https://doi.org/10.2352/ISSN.2470-1173.2021.17.AVM-171

GG-Net: Gaze Guided Network for Self-driving Cars

M.Abdelkarim; mohd.abdalkarim@gmail.com; Cairo University; Cairo, Egypt
M.K. Abbas; muhammad.abbass@outlook.com; Cairo University; Cairo, Egypt
Alaa Osama; alaa.oabdelfattah@outlook.com; Cairo University; Cairo, Egypt
Dalia Anwar; dalia.anwar112@gmail.com; Cairo University; Cairo, Egypt
Mostafa Azzam; mostafaazzam74@gmail.com; Cairo University; Cairo, Egypt
M.Abdelalim; MohammedAbdelalim14@gmail.com; Cairo University; Cairo, Egypt
H.Mostafa; hmostafa@uwaterloo.ca; Associate Professor at Zewail City of Science and Technology; Cairo, Egypt
Samah El-Tantawy; samah.elshafiey@gmail.com; Assistant Professor at Faculty of Engineering, Cairo University; Cairo, Egypt
Ibrahim Sobh; ibrahim.sobh@valeo.com; Valeo Group; Cairo, Egypt

Abstract

Imitation learning is used massively in autonomous driving for training networks to predict steering commands from frames using annotated data collected by an expert driver. Believing that the frames taken from a front-facing camera are completely mimicking the driver's eyes raises the question of how eyes and the complex human vision system attention mechanisms perceive the scene. This paper proposes the idea of incorporating eye gaze information with the frames into an end-to-end deep neural network in the lane-following task. The proposed novel architecture, GG-Net, is composed of a spatial transformer network (STN), and a multitask network to predict steering angle as well as the gaze map for the input frame. The experimental results of this architecture show a great improvement in steering angle prediction accuracy of 36% over the baseline with inference time of 0.015 seconds per frame (66 fps) using NVIDIA K80 GPU enabling the proposed model to operate in real-time. We argue that incorporating gaze maps enhances the model generalization capability to the unseen environments. Additionally, a novel course-steering angle conversion algorithm with a complementing mathematical proof is proposed.

Introduction

Nearly 1.25 million people die in road crashes each year, on average 3,424 deaths a day. Up to 90% of accidents are due to human factor [14]. As a result, Self-driving vehicles have taken a huge interest worldwide [9]. The problem of self-driving cars can be approached by a modular approach or an end-to-end approach.

Imitation learning is mimicking the behavior of an expert which in the case of self-driving cars is the human driver. Humans learn to drive by watching other people drive and then try to observe, learn, and replicate their actions in similar situations faced by the expert. Practically, the end-to-end systems have been widely used in this technique as it optimizes all process steps simultaneously without the need for dividing it into sub-problems [3].

Behavioral cloning is the simplest way to conduct imitation learning where an expert provides trajectories in terms of state-action pairs, and imitation learning can be conducted through supervised learning. Consequently, this powerful end-to-end approach has been used in tasks such as lane and road following. By using a small amount of training data labeled with steering commands, the system learned the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, path planning, and control [3].

Moreover, while a beginner human driver learns from an expert, the beginner not only observes the expert's steering actions but also observes his full-body actions, including head and eye movements, the field of attention in different situations, emotional status, besides the verbal instructions which are given. It is interesting to experiment giving the end-to-end model access to more information about the human driver status during driving and observe the improvement in the learning process.

Furthermore, among the human senses, the eye is the most human attention and intent-expressing unit [16]. While humans are able to perceive the environment mainly by visual cues, computer vision based on camera perception alone is not as efficient [18]. We argue that incorporating the human gaze behavior as an additional source of information during the training process helps the model to better recognize the task critical objects to focus on them, which helps "humanizing" the self-driving cars.

In this paper, the gaze information is incorporated with the front-facing camera frames in a multitasking deep neural network with an attention mechanism -which is STN- added, to improve steering angle estimation and fulfilling the real-time autonomous cars system requirements. The rest of the paper is organized as follows: The second section presents the studies that are closely related to that topic. The third section discusses and analyzes the dataset. The fourth section shows the details of multiple experiments done to optimize gaze information utilization. The fifth section discusses the results of the experiments. And the last section shows the ability of the model to generalize to different unseen environments through a real-time on-road test demonstration.

Related Work

Imitation learning, in theory, can leverage data from large fleets of human-driven cars. Behavioral cloning, in particular, has been successfully used to learn simple visuomotor policies end-to-end, but it fails when scaling to the full spectrum of driving behaviors. Some well-known limitations of behavioral cloning are dataset bias, over-fitting, and generalization issues [7].

When intelligent agents learn visuomotor behaviors from human demonstrations, they may benefit from knowing where the human is allocating visual attention, which can be inferred from

the eye gaze. Consequently, modeling human visual attention and guiding the learning agent by a learned attention model could lead to significant improvement in task performance. The paper [22] proves this idea by collecting high-quality human action and gaze data while playing Atari games, training a network to predict the human visual attention, and feeding the network's output to another network that predicts the human actions. Furthermore, another work used the idea of attention mechanism to make an attention guided convolution neural network (CNN) to localize the important region in an input image which improved the accuracy in the field of disease classification on chest X-ray images [10].

Incorporating gaze information in self-driving cars is mostly used in advanced driver assistant systems (ADAS) [11][15]. Two papers published in late 2019 discussed the incorporation of gaze information into an end-to-end self-driving model [12][6]. They used two different approaches to incorporate eye gaze into the training process. In particular, their work used generative adversarial network (GAN) [12] to estimate gaze maps, then fed the estimated gaze map stacked with the front-facing camera frame to a deep neural network (DNN). The work described in [6] used the generated gaze map -by the GAN- as a mask to give weighted dropout probability with spatial dependence to each region in the frame. Both of these experiments showed an improvement in the system performance. However, GAN is known to be computationally expensive, making it challenging to be used in real-time self-driving car applications. Moreover, these experiments were only conducted in simulation and the performance was not tested on a dataset containing real driving frames.

Dataset

In this section, the dataset used for training, gaze-map generation, and a novel course-steering angle conversion algorithm used for labeling the dataset are presented.

Dataset Selection

In this work, a dataset containing front-facing camera frames labeled with the driver's gaze position information and the steering angle is needed. "DR(eye)VE" dataset [1] is currently the largest public driving dataset including gaze information in automotive settings. It consists of 74 video sequences with a total of 555,000 frames, covering different weather conditions (sunny, cloudy, and rainy), different lighting, and different scenarios (countryside, highway, and downtown). Each frame is labeled with: speed, course angle, and x-y eye gaze position. Videos were recorded with a roof-mounted camera (1080p, 25fps).
The following experiments are based on the lane-following task at a constant speed. As a result, the chosen videos from the DR(eye)VE dataset are from the countryside and highway contexts, as there are no pedestrians crossing roads, no crossroads, and no multi-paths in these contexts. For this work, 12 videos were chosen to contain about 85,000 frames covering all weather conditions. These videos were split into 80% for training and 20% for testing.

Gaze-map Generation

The work in [17] introduced a computer vision model that is able to replicate the human attention behavior during the driving task using DR(eye)VE dataset. The paper argues that the act of driving combines complex attention mechanisms guided by the driver's past experience, short reactive times, and strong contextual constraints. Therefore, very little information is needed to drive if guided by a strong focus of attention (FoA) on a limited set of targets and purposes. In addition, the work of [17] introduced a multi-branch deep neural network (DNN) model that aims at predicting these targets "Gaze-maps".

This multi-branch model was used to generate the corresponding gaze maps for the selected dataset frames. However, this multi-branch DNN has a relatively large inference time as it is composed of three different branches, each of which has its own set of parameters. Afterward, the predictions from the three branches are summed to obtain the final gaze map. In particular, the three branches are RGB Image, optical flow, and semantic segmentation branches. The semantic segmentation branch uses the network in [21], which is accurate but too large with 134 million parameters taking approximately 23 sec./frame to make a prediction. (All the experiments in this paper are conducted on NVIDIA K80 GPU).

In order to accelerate gaze-maps generation, the semantic segmentation network was replaced by [23] after modifying its output to be as close as possible to the output of [21]. Consequently, this modification reduced the inference time required to obtain semantic segmentation frames from 23 sec./frame to 3 sec./frame.
To speed up the generation of the gaze maps from the multi-branch DNN, and due to the fact that eye-gaze movements across frames are slow relative to the high frame rate of the videos in the dataset (25 fps), pixel-wise linear interpolation across frames in time was used. Only one frame every 7 frames was obtained using the multi-branch DNN, while the frames in between were obtained using interpolation. The obtained frames using interpolation had less than 1% error from the ground truth gaze-maps generated by the multi-branch network. Using this approach, the time to generate gaze-map was reduced by 10X of the author's original setup inference time.

Course-Steering Angle Conversion

There is only one drawback in the DR(eye)VE dataset, which is the absence of the steering angle labels. Instead, the dataset is labeled with "course angle", which is the angle between the head of the car and the North direction. There is no direct approach for the conversion between steering and course angles. Accordingly, a conversion method is developed and mathematically proved:

$$\Theta_s = SR \times \tan^{-1}\left(\frac{(\theta_c^1 - \theta_c^2) \times L}{v \times t}\right) \tag{1}$$

where θ_s is steering wheel angle, t is the inverse of the frame rate of the dataset = 1/25, v is the car's velocity at each frame, L is wheelbase length ≈ 2.6, SR is steering ratio ≈ 17, and θ_c is course angle.

Using discrete derivative and motion equations, a relation presented in 1 which relates every two consecutive coarse angles to the corresponding approximated steering angle was derived mathematically. For detailed mathematical proof, please refer to the appendix. Using 1, the steering angles for the chosen dataset frames were generated.

Experimentation

In this section, the baseline architecture used as well as multiple experiments utilizing different approaches and architectures are described, analyzed, and compared. As described in the introduction, the work presented by [3] is replicated and the Pilot-Net model is set as a baseline for the experiments. Pilot-Net was trained on the dataset discussed in the "Dataset" section, with the camera frame and steering angle only without the gaze information to compare the results and the improvement achieved after adding the gaze information.

Fusion Architecture

Fusion techniques include middle and late fusion. In fusion networks, there are multiple inputs. Features are extracted from each input, then fused to make a prediction. The point at which fusion takes place defines the type of fusion. In particular, if the fusion occurs directly after extracting features, then it is a middle fusion. And, if fusion between features occurs after passing the extracted features through more than one fully connected layer, it is called late fusion. Middle fusion was conducted after 5 layers of Pilot-Net as shown in Fig.1 to give the network a chance to extract features from the input. Due to the presence of two inputs: the RGB frame and the gaze map, the network consisted of two branches for each one. We started by using the convolution layers (first 5 layers) in Pilot-Net as feature extractors for both the RGB frame and the gaze map. Afterward, we decreased the number of layers and filters in the gaze map branch. Because we used the 1D grayscale gaze heat map, which contains much fewer features to be extracted than the RGB frame. Then, the extracted features are fused to one feature vector and go through a series of fully-connected (FC) layers ending with the final prediction. This architecture achieved an improvement of 30% over the baseline. Consequently, this proved that gaze map incorporation improved the steering angle prediction accuracy. However, the inference time was too large as it had the Gaze Net inference time added.

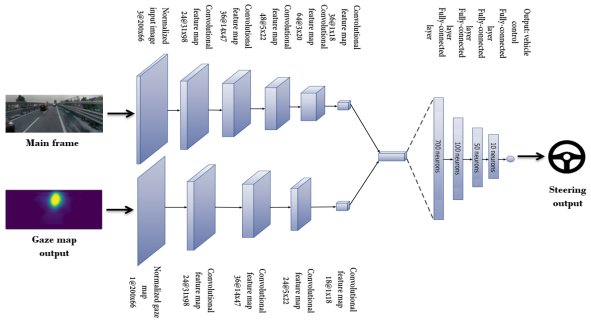


Figure 1. *Fusion Network Architecture*

STN-based Architecture

Spatial Transformer Network (STN) was introduced by Google DeepMind [20]. It was mainly directed to image classification problems. And to our knowledge, this is the first time to use the STN in a regression problem in the self-driving cars (SDC) field. STN is added to the neural network model to enable it to transform the input image using 6 transformations: cropping, isotropic scaling, rotation, etc. The STN structure consists of 3 main parts: localization net, grid generator, and sampler. The localization net takes the input frame then outputs the transformation parameters to be applied to the input frame by the two other components, which make the output transformed frame much better and easier for the rest of the network to deal with. In particular, the transformations focus on the important parts of the frame. Therefore, we thought that adding the STN to the SDC end-to-end model with the help of the information extracted from the gaze maps will make the network able to treat the frame the same way a human does. In other words, it will make the network focus on the important regions only which is a form of "attention." The localization network final design in our architecture is made of a series of alternating max-pooling and convolution layers **(Max pooling** $3@100\times33$**, Conv. layer** $24@96\times29$**, Max pooling** $24@48\times10$**, Conv. layer** $36@44\times10$**, Max pooling** $36@21\times5$**, Conv. layer** $48@10\times2$). The input of the localization network was used to inject the gaze information by inputting the gaze map frame, then using the output of the localization network which is reduced to have 3 transformations only (cropping, translation, and isotropic scaling) to transform the main frame. Afterward, the output is resized to be (200×66) then fed to Pilot-Net as shown in Fig.2. The whole network was trained end-to-end. As expected, the STN learned to focus on and crop the street, and specifically on the lanes. It tends to shift right a little bit to get the right lane as it is more important than the left lane, nearer, and more clear in most cases in the dataset. This architecture achieved an improvement of 20.8% over the baseline. However, the inference time was too large as it had the Gaze Net inference time added like the previous architecture.

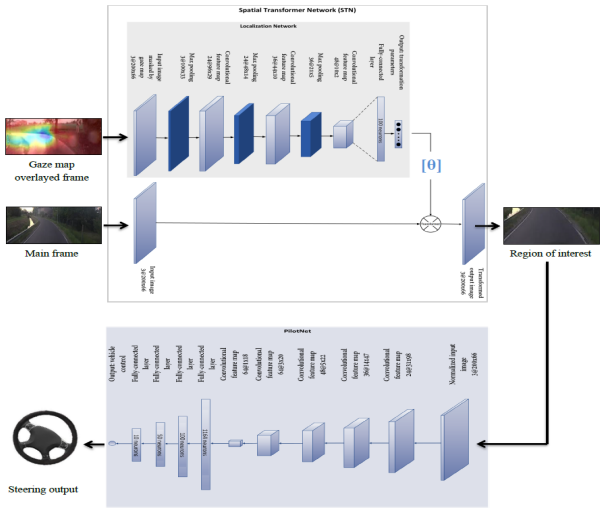


Figure 2. *STN-based Architecture*

MTL Architecture

The Multi-task learning (MTL) approach is a successful approach in the field of natural language processing (NLP) [8] and speech recognition [19], but most importantly computer vision [13] which is the main use of interest in this work and many more applications.

In MTL, more than one loss function for various tasks can be effectively optimized, as long as these tasks are somehow

correlated. In addition, the chance of getting an auxiliary task will help to improve the optimization process and hence, improving the primary task performance [5].
In this work, two tasks are used: predicting steering angle and predicting gaze map. In the MTL, the hard parameter sharing technique is used in this approach [4], which means some features are extracted by shared layers between all tasks, which is the backbone of the network. Afterward, we feed the extracted features to the heads of the network which represent task-specific layers. This kind of sharing reduces the risk of over-fitting as [2] concludes.

The proposed multi-task architecture idea is to use Pilot-Net as the main block -making steering angle prediction is the primary task- and, another branch emerges after the first 4 layers of Pilot-Net to form a decoder-like network to predict and construct the gaze map **(Deconv.** $36@47 \times 14$**, Conv.** $36@47 \times 14$**, Deconv.** $24@98 \times 31$**, Conv.** $24@98 \times 31$**, Deconv.** $3@200 \times 66$**, Conv.** $3@200 \times 66$**)**. Furthermore, each task has its own defined loss function: steering branch loss function is the root mean squared error (RMSE) and the gaze map branch is the pixel-wise root mean squared error (RMSE) between the ground truth gaze map and the predicted gaze map. Consequently, the overall system loss function is a weighted sum of losses with the steering branch having full weight, and the gaze branch multiplied by 0.1 to give the steering angle task a higher loss value to force the system to focus on this task more than the other auxiliary task. The MTL architecture is illustrated in Fig.3.

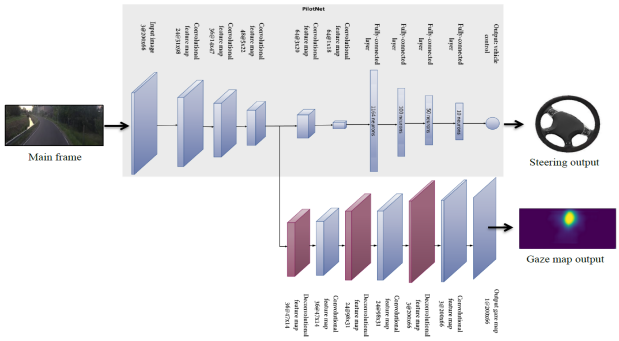


Figure 3. *Multi-task Network Architecture*

GG-Net (STN + MTL) Architecture

In the proposed architecture shown in Fig.4, the strengths of the two previously discussed architectures are combined together in one architecture: STN with its ability to focus on specific regions of the frame, and the MTL with its very small delay as it does not need to have Gaze Net before it to predict the gaze map. Furthermore, three forms are experimented: the STN is set to include 1, 2, and 3 parallel transformer layers on the input frame in 3 different experiments. In particular, it can be regarded as stacking 3 STN in parallel with the same input frame and 3 different output frames which gives the network more degrees of freedom. Afterward, the output images are stacked into a single frame, then resized to (200×60). Resizing is done so that the STN output image is in the same input size as the MTL network discussed in the "MTL Architecture" section as shown in Fig.4. After training, in the 1-transformer STN trial with multitask added, it was hard for the network to focus only on one region in the input frame because each one of the 2 tasks pushed the STN to focus on a different region of the frame, the steering branch pushed the STN to focus on the right lane as discussed before, but the gaze branch pushed it to focus on the far end of the road to get the gaze position as it is mostly at the far end of the road. Consequently, 1 transformer is not enough, paving the way for the 2-transformer branches trial which gave the network one more degree of freedom. It is noticeable that each task of the two tasks guide one of the 2 transformer branches to focus on its preferred region of interest, so the network learned to focus on the right lane and the far end of the street. Using the same idea of increasing the degrees of freedom, the 3-transformer branches STN was conducted. After training, the network learned to focus on the right lane and the far end of the street like the 2-transformers network, in addition to the third transformation which focused on the left lane which is less important than the right lane. The different transformations for each experiment are illustrated in Fig.5.

Results Analysis

From the previous analysis, it is clear that the GG-Net makes very satisfying predictions and interesting behavior. Moreover, it learned to detect the far end of the street like what a human driver does which increased the accuracy of the steering angle prediction and improved the generalization ability.

Besides the very high accuracy improvement, the inference time was very reasonable which is counted as another advantage of the GG-Net final architecture. The previous architectures like the one in [12] have the network that generates the gaze maps put in series before the model, which takes the input frame and generates the gaze map which is used by the network as an input. Because the inference time of the DR(eye)VE Gaze Net is very large as discussed in the "Gaze-map Generation" subsection -approximately 21 seconds/frame-, the presence of the Gaze Net was a problem eliminating the ability to operate in real-time. However, the MTL architecture replaces the Gaze Net completely, and its inference time is 0.0028 seconds/frame. For visualization of the model output, please refer to the video demo at **https://youtu.be/2I1rOys-Cc0**.

The proposed approach that incorporates gaze information and camera frames by combining STN and MTL makes an improvement of 32.37% over the baseline for the 2-transformers network and makes an improvement of **36.18%** over the baseline for the 3-transformers network. Moreover, it also has inference time of **0.015 seconds/frame** which makes a frame rate of **66 frames/sec**. Thus, this enables the model to be used in real-life systems. The results of the discussed architecture are summarized in Table 1.

On-Road Test

To ensure the ability of the GG-Net to generalize well in real unseen environments, a real-time road-test in Cairo, Egypt was conducted by fixing a front-facing camera on the rooftop of the car, then sending the frames to a laptop inside the car having the final trained model running on its GPU. We used an online visualization of the predicted steering angle running on the laptop's screen and visually compared it with the real steering angle made by an expert driver.

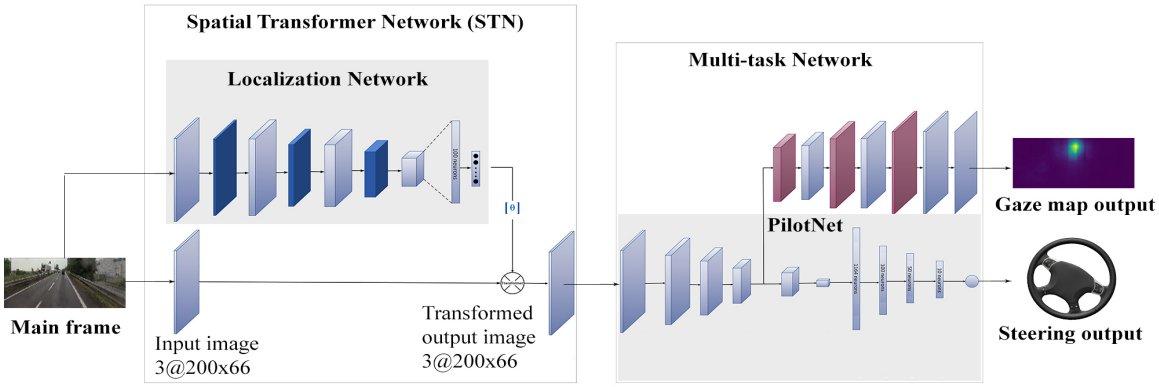


Figure 4. *GG-Net (STN+MTL) Architecture. The main frame is fed to the localization network to acquire the transformation parameters to be applied to it. The main frame and the transformation parameters from the localization network are fed to the transformer. The transformed output image is fed to Pilot-Net and Multitask Network to predict the steering commands and the gaze maps respectively.*

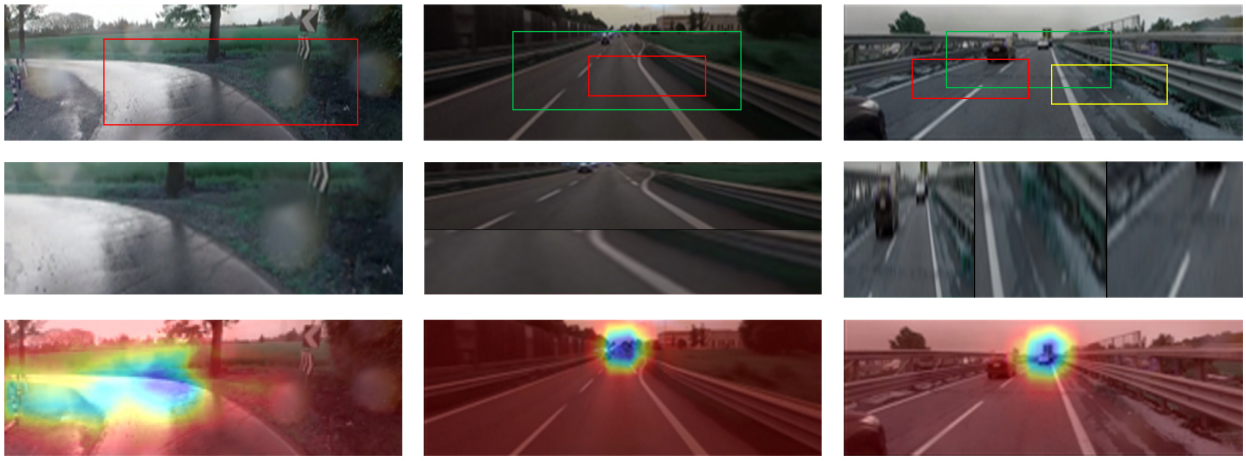


Figure 5. *From top: Input frame with boundary boxes indicating the region of interest selected by STN, stacked and resized image/s generated by STN, and the input frame overlaid with the gaze map output prediction of the gaze-branch of the MTL network.*

The results were very good as the predicted steering angle was almost the same as that of the driver. Moreover, the predicted steering angles did keep the car inside the lane for nearly 5 min of testing at a speed of 50 Km/hr. This experiment indicates the model capability of following the lane in a different distribution and unseen real environment without any further training or fine-tuning.

All the prepossessing needed was "histogram matching" which is a technique used to match the color distribution of the new testing frames to that of the training frames. The STN made it possible to use the histogram matching technique in this problem because the STN output frame which is a zoomed and cropped part of the original frame is always focused on the road and the lanes. Consequently, there are nearly two main colors in the frame which are: a degree of the black color of the road, and another degree of the white color of the lanes, and no more objects with different colors exist, which can vastly change the color components of each frame making it hard to compensate for the differences between the training data and testing data as shown in Fig.6.

Conclusions and Future Work

Inspired by how humans learn to drive, in this paper, the gaze information is incorporated with the front-facing camera frames in a multitasking deep neural network to improve steering angle estimation and fulfilling the real-time autonomous cars system requirements.

Furthermore, this work shows that autonomous driving is enhanced by **36.18%** by incorporating eye gaze information enabling generalization and operation in different unseen environments without being explicitly trained in them.

Table 1. Results Summary

Architecture	RMSE	Improvement (%)	Inference time (sec.)
Baseline (PilotNet)	0.0105	-	0.0028
Middle-fusion	0.007113	30.45%	27.008
STN-based	0.008312	20.83%	27.005
Multi-task	0.008250	21.42%	0.0028
STN+MTL (2 trans.)	0.007101	32.37%	0.01
GG-Net STN+MTL (3 trans.)	**0.006701**	**36.18%**	**0.015**

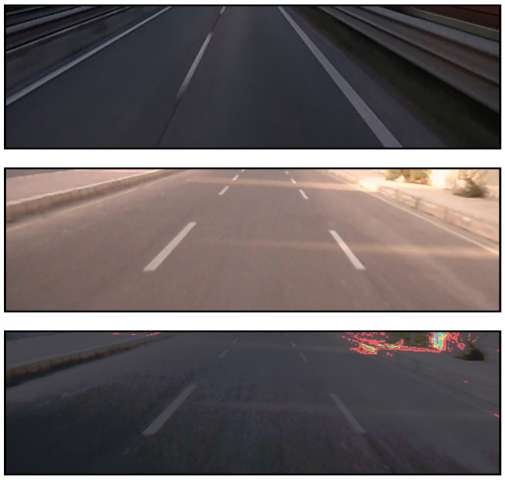


Figure 6. *Histogram matching.from top: a sample frame from the training dataset, input frame from the new environment, and the final histogram matching output, which is the same as the input frame but with the color distribution of the training dataset frames*

This paper made use of a Spatial Transformer Network (STN) for the first time in SDC which enabled the model to focus on the important features in the road such as the lanes and eliminate unnecessary elements. Moreover, this work enhances the idea of using a multitask learning approach to improve steering angle predictions and generate gaze maps as an auxiliary task in addition to decreasing the inference time greatly. Due to the nature of this specific problem, being able to operate in real-time is a must. Consequently, the inference time is optimized to reach 0.015 sec/frame on NVIDIA K80 GPU.

Moreover, the GG-Net is tested on a recorded dataset from Egypt and the model was able to predict accurate steering commands without any fine-tuning which is proof of its ability to generalize to unseen environments. For future work, this technique and architecture can be implemented in more complex environments like urban cities. Also, the multitask network can have more than two auxiliary tasks that are expected to improve the performance further.

Acknowledgement

This work was partially funded by the Academy of Scientific Research and Technology (ASRT) and ONE Lab at Zewail City of Science and Technology and at Cairo University.

References

[1] Stefano Alletto, Andrea Palazzi, Francesco Solera, Simone Calderara, and Rita Cucchiara. Dr (eye) ve: a dataset for attention-based tasks with applications to autonomous and assisted driving. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops*, pages 54–60, 2016.

[2] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[4] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias icml. *Google Scholar Google Scholar Digital Library Digital Library*, 1993.

[5] R Caruana. Multitask learning. autonomous agents and multi-agent systems. 1998.

[6] Yuying Chen, Congcong Liu, Lei Tai, Ming Liu, and Bertram E Shi. Gaze training by modulated dropout improves imitation learning. *arXiv preprint arXiv:1904.08377*, 2019.

[7] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9329–9338, 2019.

[8] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[9] I Mohamed Elzayat, M Ahmed Saad, M Mohamed Mostafa, R Mahmoud Hassan, Hossam Abd El Munim, Maged Ghoneima, M Saeed Darweesh, and Hassan Mostafa. Real-time car detection-based depth estimation using mono camera. In *2018 30th International Conference on Microelectronics (ICM)*, pages 248–251. IEEE, 2018.

[10] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, and Yi Yang. Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. *arXiv preprint arXiv:1801.09927*, 2018.

[11] Qiang Ji, Zhiwei Zhu, and Peilin Lan. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE transactions on vehicular technology*, 53(4):1052–1068, 2004.

[12] Congcong Liu, Yuying Chen, Lei Tai, Haoyang Ye, Ming Liu, and Bertram E Shi. A gaze model improves autonomous driving. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pages 1–5, 2019.

[13] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[14] Ahmed Mahmoud, Loay Ehab, Mohamed Reda, Mostafa Abdelaleem, Hossam Abd El Munim, Maged Ghoneima, M Saeed Darweesh, and Hassan Mostafa. Real-time lane detection-based line segment detection. In *2018 New Generation of CAS (NGCAS)*, pages 57–61. IEEE, 2018.

[15] Andreas Mogelmose, Mohan Manubhai Trivedi, and Thomas B Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

[16] Oyewole Oyekoya and Fred Stentiford. Exploring human eye behaviour using a model of visual attention. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 945–948. IEEE, 2004.

[17] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver's focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018.

[18] Ibrahim Sobh, Loay Amin, Sherif Abdelkarim, Khaled Elmadawy, Mahmoud Saeed, Omar Abdeltawab, Mostafa Gamal, and Ahmad El Sallab. End-to-end multi-modal sensors fusion system for urban automated driving. In *Proceedings of Machine Learning for Intelligent Transportation Systems Workshop NeurIPS*, 2018.

[19] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. *arXiv preprint arXiv:1704.01631*, 2017.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[21] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[22] Ruohan Zhang, Zhuode Liu, Luxin Zhang, Jake A Whritner, Karl S Muller, Mary M Hayhoe, and Dana H Ballard. Agil: Learning attention from human for visuomotor tasks. In *Proceedings of the european conference on computer vision (eccv)*, pages 663–679, 2018.

[23] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

Appendix

A car with a certain constant steering angle will move on a circular path as shown in Fig.7. Each one of the front wheels will move on a circle with a different radius and each wheel will have a slightly different angle to get a smooth motion on the circular path. For the model simplicity

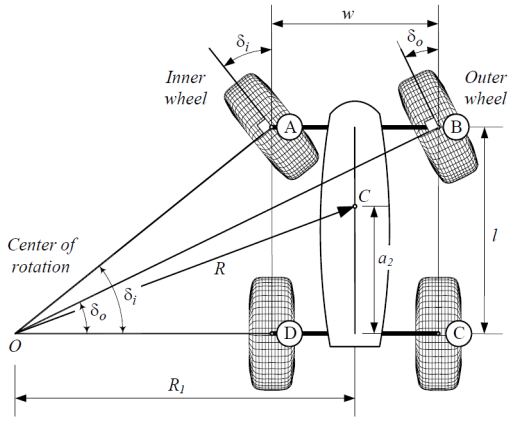


Figure 7. *Car steering geometry.*

and without losing generality, only the circle with the small radius "the one drawn by the wheel nearer to the center of the turning circle" is used. The point O is the intersection of 2 lines drawn perpendicular to the front and the back wheels and it's the center of the motion circle. Its position changes with steering angle, consequently, changing the center and the radius of the circle. The AD side length is l which is the "wheelbase length." It's the distance between the front and the back wheels and it differs from one car to another. From the trigonometry of the right-angle-triangle ODA with the "left wheel steering angle" = $\delta_i = \mathbf{SA}$,

$$SA = \tan^{-1}(\frac{L}{R}) \quad . \tag{2}$$

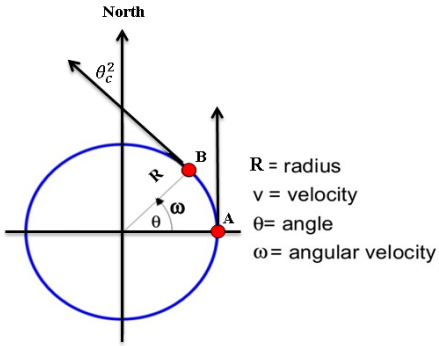


Figure 8. *Course angle*

This view was in a specific time instance "static form." As the car is moving, we should consider the dynamic form as follows: In Fig.8, it is assumed that the car was at point A at time Zero and the course angle in this case will be θ_c^1 = Zero, after time t with a constant steering angle SA the car will move on the arc "AB" and reach point B the course angle will be θ_c^2 , from the circular motion equations:

$$v = \frac{2\pi r}{t} \tag{3}$$

From the geometry, it is concluded that $\theta = \theta_c$, so:

$$\theta_c^1 - \theta_c^2 = \frac{t\,v}{R} \tag{4}$$

From 2, 3, and 4, the final relation between steering and course angles becomes:

$$\Theta_s = SR \times \tan^{-1}(\frac{(\theta_c^1 - \theta_c^2) \times L}{v \times t}) \tag{5}$$

SR indicates the steering ratio which is the ratio of the steering wheel angle to that of the car wheels steering angle. As the type of car used in the dataset is not specified, an average number for all the constants of the car types in 5 for steering ratio and the wheelbase length is used, t is the time between two frames of the dataset (25 fps), the velocity (v) at each frame is given in the dataset, leading to mathematical proof to the course-steering conversion algorithm.

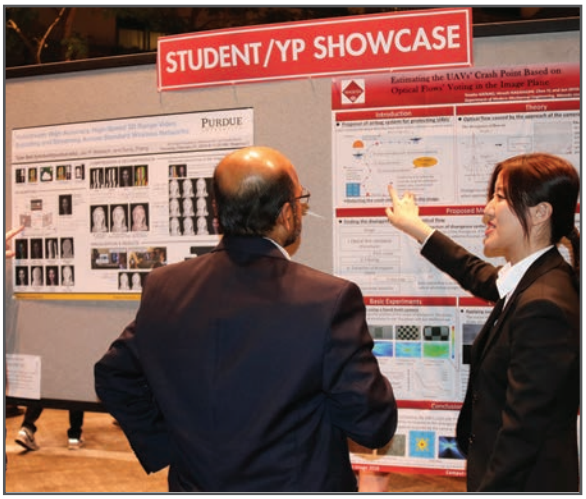
JOIN US AT THE NEXT EI!

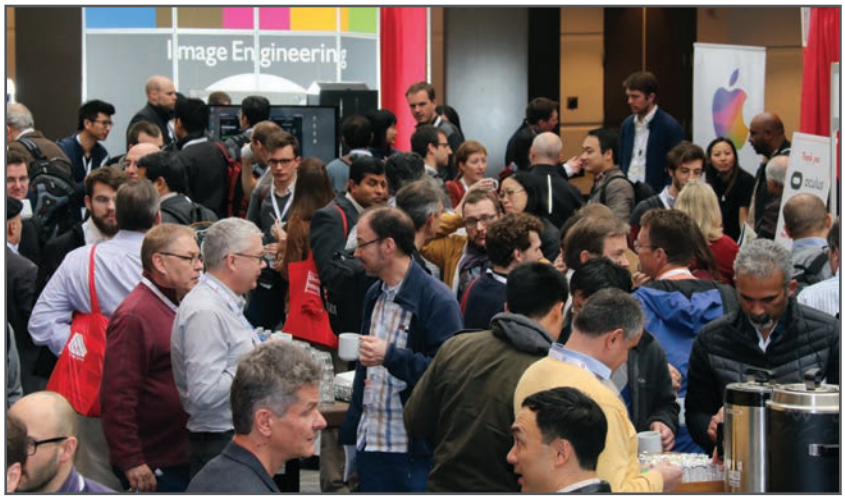
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org