A Study on Neural Networks From Pixels to Actions: Learning to Drive A Car

Mr. Rahul Kumar Verma
Scholar
Yenepoya Institute of Technology, Moodbidri

Mr. Vijeth Pereira
Scholar
Yenepoya Institute of Technology, Moodbidri

***Abstract*—Self-driving cars offer many advantages. They have the ability to outperform human drivers in some circumstances and can be safer, as computers don't get tired nor lose focus. They offer great economic advantages as they remove the need for drivers. An end-to-end neural network to predict a car's steering actions on a highway is being analyzed. The inputs of the network are images from a single car-mounted camera. Neural networks have gained a lot of popularity from their successes in large-scale image classification benchmarks. They have since been applied in many different areas, often resulting in substantial improvements. The performance of a self-driving car system is crucial because errors can result in the death of a human being. The first aspect is the format of the input data. The second aspect is related to the temporal dependencies between consecutive inputs. A stacked frames approach which increases the performance of this network is being used. Training and testing of data is done on real-life datasets and qualitatively shown the importance of recovery cases as well as demonstrate that the standard metrics that are used to evaluate networks that are trained on datasets - accuracy, MCA, MAE, MSE do not necessarily accurately reflect a system's driving behaviour.**

Keywords- AI Artificial Intelligence, GTA V Grand Theft Auto Five, LSTM Long Short-Term Memory, MAE Mean Absolute Error, MCA Mean Class Accuracy, MSE Mean Squared Error, NN Neural Network

I. INTRODUCTION

The idea of self-driving cars has been around for longer but became a popular topic in the last few years, due to the continuous increase in computational power and the development of more intelligent algorithms. Lately, the control of more and more functions is being handled by the car itself and it is likely that the public will have access to fully self-driving cars in the near future.

Such self-driving cars are advantageous because they have the ability to outperform human drivers in some circumstances. A few examples are that self-driving cars can be safer than human-driven cars as computers don't get tired nor lose focus. Self-driving cars can reduce the amount of traffic jams by driving at optimal speeds, keeping the right distance from the previous car and not executing selfish manoeuvres. They can be more ecological by avoiding unnecessary accelerations. Perhaps the most persuading advantage is the economical gain: it removes the need for drivers. This allows transport companies to cut out driver costs and employees can work while being driven around, to give some examples.

Of course, there are several challenges that need to be overcome. Developing a self-driving car is a very complex and delicate process: the software is safety-critical and must be fail-safe since a single error can result in the death of a human being. It is difficult to guarantee the safety of a self-driving car, especially when it's running AI software which works in ways that are not 100% clear. Even in a perfectly controlled situation the car might face ethical choices. Another challenge is the need for an adequate legal framework, which is currently missing in most countries. These are just some examples of the difficulties that can arise in the development of autonomous vehicles.

In this thesis the author has developed an end-to-end AI algorithm to control a self-driving car, using only the images from a single car-mounted camera as input. To do so, he has used a neural network that predicts the actions of the car. Because this is a vast topic, he has focused mainly on a simplified version of the problem. They only control the steering of the car, while it is driving on a highway. This system is illustrated in Figure 1.1.

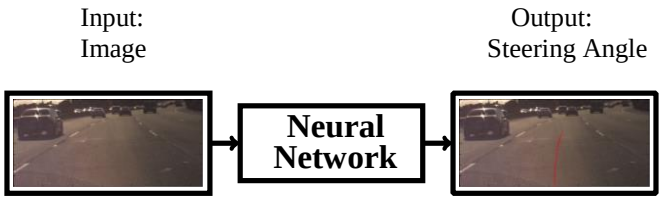

FIG 1.1 PROCESS FLOW OF THE SYSTEM.

There are good reasons to use neural networks for this. Artificial intelligence is on the rise. The continuous increase in hardware performance enables a broader public to use more computationally intensive machine learning algorithms, such as learning-based representations (deep learning) . This is promising because AI can, with supervised or unsupervised learning, learn by itself how to solve a problem. Compared to human-made solutions, this allows to solve existing problems in new ways or even allows to solve problems that were previously unsolved. Neural networks has gained a lot of popularity. They have since been applied in many different areas, often resulting in substantial improvements. Examples of this are Google Translate and Facebook using NNs to improve their services. Since the author is mapping images to steering actions, he uses convolutional neural networks. This is often done as a "convolutional neural network is the most popular algorithm for image cognition and video analysis".

Most state-of-the-art works use a mediated perception approach, which is based on object detection to make driving

decisions. A comparison between this approach and our end-to-end approach can be found in chapter 2. Little research is available about using end-to-end neural networks to directly map images to driving actions. Therefore, to achieve a system with a good performance, this research focuses on high-level aspects of the system. Our results can then be used in further research that focuses on smaller aspects, such as parameter sweeps, to improve performance.

The first aspect of this research is the format of the input data that is being fed to the network. The network is trained with supervised learning: i.e. train it on input images together with input steering wheel angle measurements. A look into the influence of the quantization granularity of the steering wheel's angle measurements onto the system's performance is done. Colour scheme of the input images are also compared based on different colour schemes and grayscale. This shows us to what extent the system can make use of the information that lies within the colour of the images.

The second aspect analyses the temporal information that can be found in successive input images. This amount of information could be significant because these images can be seen as the subsequent frames of a video, so there is information contained in such a sequence. Two techniques that can give the network more capabilities to utilize this information are being fertilized. The first technique consists of concatenating multiple subsequent images and feeding them to the network as a single input. This leads to an increased input size, but the architecture of the network remains the same. The second technique consists of inserting recurrent NN layers into the architectures that is being used here. By definition, recurrent layers can retain information between consecutive inputs and thus utilize the temporal information.

The final aspect is the origin of the data. If a simulator is sufficiently advanced and mimics the circumstances of the real world well enough, it can be used to train the neural networks and this would reduce the need for real-world data to train on. In a simulator it is very simple and cheap to gather data and the settings of this data are easy to change. This leads to bigger and more diverse datasets, which can be used to train more robust and better performing networks.

II. RELATED WORK

The related work regarding self-driving cars can be divided into two categories: mediated perception approaches and end-to-end approaches. The work is positioned with respect to these two categories.

A. Mediated Perception Approaches

Most state-of-the art systems use a mediated perception approach. These approaches rely on the detection and classification of surrounding objects such as traffic signs, cars, roads, buildings and pedestrians. They parse the entire scene into an internal map of the surroundings and use this to make driving decisions. A common practice in this type of approaches is that objects and other scene elements that are considered "relevant" need to be pre-defined. Another common characteristic of mediated perception approaches is their requirement of multiple sensors for reliable object detection and classification. Examples of these sensors include cameras, lasers and radar. Mediated approaches usually consist of two steps:

i) Detection of "relevant" visual elements, and
ii) Decision making based on those elements.

This has both advantages and disadvantages. On the one hand, a disadvantage is that it is possible that a designer fails to identify certain relevant objects. Moreover, it is also possible that useful information may get lost between the two steps. The objects that are to be detected are usually hand-picked. Because of this, certain detected objects may not be important for the decision making while other meaningful objects may not be detected and this may deteriorate the system's performance. On the other hand, an advantage is that this level of indirection makes it easier for the network to focus on certain details. For example, it is important to detect if a pedestrian has the intention to cross the street and a mediated perception approach can use a network that specializes in detecting this. But for an end-to-end network, it may be difficult to learn that it is important or to notice this type of situations. Different from mediated perception approaches, we focus on end-to-end systems where the network is given the task of identifying which visual elements are important for the task at hand. Therefore, no object or any other visual element needs to be pre-defined or hand-picked.

B. End-to-end Approaches

In end-to-end systems, images are directly mapped to driving decisions with the use of machine learning algorithms. The system here uses this approach. Sometimes multiple cameras are used to create recovery cases or a simple real-world simulator. As said earlier, a disadvantage of end-to-end approaches is that they lack a second processing step or controller that makes decisions. Because of this, the system does not keep track of the bigger picture. This makes it difficult to teach the network certain specific things, for example to abruptly avoid any children that run in front of the car. This is difficult because unless the dataset is artificially created, there are few such situations in the dataset while many different situations should be present to create a robust system. Another disadvantage is that the driving behaviour in the training data has a direct influence as the network learns this imperfect driving style, such as driving too close to the right lane markings. If possible, the images and measurements should be very carefully selected or corrected. This is less of a problem for mediated perception approaches than it is for end-to-end approaches.

C. Other Related Problems

An additional motivation for the use of end-to-end systems can be found in related problems. Experiment on to flying a drone through a room is considered [2]. They stress that using pretrained networks is a good alternative to learning from scratch for end-to-end networks. It saves training time and requires less data because it is less prone to over fitting. They also stress that training LSTM networks using a limited time window produces better performance than when training it on all previous input samples. Moreover, it has been clearly indicated that there is a clear trend in which LSTM networks outperform standard feed forward networks. Another

observation is that recovery cases have a big impact on performance. Following the same directions, it was demonstrated that it is possible for networks trained on simulation data to be generalized to the real world. Taking these works into account, it is plausible to assume that many observations and conclusions drawn from the autonomous drone problem, such as the generalization of simulators and the better performance of LSTM networks, also hold for the problem. Finally, motivation for incorporating inter-frame dependencies is found in language modeling. Just like different words in a sentence are also related in order to convey meaning, the input images are temporally dependent because they are consecutive frames of a video. It has been proved that LSTMs can outperform standard feed forward neural networks on such tasks.

III. METHODOLOGY

The basic system set-up is as follows: images from the camera are fed into the network and the network predicts the steering angle from these image(s). During training, the steering wheel angle measurements, i.e. annotations, are also fed to the network. An illustration of our system is given in Fig 1.1

A. Network Architectures

Throughout the experiments, the neural networks varies in two areas: their main architecture and their output layer. The main architecture is a variation of either the NVIDIA , AlexNet or VGG19 architecture. In the Alexnet architecture, the author removed the dropout of the final two dense layers and reduced their sizes to 500 and 200 neurons as this resulted in better performance. The output layer of the network depends on its type (regression or classification) and, for a classification network, on the amount of classes. In this analysis the experiments are conducted with both, classification and regression, types. For the case of the classification type, the steering angle measurements have been quantized into discrete values, which represents the class labels. This quantization is needed as input when training a classifier network and allows to balance the data through sample weighting. This weighting acts as a coefficient for the network's learning rate for each sample. A sample's weight is directly related to the class that it belongs to when quantized. These class weights are defined as 1 divided by the amount of samples in the training set that belong to that class, multiplied by a constant so that the smallest class weight is equal to 1. Sample weighting is done for both classifier networks and regression networks. Note that for the latter, the class is used, to which the continuous value would be mapped. This weighting is done to ensure that the network is equally trained on all classes, in the hope that it learns to handle all these different situations well. Otherwise, the network might be biased toward a certain class.

B. Dataset

Training and evaluation of different networks is conducted on the Comma.ai dataset, which consists of 7.25 hours of driving, most of which is done on highways and during daytime. Images are captured at 20 Hz which results in approximately 552,000 images. Discarded the few sequences that were made during the night due to their high imbalance when compared to those captured during daytime. In addition, in order to focus on sequences with continuous / uninterrupted driving, the author has limited himself to only considering images that were captured while driving on highways. The remaining data is then split into two mutually exclusive partitions: a training set of 161,500 images and a validation set of 10,700 images. This is done on a random per-file basis to ensure independence between training and validation and to ensure that both sets contain various traffic and weather conditions. These two datasets are used in all conducted experiments.

C. Performance Metric

The performance of the networks is predicted using the following performance metrics: accuracy, mean class accuracy (MCA), mean absolute error (MAE) and mean squared error (MSE) metrics. Conclusions based on the MSE metric, since it allows to take the magnitude of the error into account and assign a higher loss to larger errors than MAE does. This is desirable since this may lead to better driving behaviour, as the author assumes that it is easier for the system to recover from many small mistakes than from a few big mistakes. A large prediction error could result in a big sudden change of the steering wheel angle. For example, larger errors create dangerous situations as the car might swerve onto an adjacent lane or go off-road.

For every metric individually, the best performance over all epochs is chosen. These values are then compared between networks and the best network is selected based on the MSE metric. Note that the absolute performances are of relatively low importance to and that more interest is on the relative performances between the different network variants in the experiments. After analyzing the high-level aspects of the presented problem, better performance can later be achieved by optimizing around the results of those experiments.

D. Input Data Format

1. Quantization granularity

In this first experiment, the author looks into the influence that the specifications of the class quantization procedure have on the system's performance. These specifications consist of the amount of classes and

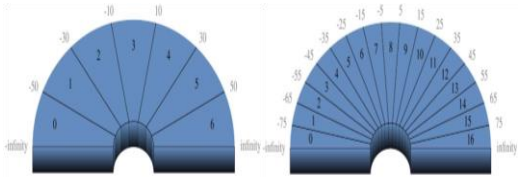

FIG 3.1 MAPPING OF ANGLE MEASUREMENTS FROM CONTINUOUS VALUES (OUTSIDE) TO DISCRETE CLASS-LABELS (INSIDE) FOR 7 AND 17 CLASSES, RESPECTIVELY THE MAPPING FROM THE INPUT RANGE TO THESE CLASSES.

comparison of classifier networks with varying degrees of input measurement granularity is done. Also they are compared to regression networks, which can be seen as having infinitely many classes, although using a different loss

function. It is plausible that the granularity has a significant impact on the system's performance. For metrics such as global accuracy and mean class accuracy, this is obvious since it is more difficult to choose the right class for a fine quantization configuration that has a higher number of classes. Coarse-grained classes however have a bigger quantization error and this influences the magnitude of the error. For metrics where the magnitude of the error is taken into account, such as MSE and MAE, it is possible that configurations with many fine-grained classes will perform better. Because classifier networks are trained with categorical cross entropy as loss function, they are expected to perform well when compared using class accuracy as metric. On the other hand, regression networks will probably outperform the classifier networks on metrics that take the error distance into account, as their loss function (e.g. MSE) also uses the error distance.

An experiment is being conducted by comparing a coarse-grained quantization scheme with 7 classes and a fine-grained scheme with 17 classes. Both classifier and regression networks are done.

All of these networks are tested on the three architectures previously explained and evaluated. The difference between 7 and 17 for regression is in the class weighting. Each sample is given a weight based on their relative occurrences in 7 or 17 classes. (Similar to class weighting for the classification networks.) Also, to be able to compare regression vs. classification, the predicted regression outputs were discretized into 7 and 17 classes to calculate MCA in the same way this happened for the classification networks. The results of this experiment are found in Figures 3.2 through 3.5. Several observations can be made. First, it is logical that the coarse-grained scheme scores better on the accuracy and MCA metric. More importantly, we see that regression networks significantly outperform classifier networks on the MAE and MSE metrics, which have been discussed and concluded to be the most important metrics.

This aligns with authors expectations, since regression networks have a loss function that takes the error magnitude into account. Finally, it's been noticed that class weighting does not have a significant impact on the performance of regression networks. A possible explanation is that this is due to their loss function, which also takes the error magnitude into account. Samples which are less common generally will get a higher loss, as their steering angle is mostly predicted a lot worse than common samples.

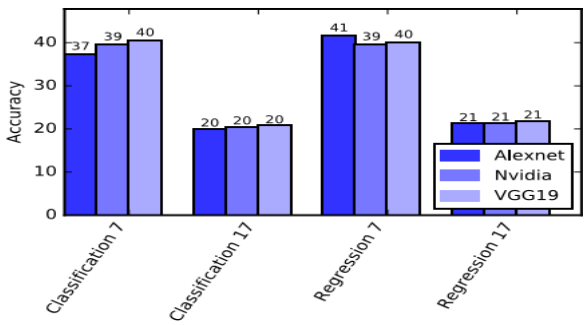

FIG 3.2 CLASSIFICATION ACCURACY OF THE GRANULARITY EXPERIMENT.

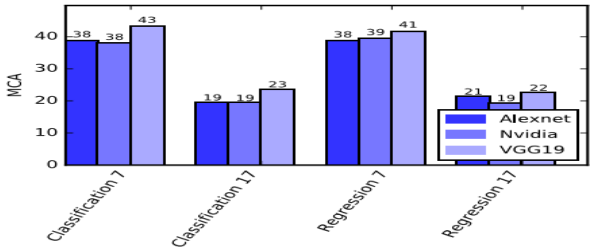

FIG 3.3 MEAN CLASS ACCURACY (MCA) OF THE GRANULARITY EXPERIMENT.

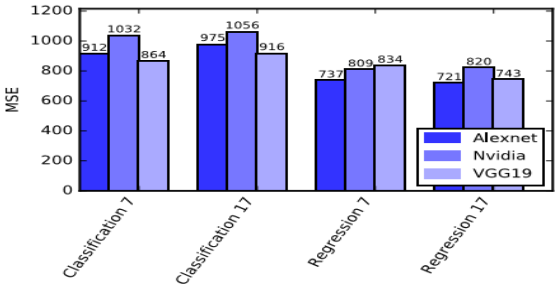

FIG 3.4 MEAN SQUARED ERROR (MSE) OF THE GRANULARITY EXPERIMENT.

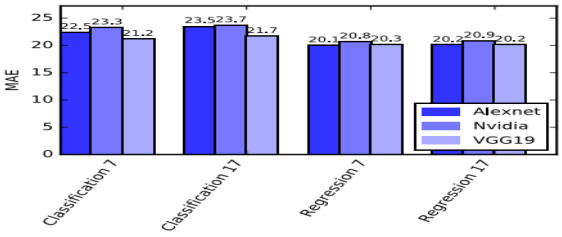

FIG 3.5 MEAN ABSOLUTE ERROR (MAE) OF THE GRANULARITY EXPERIMENT.

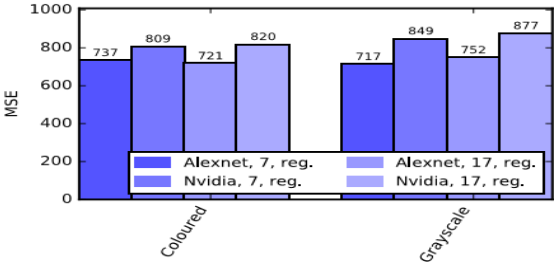

FIG 3.6 MEAN SQUARE ERROR (MSE) OF THE COLOUR EXPERIMENT.

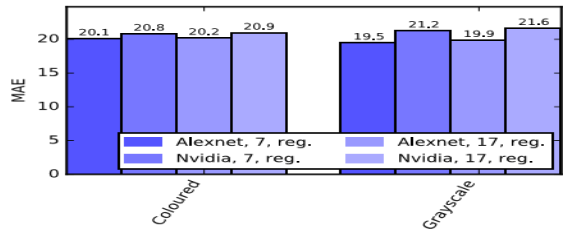

FIG 3.7 MEAN ABSOLUTE ERROR (MAE) OF THE COLOUR EXPERIMENT.

E. Incorporating Temporal Dependencies

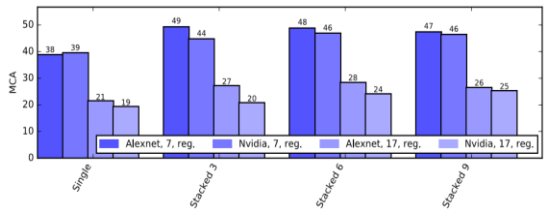

FIG 3.8 MEAN CLASS ACCURACY (MCA) OF THE STACKING EXPERIMENT.

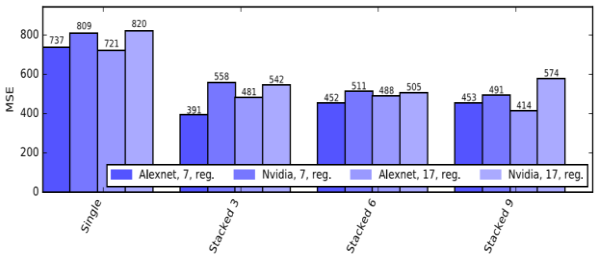

FIG 3.9 MEAN SQUARE ERROR (MSE) OF THE STACKING EXPERIMENT.

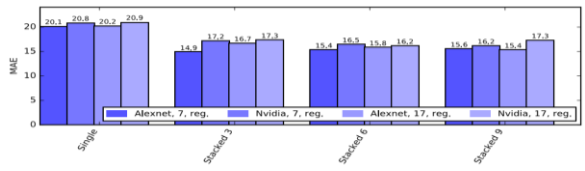

FIG 3.10 MEAN ABSOLUTE ERROR (MAE) OF THE STACKING EXPERIMENT.

Image Colour Scheme:

Investigation on to what extent the system can exploit the colour information that is present in the input images. It was started off by comparing coloured images to grayscale ones. The images from the dataset already have the RGB colour scheme. Since the previous network proved regression networks to outperform classifier networks in the problem at hand, the author focused this experiment on regression networks only.

The results from this experiment can be found in Figures 3.6 and 3.7. From these results, it can be observed that there is no significant difference in performance between networks that use coloured and grayscale images as input. This suggests that, for the task at hand, the system was not able to take much advantage of the colour information. Therefore, it is not worthwhile to investigate this aspect any further and compare different colour schemes.

IV. Application of Simulated Data

A last aspect investigated is the origin of the data. Up until now, training and evaluation of the system was done using real-world datasets. Here the author looks into the advantages of a simulator over a real-world dataset and the uses of such a simulator. Research on the impact of recovery cases on a network's performance and verify if the performance metrics that are typically used are a good indicator of a network's real driving behaviour.

A simulator brings many advantages. Some examples are that data gathering is easy, cheap and can be automated. Recovery cases can easily be included in the dataset. Infrequently occurring situations can be simulated and added to the dataset. Driving conditions such as the weather and traffic can be set as desired. Testing in simulators is safe, cheap and easy.

A. Udacity Simulator

First the Udacity simulator is used to generate three datasets. This simulator is very minimalistic and has no other cars, pedestrians, or complex traffic situations. Only simple test-tracks are implemented. The first dataset is gathered by manually driving around the first test-track in the simulator. The second dataset consists of recovery cases only. It is gathered by diverging from the road, after which the recovery to the middle of the road is recorded. This process is repeated many times to get a sufficiently large dataset. A third validation dataset is gathered by driving around the track in the same way as with the first dataset. For the following experiments, the NVIDIA architecture with a regression output is used and no sample weighting is applied during training.

1. Training on Simulated Data

The first experiment tests the performance of a network trained solely on the first dataset. After training, the best epoch is selected based on MCA. The metrics are comparable to other runs on the real dataset. As the confusion matrix has a dense diagonal, good real-time driving performance is expected. When driving in the simulator, the network starts off quite well and stays nicely in the middle of the road. When it encounters a more difficult sharp turn, the network slightly miss-predicts some frames. The car deviates from the middle of the road and is not able to recover from its miss-predictions, eventually going completely off-track. Conclusion is made that despite promising performance on the traditional metrics, the system fails to keep the car on the road.

2. Recovery Cases

The second experiment evaluates the influence of adding recovery data. First a new network is trained solely on the recovery dataset. The confusion matrix together with its MCA. As can be expected, the confusion matrix is focused on steering sharply to the left or right. As it does not look very promising and the MCA is very low, it is expected this network will not perform very well during real-time driving. Despite the low performance on these metrics, the network manages to keep the car on track. The car however does not stay exactly in the middle of the road. Instead, it diverts from the center of the road, after which it recovers back towards the middle. It then diverts towards the other side and back to the middle again, and so on. The car thus wobbles softly during the straight parts of the track, but handles the sharp turns surprisingly well. A third network is trained on both datasets and has a confusion matrix similar to the first network. In the simulator, it performs quite well, driving smoothly in the middle of the lane on the straight parts as well as in sharp turns. Hence it's concluded that recovery cases have a significant impact on the system's driving behaviour. By adding these recovery cases, the driving performance of the system is improved while its performance

on metrics deteriorates. This again suggests that the standard metrics might not be a good tool to accurately assess a network's driving behaviour.

FIG 3.11 THE CONFUSION MATRIX

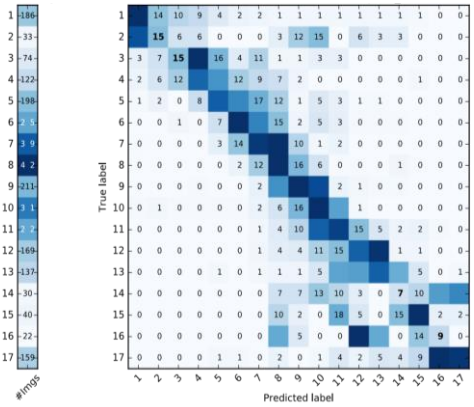

FIG 3.12 CONFUSION MATRIX FOR NN TRAINED WITHOUT RECOVERY DATA .MEAN CLASS ACCURACY (MCA) IS 32.5%

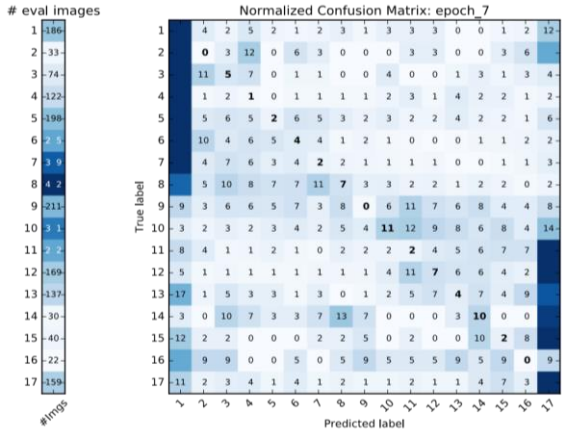

FIG 3.13 CONFUSION MATRIX FOR NN TRAINED WITH ONLY RECOVERY DATA. MEAN CLASS ACCURACY (MCA) IS 9.9%.

B. GTA V Simulator

As an extension to the simplistic Udacity, GTA V is integrated as a more realistic simulator platform. Next to being nearly photo-realistic, GTA V provides a big driving playground of a vast 126 km^2 with various lighting and weather conditions, many different cars, and driver's traffic scenes. A big dataset with 42 hours of driving is available. This data also includes recovery cases. This dataset is composed by 600 k images split into 430k training images and 58k validation images. A NVIDIA and an AlexNet regression network, as described above, are trained on the dataset with sample weights based on 17 classes. The network shows performance metrics similar to the NVIDIA regression network trained on the real-world dataset. Evaluation of real-time driving performance on an easy, nonurban road with clear lane markings is being done. The network performs quite well and stays around the center of the lane. When approaching a road with vague lane markings, such as a small bridge, the car deviates towards the opposite lane (Figure 3.16 middle). When it reaches a three-way crossing (Figure 3.17 bottom), the network cannot decide whether to go left or right, as it was equally trained on both cases. Because of this, it drives straight and goes off-road. In an urban environment, the network struggles with the same problem, resulting in poor real-time performance.

Again, observations from this experiment suggest that current metrics are not always representative for real-time driving performance. In this regard, further research must be conducted towards developing new performance metrics and setting up automatic testing environments that are able to match performance at training time and performance during real-time driving. Some possible metrics could be distance from the middle of the lane, smoothness of driving (penalizing abrupt braking or turning), or a metric based on how long the car stays on the road without accidents.

FIG 3.15 SEQUENCE: ROAD COVERED BY SHADOWS

FIG 3.16 SEQUENCE: ROAD WITH VAGUELY MARKED LINES

FIG 3.17 SEQUENCE: THREE-WAY ROAD CROSSING.SOME VIDEO SEQUENCES SHOWING THE EVALUATED MODEL DRIVING IN THE GTA V SIMULATOR

V. CONCLUSION

In this paper, the author has analyzed an end-to-end neural network to predict the steering actions of a car on a highway from an input captured by a single car-mounted camera. This analysis covered several high-level aspects of the neural network. These aspects were the format of the input data, the temporal dependencies between consecutive inputs and the application of simulated data.

Regarding the first aspect, it was showed that the amount of classes of a classifier does not seem to have a big influence on

the performance and that regression networks outperform classifier networks. This is likely due to the nature of their loss function which is similar to the metrics used for evaluation, takes the magnitude of a prediction error into account. Moreover the author showed that, for the task at hand, there is no major difference between networks that use coloured images and ones that use grayscale images. Middle of the lane, smoothness of driving (penalizing abrupt braking or turning), or a metric based on how long the car stays on the road without accidents. Regarding the second aspect, while the author was unsuccessful in improving performance by implementing LSTM layers, the stacked frames approach delivered good results. By feeding the network 3 concatenated images, he got a significant decrease of 30% in mean square error (MSE). Further increasing the amount of concatenated images only brought diminishing returns that did not outweigh the drawbacks. Regarding the third aspect, the author was able to gather simulated data and train networks that have a performance comparable to the networks that they trained on real-life datasets. They have qualitatively shown the importance of recovery cases. They also qualitatively showed that the standard metrics that are used to evaluate networks that are trained on datasets accuracy, MCA, MAE, MSE - do not necessarily reflect a system's driving behaviour. Also its been shown that a promising confusion matrix may result in poor driving behaviour while a very ill-looking confusion matrix may result in good driving behaviour. A structured framework is needed that allows to quantitatively measure more meaningful metrics.

REFERENCES

[1] From Pixels to Actions: Learning to Drive a Car with Deep Neural Networks 2018 IEEE Winter Conference on Applications of Computer Vision.

[2] K. Kelchtermans and T. Tuytelaars. How hard is it to cross the room?–training (recurrent) neural networks to steer a uav.arXiv preprint arXiv:1702.07600, 2017.