. Niazi, A. Rafiq and S. Saeed, "Steering Angle Prediction Techniques for Autonomous Ground Vehicles: A Review," in IEEE Access, vol. 9, pp. 78567-78585, 2021, doi: 10.1109/ACCESS.2021.3083890.
[24] F. Johnson and K. Dana, "Feudal Steering: Hierarchical Learning for Steering Angle Prediction," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 4316-4325, doi: 10.1109/CVPRW50498.2020.00509.
[25] Chi, L., & Mu, Y. (2017). Deep Steering: Learning End-to-End Driving Model from Spatial and Temporal Visual Cues. ArXiv, abs/1708.03798.
[26] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, “Event-based vision meets deep learning on steering prediction for self-driving cars,”2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5419–5427, 2018
[27] R. Valiente, M. Zaman, S. Ozer and Y. P. Fallah, "Controlling Steering Angle for Cooperative Self-driving Vehicles utilizing CNN and LSTM-based Deep Networks," 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 2423-2428, doi: 10.1109/IVS.2019.8814260.
[28] T. Wu, A. Luo, R. Huang, H. Cheng and Y. Zhao, "End-to-End Driving Model for Steering Control of Autonomous Vehicles with Future Spatiotemporal Features," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 950-955, doi: 10.1109/IROS40897.2019.8968453.
[29] F. Munir, S. Azam and M. Jeon, "Visuomotor Steering angle Prediction in Dynamic Perception Environment for Autonomous Vehicle," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1-6, doi: 10.1109/ICCE-Asia49877.2020.9276907.

AUTHORS

Jason Munger is a graduate student in the Department of Robotics Engineering at Worcester Polytechnic Institute. He has a B.Sc. in Mechanical Engineering from the Georgia Institute of Technology. He works full time at NASA's Jet Propulsion Laboratory in Pasadena, California as a Mechanical Engineer on a variety of space-bound instruments and mechanisms.

Carlos W. Morato is Professor in the Department of Robotics engineering at Worcester Polytechnic Institute. He is leading researcher in the fields of human robot interaction, collaborative robots self-driving vehicles, and meta-intelligence for augmented intelligent systems. He has a PhD from the Department of Mechanical Engineering, University of Maryland at College Park, USA, a M.Sc degree in Aerospace Engineering and a M.Sc degree in Computer Science both focused in intelligent robots.