Trajectory Prediction Using Video Generation in Autonomous Driving

David-Traian IANCU*, Mihai NAN, Ștefania-Alexandra GHIȚĂ, Adina-Magda FLOREA
University Politehnica of Bucharest, 313 Splaiul Independenței, Bucharest, 060042, Romania
david_traian.iancu@upb.ro (**Corresponding author*), mihai.nan@upb.ro,
stefania.a.ghita@upb.ro, adina.florea@upb.ro

Abstract: Trajectory prediction for the surrounding cars is a useful task in autonomous driving for obvious reasons. The traditional methods for predicting the future trajectories of surrounding cars involved complex motion models and patterns, complex maneuvers or physical models of the car trajectories. More recent works aim to predict the future car positions by using deep learning and neural networks. In this paper, video generation models were employed, which provide an estimation of the future frames related to the car positions based on an existing video and can obtain the position of the selected cars by employing an object detection algorithm along with additional information obtained by a segmentation module that uses a semantic segmentation network. The results were validated by employing the Root Mean Square Error (RMSE) metric in order to predict the locations of the surrounding cars and estimate their depth. Apparently, this approach has never been implemented in order to obtain the trajectory and the future position of the surrounding cars in autonomous driving.

Keywords: Trajectory prediction, Video generation, Object detection, Semantic segmentation, Depth prediction, Autonomous driving.

1. Introduction

In recent years, a lot of research has been carried out with regard to developing an autonomous car, in both academia and private corporations, especially car manufacturers. However, due to its complexity, the problem is not yet solved, even if some companies have developed cars with a certain degree of autonomy. An autonomous car involves a lot of components such as object detection, segmentation, depth estimation, trajectory prediction, route planning and path following. The most important tasks are those regarding scene understanding and the prediction of the surrounding objects, especially cars and people. Even if the scene understanding is perfect, if the system doesn't know that the car from behind is going to overtake another car, accidents can happen. Trajectory prediction is also important for knowing when to brake when a car comes in front of the autonomous vehicle - the system should know if the car will go faster or slower. The problem of trajectory prediction has been tackled for many years (Payeur, Le-Huy & Gosselin, 1995). Older approaches tried to model the velocity, direction and other physical parameters into a motion model, to obtain the trajectory and compute future positions (Houenou et al., 2013).

Newer approaches use deep neural networks, such as LTSMs (Altché & La Fortelle, 2017), and avoid complex physical modelling. However, in recent years, a new task emerged - the prediction of future frames from an existing video. The video generation task is generally performed with neural networks, too, such as Generalized Adversarial Networks (GAN) or Variational Autoencoders (VAE) and has the advantage that one can easily obtain the future positions of the surrounding cars, using an object detection algorithm.

In combination with information regarding semantic segmentation, the prediction of the car position can be further improved. Because video prediction is more complex than trajectory prediction, as far as one can tell this approach was not applied until now. However, the problem with a trajectory prediction model is that it is harder to train, because it requires a certain amount of annotated data regarding the surrounding cars and their trajectories, compared to a video prediction model, which requires only the frames as training data and can be trained with any existing driving video.

This paper proposes a trajectory prediction system using video generation, object detection and semantic segmentation and depth prediction was also used for validating the predicted depth compared to the actual depth of the position of a car, which is another useful information (for braking, for example). The remainder of this paper is structured as follows. Section 2 presents the related works regarding each of these tasks - trajectory prediction, video prediction, object detection, semantic segmentation and depth prediction. Section 3 analyses the architecture used for the experiments carried out, the models used

https://doi.org/10.24846/v31i1y202204

for the video generation task and the manually annotated dataset from University Politehnica of Bucharest. Section 4 describes the experiments and the metrics employed for analysing the performance of the proposed architecture for trajectory prediction. Section 5 presents the results obtained for the proposed dataset and Section 6 includes the conclusion of this paper and the proposals for future work.

2. Related Work

The proposed approach involves multiple components - object detection, video generation, semantic segmentation, depth estimation and trajectory prediction. This is the last paper in a series of autonomous driving studies applied on manually annotated datasets from the University Politehnica of Bucharest. The analysis focused on object detection (Iancu, Sorici & Florea, 2019), semantic segmentation (Iancu, Sorici & Florea, 2020) and depth estimation (Iancu et al., 2021). Based on the previous results, the best performing network was for each of these tasks. In this section, the existing architectures will be briefly discussed for each of the aforementioned tasks. Also, trajectory prediction and video generation will be analysed.

Object detection

The object detection task consists in detecting the bounding boxes of the objects in each selected image. In a previous study by Iancu, Sorici & Florea (2019), the most important object detection architectures were analysed. There are three major types of object detection networks – two-stage detectors like Faster R-CNN (Ren et al., 2017) or R-FCN (Dai et al., 2016), one-stage detectors like YOLO (Redmon et al., 2016), DSSD (Fu et al., 2016), RetinaNet (Lin et al., 2017) and anchor-free networks like Cornernet (Law et Deng, 2018), CenterNet (Duan et al., 2019) or Fully Convolutional One-Stage Object Detection (FCOS) (Tian et al., 2019). The first two types of networks use predefined anchors in order to detect the objects but differ regarding the number of stages involved in the detection. The anchor-free networks are newer architectures and don’t use anchors anymore. In this paper, YOLO v4 (Bochkovskiy, Wang & Liao, 2020) was chosen.

Semantic segmentation

The semantic segmentation detects the objects in an image, too, but unlike the object detection task, which offers a bounding box for an object, the semantic segmentation maps each pixel in the image to a given label (car, person, road, etc). There are different types of segmentation depending on how they classify the objects and the background pixels. The semantic segmentation classifies each pixel in the image without taking into account different objects in the same class, for example, Fully Convolutional Networks (FCN) (Long, Shelhamer & Darrell, 2015), PSPNet (Zhao et al., 2017), DeepLab (Chen et al., 2017), or SegNet (Badrinarayanan, Kendall & Cipolla, 2015). The instance segmentation assigns a different label to each object but does not take into account the background and generally is based on an object detection network, for example, Mask R-CNN (He et al., 2017), SDS (Hariharan et al., 2014) or CenterMask (Lee & Park, 2020). The panoptic segmentation merges the semantic segmentation and the instance segmentation into one single task, assigning different labels to different objects and assigning a class for each pixel, including the background, for example, Panoptic FPN (Kirillov et al., 2019), Efficient Panoptic Segmentation (Mohan & Valada, 2020) or DeeperLab (Yang et al., 2019). The study of Iancu, Sorici & Florea (2020) analysed the most important semantic segmentation networks for road detection, which are also used for this task, as it is explained in Section 3. For the experiments presented in this paper the FCN network was used.

Depth estimation

The depth estimation task is more difficult to evaluate because there are only a few sensors that can estimate the distance from the car to the other elements in the traffic. The biggest problem is that these sensors are very expensive and cheaper ones are not that precise. However, in this paper, depth estimation is used in order to obtain qualitative data rather than quantitative data, regarding the predicted distance to the surrounding cars compared to the actual distance. The depth estimation networks can be divided into monocular depth estimation networks and stereo depth estimation networks. Furthermore, the monocular depth networks can be divided into supervised networks, unsupervised networks and semi-supervised networks. The monocular depth estimation networks, such as Monodepth2 (Godard et al., 2018), Megadepth (Li & Snavely, 2018), DORN (Fu et al., 2018) or LKVOLearner (Liu et al., 2015) have certain practical advantages and have been used in this paper. For the current experiments, Monodepth2 was used in order to evaluate the estimated distance to the predicted position of the surrounding cars compared to the real distance.

Trajectory prediction

Trajectory prediction approaches have evolved since the emergence of this research topic. A good review of the methods used can be found in (Leon & Gavrilescu, 2021) and at (Rudenko et al., 2019). The most important distinction that could be noticed is that older models didn't use neural networks, instead, they were based on the physical properties of the cars and the system (Batz, Watson & Beyerer, 2009), on recognizing the manoeuvres of the vehicles (Houenou et al., 2013) or trying to estimate the trajectory function of the vehicles using different techniques, such as hidden Markov Models (Huanget et al., 2019) or Gaussian Mixture models (Palmieri et al., 2019; Wiest et al., 2012). However, the most interesting works are using different types of neural networks. The most used architectures are long short-term memory (LSTM) neural networks (Altché & La Fortelle, 2017; Ma et al., 2019) recurrent neural networks (RNNs) like (Kim et al., 2017), encoder-decoder LSTM (Deo & Trivedi, 2018) or even generalized adversarial networks (GANs) (Gupta et al., 2018) or encoder-decoder attention networks (Zheng et al., 2020). Some of the networks include social elements, taking into consideration that even if some paths are possible, they are not socially acceptable, for example social GAN (Gupta et al., 2018), social LSTM (Alahi et al., 2016; Deo & Trivedi, 2018). Even if some of the models were made for human trajectory prediction, the same strategies can be applied to vehicle trajectory prediction for the surrounding cars. One of the best trajectory prediction models is TraPHic (Chandra et al., 2019), which uses both LSTM and convolutional layers, and was used in the experiments described in this paper to compare a state-of-the-art trajectory prediction network with the proposed system which represents a video prediction network with semantic segmentation features.

Video generation

The task of video generation is harder than trajectory prediction because the employed architecture has to predict an entire frame, not just some trajectory, which can be inferred by taking into account manoeuvres, physical models, etc. Because of the difficulty of the task, the existing models are not so many as the trajectory prediction models and most of them work only for small images with limited details - a small number of pixels and if the case only one object involved. A comprehensive review of frame prediction can be found in (Oprea et al., 2020). Unlike the previous models, video generation is performed almost always using neural networks. Some examples of architectures are LSTMs (Srivastava, Mansimov & Salakhutdinov, 2015), convolutional LSTMs (Kalchbrenner et al., 2016; Finn, Goodfellow & Levine, 2016), RNNs (Oliu, Selva & Escalera, 2017) and CNNs (De Brabandere et al., 2016), but in the latest years, the emergence of GANs has increased, using different approaches: two discriminators (Tulyakov et al., 2017), Wasserstein models (Wu et al., 2017, Kratzwald et al., 2017), stacked convolutions (Vondrick, Pirsiavash & Torralba, 2016), a multi-scale architecture (Mathieu, Couprie & LeCun, 2016) and GANs have also been used for saliency prediction (Pan et al., 2017). Variational autoencoders (VAE) are also used for this particular task (Pan et al., 2019) or even a combination of VAE and GAN (Le et al., 2018). An important consideration when discussing about video generation is that some architectures try to generate the real future frames (trajectory prediction approach) while others don't try to guess what will happen, but rather to generate some valid video sequences based on a certain frame and eventually an action that must happen (Siarohin et al., 2021). In the experiments carried out, PredNet was employed, a convolutional LSTM network with good results on real videos (Lotter, Kreiman & Kox, 2016), along with Seg2Vid (Pan et al., 2019), a convolutional VAE with semantic segmentation features, which was also tested on driving video sequences and Stochastic

Adversarial Video Prediction (SAVP) (Lee et al., 2018) which combines the autoencoders with generative models.

3. Architecture and Dataset

This section describes the architecture used in the experiments which were carried out, which involves video generation, semantic segmentation, object detection and depth estimation, and the proposed dataset which was created at University Politehnica of Bucharest and manually annotated.

Architecture

The first step of the proposed architecture consists in taking the data and putting it into a video generation model. As it was previously stated, one used three different existing architectures with pre-trained weights on standard driving datasets such as Cityscapes (Cordts et al., 2015) or KITTI (Geiger et al., 2013), which are bigger than the employed dataset and, thus, more suitable for training. The model used for SAVP can be found at the public repository of Lee et al. (2018), the model used for Seg2Vid can be found at the public repository of Pan et al. (2019) and the model used for PredNet can be found at the public repository of Lotter, Kreiman & Kox (2016).

The second step in the workflow is to detect all the surrounding cars from the predicted frames. This is done with YOLO v4, a state-of-the-art detector. The final step is to improve the prediction by using the proposed segmentation module. The segmentation module receives the predicted frames along with the predicted positions of the surrounding cars and the segmentation for the last frame before the predicted ones.

The module computes the relative coordinates of the cars with regard to the road and the relative coordinates of the predicted position of the cars with regard to the estimated segmentation for the road and learns the best way to combine the two sets of relative coordinates in order to make a better prediction. The segmentation is made with FCN using the reference model found in (Yang & Chung, 2018). For evaluation purposes, the predicted frames are also included into a depth estimation network in order to see the difference between the estimated depth and the actual depth of the cars. The actual depth is considered by taking the real coordinates of the surrounding cars and the estimated depth is measured using the predicted coordinates. The depth is computed using Monodepth2 and the public reference model that can be found at the public repository of Godard et al. (2018), which also provides pretrained weights for KITTI. The code used for the experiments that were carried out and the detailed results can be found in (Iancu, 2021). All the experiments that were carried out are described in the following section. The proposed architecture is illustrated in Figure 1.

Dataset

For the experiments that were carried out, the dataset from the University Politehnica of Bucharest was used.

The dataset consists of several short videos with 35 frames, the last 5 frames being predicted by each video generation model based on the previous ones. The data is divided into three categories according to the time of the day when the videos were recorded, following the idea in (Iancu, Sorici & Florea, 2019) - videos recorded during the day, at dusk or dawn and during the night. Each video was selected to contain at least one car in every image, especially in the last ones that should be predicted. 107 short videos were recorded during the day, 37 videos at dusk and 47 videos during the night.

There is a double motivation behind using the employed dataset. The first reason is that it was necessary to use a dataset with a semantic

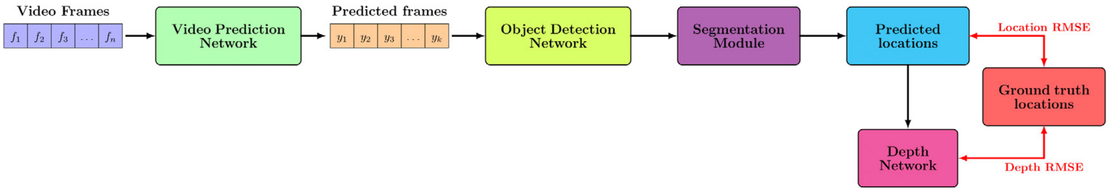

Figure 1. The proposed architecture

annotation of the road in order to incorporate features about the road into predicting the position of the surrounding cars and a manually annotated dataset created at University Politehnica of Bucharest was already available.

The second reason is that the research that was carried out is part of a series of studies regarding autonomous driving which were carried out at University Politehnica of Bucharest. Beside the manual annotation of each frame with regard to road segmentation, each car was also manually annotated in order to include the real positions of the surrounding cars. There were over 4000 cars annotated in the totality of frames. All the cars were annotated in the predicted videos, too, in order to have an upper bound for the prediction, if the object detection task worked perfectly. For depth estimation, all the frames were evaluated using Monodepth 2.

4. Experiments and Metrics

This section describes the experiments that were carried out and the metrics used in order to evaluate the predictions made. Taking into account that there are no similar approaches and that the experiments were carried out based on the proposed dataset, where the road is segmented in all the images, the obtained results should be seen as being rather qualitative (in that they can help one compare different architectures) than quantitative.

Metrics

For the experiments that were carried out, two relevant metrics were used. For prediction purposes, the Root Mean Square Error (RMSE) was computed for all the four predicted coordinates. In the following formula, N stands for the number of cars, x_{ip} is the x coordinate of the i-th predicted car, x_{ir} is the real x coordinate of the i-th car, y_{ip} is the y coordinate of the i-th predicted car and y_{ir} is the real y coordinate of the i-th car.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_{ip} - x_{ir})^2 + (y_{ip} - y_{ir})^2}{N}} \qquad (1)$$

The RMSE was computed in order to employ an evaluation metric that would be similar to the reference trajectory prediction model that was used. The RMSE was computed for all the experiments described in this section.

The RMSE for the average depth of the predicted position of the surrounding cars was also computed by considering the squared difference between the mean of the depth pixels in the predicted boundary box and the mean of the depth pixels in the real boundary box. In the following formula, N stands for the number of the cars, N1 stands for the number of the pixels in the predicted bounding box of the j-th car and N2 stands for the number of the pixels in the real bounding box of the j-th car.

$$RMSE_{Depth} = \sqrt{\frac{\sum_{j=1}^{N}\left(\left(\frac{\sum_{i=1}^{N1} PredictedDepthPixel_i}{N1}\right) - \left(\frac{\sum_{i=1}^{N2} RealDepthPixel_i}{N2}\right)\right)^2}{N}} \qquad (2)$$

Even if the RMSE is computed for the predicted coordinates of a car position, the RMSE for the average depth of the position of a car is a better qualitative metric for estimating the depth difference between the predicted position and the real position of a car. Even if, in theory, there could be similar places in an image at the same distance from the selected car, if the predicted distance is close to the real one the prediction is good enough. Furthermore, the distance from the car is what matters the most in the case of an autonomous car.

Experiments

Multiple experiments were carried out regarding the predicted location of a car and the predicted depth for that location. For all the images the predictions were tested against the ground truth, the manually annotated positions of the cars in the corresponding images and against the detected locations obtained by YOLO. The metrics were computed once without considering the segmentation module and based on the segmentation module. The depth of a car`s location was also computed regarding the estimated ground truth depth (the depth network applied on the real image) and the estimated depth on the predicted images. Also, as segmentation is concerned, the following were considered: the ground truth segmentation of the last frame and the ground truth segmentation of the predicted frames along with the estimated segmentation of the last frame and the estimated segmentation of the predicted frames. To sum up, the following experiments were carried out:

- RMSE for the predicted location of a car and the depth of its position without segmentation
- RMSE for the predicted location and the depth with ground truth segmentation
- RMSE for the predicted location and the depth with estimated segmentation by FCN

For all of these experiments, the time of the day was considered (during the day, at dusk or during the night), which has not been done before for this task. Also, the RMSE was computed considering different car sizes and the prediction time.

5. Results

This section presents the most important results obtained from the experiments that were carried out. As it was mentioned in the previous section, two different types of measurements were made - the RMSE for the predicted location of a car and the RMSE for the depth of a car's position. The results are synthesized in Table 1, Table 2, Table 3 and Table 4. All the tables show the results obtained during the day, at dusk and during the night and the average RMSE (obtained as a weighted mean). In the following analysis, if the time of the day is not mentioned, reference is made to the average RMSE. A number of second frames predicted by all the three networks mentioned in Section 2, along with the real images taken during the day, at dusk and during the night can be seen in Figure 2.

Table 1 and Table 2 display the results regarding the predicted location for the surrounding cars. For each of the three networks tested as the video generation architecture, three metrics were included - the RMSE without the proposed segmentation module, the RMSE with the proposed segmentation module, but considering the ground truth segmentation (manually annotated) and the RMSE considering the segmentation given by the FCN network. In Table 1, the RMSE is computed considering the ground truth detection of the predicted position of the surrounding cars, manually annotated. In Table 2, the detection is made by means of YOLO v4. The results show certain interesting details. For the ground truth detection, the segmentation module does not improve the RMSE much, the results being almost the same, only slightly different for Seg2Vid. However, given that a real system does not include the future positions of the cars, it can be noticed that the segmentation module offers certain significant improvements, especially for SAVP. Nevertheless, as expected, the error is still higher than in the case of manually annotated cars. This is due to the imprecise predictions made by YOLO, caused by the distortion of the images based on some predicted frames, which could still

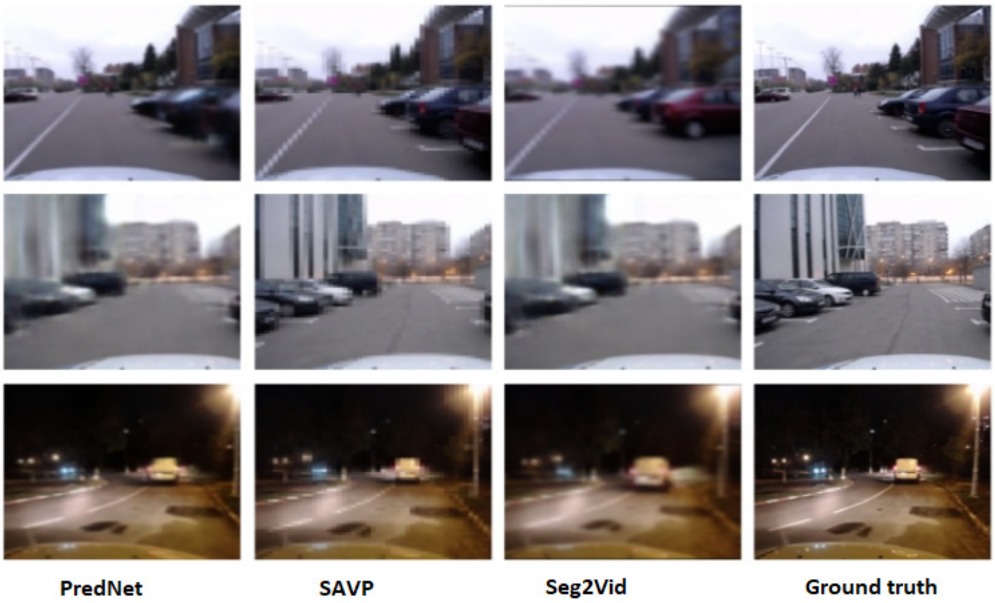

Figure 2. Predicted frames for different light conditions

be understood by humans, but much harder by a detection network. This is where the segmentation module helps in improving the quality of the prediction. As a reference point, Table 1 also includes the RMSEs for TraPHic, a state-of-the-art prediction network, which are significantly smaller, the best results being obtained at dusk and the worst during the night.

Table 1. RMSE for location (GT detection)

	Day	Dusk	Night	Avg.
PredNet				
No segmentation	318.20	71.73	94.28	247.91
GT segmentation	317.20	71.10	93.70	247.11
FCN segmentation	317.12	71.24	93.49	247.00
SAVP				
No segmentation	294.46	128.88	116.83	233.71
GT segmentation	294.12	128.84	116.34	233.41
FCN segmentation	294.12	128.85	116.15	233.41
Seg2Vid				
No segmentation	321.20	148.91	115.27	265.85
GT segmentation	316.06	145.65	113.58	261.67
FCN segmentation	320.19	147.16	114.63	264.91
TraPHic	83.78	54.60	114.27	86.29

Table 2. RMSE for location (YOLO detection)

	Day	Dusk	Night	Avg.
PredNet				
No segmentation	280.01	193.69	46.45	269.98
GT segmentation	275.45	157.91	46.40	263.69
FCN segmentation	275.45	158.58	46.40	263.69
SAVP				
No segmentation	615.78	568.45	543.74	561.84
GT segmentation	494.61	480.95	322.51	390.35
FCN segmentation	497.73	478.46	323.96	393.00
Seg2Vid				
No segmentation	320.88	278.37	103.58	310.92
GT segmentation	306.16	191.44	98.29	287.88
FCN segmentation	306.36	196.98	102.15	289.27

Another interesting fact is that the results are almost identical if one considers the ground truth of the segmentation and the results for the segmentation given by the FCN network. Given that the segmentation is performed for the last real frame, the results for the segmentation network are very close for the ground truth segmentation, hence the small difference between the results obtained for the ground truth and the FCN. This relation is maintained even if the results for YOLO were used instead of the manually annotated positions.

An unexpected fact is that the results obtained during the night are better and those obtained during the day are worse for both the ground truth and the YOLO predictions, but this is due to the higher number of cars and predictions made during the day, the average being closer to the results obtained during the day. From over 4000 cars annotated in the real images, in the images obtained by GT detection about 3000 cars were detected for Seg2Vid and PredNet and 2500 cars for SAVP, which is still much more than the detection made by YOLO - about 500 results for Seg2Vid, 170 for PredNet and only 100 for SAVP, the recall varying between 4% and 20%, which is somehow expected given that the frames are predicted and the quality of the image is distorted.

Another interesting result is that PredNet has the smallest error with regard to YOLO predictions, even if for ground truth predictions the results are slightly better for SAVP. However, PredNet obtained predicted frames of a higher quality and the cars can be better detected by YOLO. Table 3 and Table 4 offer the same statistics for the depth prediction but they include two different metrics – one regarding the depth of the real frames and one regarding the depth of the predicted ones.

Table 3. RMSE for depth (GT detection)

	Day	Dusk	Night	Avg.
PredNet				
No segm.	142.08	8.63	23.60	104.78
No segm. (pred.)	142.98	14.02	25.74	105.68
GT segm.	145.13	8.36	22.89	111.06
GT segm. (pred.)	145.63	13.18	24.13	111.60
FCN segm.	145.58	8.54	23.46	111.37
FCN segm. (pred.)	146.54	13.92	25.48	112.16
SAVP				
No segm.	141.49	13.34	25.95	103.33
No segm. (pred.)	126.14	18.5	31.47	93.01
GT segm.	141.97	13.24	25.23	106.49
GT segm. (pred.)	126.38	17.41	31.29	95.68
FCN segm.	141.98	13.24	25.70	106.49
FCN segm. (pred.)	126.88	18.50	31.30	95.96
Seg2Vid				
No segm.	134.10	18.68	27.52	103.76
No segm. (pred.)	129.25	23.52	32.47	100.62
GT segm.	138.29	18.57	27.41	110.01
GT segm. (pred.)	132.85	22.54	30.79	106.21
FCN segm.	138.28	18.66	27.41	110.01
FCN segm. (pred.)	132.84	23.36	32.23	106.20

Table 4. RMSE for depth (YOLO detection)

	Day	Dusk	Night	Avg.
PredNet				
No segm.	69.69	20.21	6.87	65.08
No segm. (pred.)	168.77	24.93	0.64	156.86
GT segm.	62.19	19.58	6.51	58.89
GT segm. (pred.)	165.66	16.27	0.55	154.70
FCN segm.	65.27	19.58	5.94	61.76
FCN segm. (pred.)	159.33	19.52	0.55	149.27
SAVP				
No segm.	38.66	29.47	32.69	33.45
No segm. (pred.)	81.66	69.23	44.97	57.21
GT segm.	34.76	31.80	36.41	36.09
GT segm. (pred.)	78.15	49.28	41.71	52.57
FCN segm.	34.95	31.80	36.41	36.09
FCN segm. (pred.)	70.73	48.20	42.04	52.57
Seg2Vid				
No segm.	62.51	25.72	20.26	56.57
No segm. (pred.)	137.39	40.86	19.79	122.98
GT segm.	64.13	25.91	19.60	58.77
GT segm. (pred.)	136.17	31.78	14.89	123.16
FCN segm.	64.10	25.91	19.60	58.75
FCN segm. (pred.)	134.93	32.61	19.78	122.65

Both metrics should be interpreted qualitatively, in order to evaluate the efficiency of the predictions. It can be noticed that the depth error is bigger for the predicted frames, but the difference is small. Also, the depth error is a little bigger for the semantic segmentation module, because the module took into account only the location, but the difference is very small.

Another unexpected result is that the depth error is smaller for the detections obtained by YOLO, but this is due to the smaller number of detected cars.

Again, the difference between the results for the ground truth segmentation and those for the segmentation obtained by FCN is almost unnoticeable. The depth error appears to be smaller for SAVP but only slightly smaller, at least for the ground truth detections. The results obtained at dusk are better for the ground truth detection, those obtained during the night are better for YOLO and the worst results were obtained during the day, the average being closer to the results obtained during the day.

The RMSE was also measured with regard to the predicted location for different object sizes - the cars were divided into 11 classes according to their size: between 0 and 100 pixels, 100 and 250, 250 and 500, 500 and 750, 750 and 1000, 1000 and 2000, 2000 and 3000, 3000 and 5000, 5000 and 10000, 10000 and 20000 and between 20000 and 30000 pixels. The results can be seen in Figure 3 and they were obtained with regard to the real architecture, with YOLO detections and the FCN network. It can be noticed that even if there is no pattern for smaller car sizes, the error tends to be smaller for bigger cars.

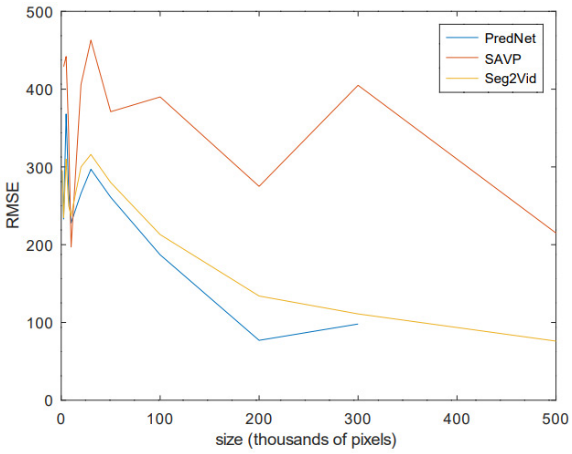

Figure 3. RMSE with regard to car size

The last experiment carried out was focused on the prediction time for the video generation networks. Because the video generation networks require a specified image size (in the contrary case the image is resized), the prediction time is almost the same for different image sizes. 5 image sizes were tested - 640x360, 640x480, 1280x720, 1920x1080 and 3840x2160. he results can be seen in Figure 4.

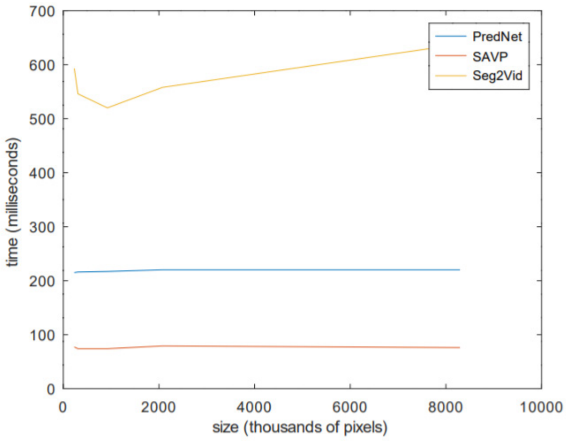

Figure 4. Prediction time

Seg2Vid obtained a slightly longer prediction time for a bigger image, but the two other networks obtained almost the same prediction time for different image sizes. Unfortunately, the inference time is too long in order to use them for an autonomous car soon enough, for example the SAVP reaches 10 fps.

6. Conclusion and Future Work

This paper presents a new trajectory prediction algorithm for the surrounding cars based on video generation for predicting new frames, object detection in order to detect the surrounding cars and semantic segmentation in order to fine-tune the results regarding the segmentation of the road, an approach that has not been implemented before. The final architecture can be used for any dataset, without having to make manual annotations for the detection or the segmentation tasks. Also, for testing purposes, the estimated depth of a car's position in relation to the surrounding cars is evaluated for the predicted frames in comparison with the estimated depth for the real frames. Three different video prediction models were used and the results were compared with those of a state-of-the-art trajectory prediction model. For each model, different experiments were carried out with regard to the time of the day for a certain prediction and the inclusion of the road semantic segmentation in the proposed architecture. The obtained results show that the inclusion of the semantic segmentation could slightly improve the predicted location of the cars in the future and that the proposed method could be used for trajectory prediction. The best model used was PredNet. The biggest advantage is that all the discussed video prediction models can be trained with any driving clip as training data, without having to manually annotate the respective trajectories. The future aim is to improve one of the video prediction models described in this paper by focusing on the trajectory prediction task, in order to obtain better predictions for the locations of the surrounding cars.

Acknowledgements

Special thanks are due to the University Politehnica of Bucharest which made this study possible. This research was funded through grants PN-III-P2-2.1-PED-2019-4995 and PN-III-P1-1.2-PCCDI-2017-0734.

REFERENCES

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L. & Savarese, S. (2016). Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 961-971).

Altché, F. & La Fortelle, A. (2017). An LSTM network for highway trajectory prediction. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, (pp. 353-359).

Badrinarayanan, V., Kendall, A. & Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *arXiv:1511.00561.*

Batz, T., Watson, K. & Beyerer, J. (2009). Recognition of dangerous situations within a cooperative group of vehicles. In *IEEE Intelligent Vehicles Symposium* (pp. 907-912).

Bochkovskiy. A., Wang, C.-Y. & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection, *arXiv:2004.10934*.

Chandra, R., Bhattacharya, U., Bera, A. & Manocha, D. (2019). Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8483-8492).

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. (2017). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*, 834-848.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition Workshop* (pp. 3213-3223).

Dai, J., Li, Y., He, K. & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In *Neural Information Processing Systems (NIPS) Conference, arXiv:1605.06409* (pp. 379–387).

De Brabandere, B., Jia, X., Tuytelaars, T. & Van Gool, L. (2016). Dynamic filter networks. In *Neural Information Processing Systems (NIPS) Conference* (pp. 667–675).

Deo, N. & Trivedi, M. M. (2018). Convolutional social pooling for vehicle trajectory prediction, *arXiv:1805.06771.*

Duan, K., Bai, S., Xie, L, Qi., H, Huang, Q. & Tian, Q. (2019). CenterNet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (pp. 6569–6578).

Finn, C., Goodfellow, I. J. & Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *Neural Information Processing Systems (NIPS) Conference, arXiv:1605.07157* (pp. 64–72).

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A. & Berg, A. C. (2016). DSSD: Deconvolutional single shot detector, *arXiv:1701.06659.*

Fu, H., Gong, M., Wang, C., Batmanghelich, K. & Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (pp. 2002-2011).

Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset, *The International Journal of Robotics Research, 32*(11), 1231-1237.

Godard, C., Aodha, O., Firman, M. & Brostow, G. (2018). Digging into self-supervised monocular depth estimation, *arXiv:1806.01260*.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S. & Alahi, A. (2018). Social GAN: Socially acceptable trajectories with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (pp. 2255-2264).

Hariharan, B., Arbelaez, P., Girshick, R. & Malik J. (2014). Simultaneous detection and segmentation. In *European Conference on Computer Vision* (pp. 297-312).

He, K., Gkioxari, G., Dollar, P. & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the International Conference on Computer Vision* (pp. 2961-2969).

Houenou, A., Bonnifait, P., Cherfaoui, V. & Yao, W. (2013). Vehicle trajectory prediction based on motion model and maneuver recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4363-4369).

Huang, Y., Bi, H., Li, Z., Mao, T. & Wang, Z. (2019). STGAT: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* (pp. 6272–6281).

Iancu, D. T. (2021). *Trajectory prediction by video generation*. Available at: <https://github.com/funkydvd/trajectory-prediction-by-video-generation>, last accessed: 17th of February, 2022.

Iancu, D. T., Nan, M., Ghiță, A. Ș. & Florea, A. M. (2021). Vehicle depth estimation for autonomous driving, *U.P.B. Scientific Bulletin, Series C*(3), 3-20.

Iancu, D. T., Sorici, A. & Florea, A. M. (2019). Object detection in autonomous driving-from large to small datasets. In *11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, (pp. 1-6).

Iancu, D. T., Sorici, A. & Florea, A. M. (2020). Neural road semantic segmentation in driving scenarios. In *12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, (pp. 1-6).

Kalchbrenner, N., Oord, A. V. D., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A. & Kavukcuoglu, K. (2016). Video pixel networks, *arXiv:1610.00527.*

Kim, B., Kang, C. M., Kim, J., Lee, S. H., Chung, C. C. & Choi, J. W. (2017). Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, (pp. 399-404).

Kirillov, A., Girshick, R., He, K. & Dollar, P. (2019). Panoptic feature pyramid networks. In *Conference on Computer Vision and Pattern Recognition (CVPR),* (pp. 6392-6408).

Kratzwald, B., Huang, Z., Paudel, D. P., Dinesh, A. & Van Gool, L (2017). *Improving video generation for multi-functional applications*, arXiv:1711.11453.

Law, H. & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV*), (pp. 734– 750).

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C. & Levine, S. (2018). Stochastic adversarial video prediction, *arXiv:1804.01523.*

Lee, Y. & Park, J. (2020). CenterMask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (pp. 13906-13915).

Leon, F. & Gavrilescu, M. (2021). A Review of Tracking and Trajectory Prediction Methods for Autonomous Driving, *Mathematics, 9*(6)*,* 660.

Li, Z. & Snavely, N. (2018). MegaDepth: Learning single-view depth prediction from internet photos. In *Conference on Computer Vision and Pattern Recognition* (pp. 2041-2050).

Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollar, P. (2017). Focal loss for dense object detection, *arXiv:1708.02002.*

Liu, F., Shen, C., Lin, G. & Reid, I. D. (2015). Learning depth from single monocular images using deep convolutional neural fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*, 2024-2039.

Long, J., Shelhamer, E. & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition* (pp. 3431-3440).

Lotter, W., Kreiman, G. & Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning, *arXiv:1605.08104.*

Ma., Y., Zhu, X., Zhang, S., Yang, R., Wang, W. & Manocha, D (2019). Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *The Thirty-Third AAAI Conference on Artificial Intelligence* (pp. 6120-6127).

Mathieu, M., Couprie, C. & LeCun, Y. (2016). Deep multiscale video prediction beyond mean square error. In *International Conference on Learning Representations*, *arXiv:1511.05440.*

Mohan, R & Valada, A. (2020) EfficientPS: Efficient panoptic segmentation, *arXiv:2004.02307.*

Oliu, M., Selva, J. & Escalera, S. (2017). Folded recurrent neural networks for future video prediction, *arXiv:1712.00311.*

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J. & Argyros, A. (2020). A review on deep learning techniques for video prediction, *arXiv:2004.05214.*

Palmieri, L., Kucner, T. P., Magnusson, M., Lilienthal, A. J. & Arras, K. O. (2017). Kinodynamic motion planning on gaussian mixture fields. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 6176-6181).

Pan, J., Canton-Ferrer, C., McGuinness, K., O'Connor, N. E., Torres, J., Sayrol, E. & Giro-i-Nieto, X. (2017). SalGAN: Visual Saliency Prediction with Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), arXiv:1701.01081.*

Pan, J., Wang, C., Jia, X., Shao, J., Sheng, L., Yan, J. & Wang, X. (2019). Video generation from single semantic label map, *arXiv:1903.04480.*

Payeur, P., Le-Huy, H. & Gosselin, C. (1995). Trajectory prediction for moving objects using artificial neural networks, *IEEE Transactions on Industrial Electronics, 42,* 147-158.

Redmon, J., Divvala, S.., Girshick, R. & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 779-788).

Ren, S., He, K., Girshick, R. & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39,* 1137-1149.

Rudenko, A., Palmieri, L, Herman, M., Kitani, K. M., Gavrila, D. M. & Arras, K. (2019). Human motion trajectory prediction: A survey, *arXiv:1905.06113.*

Siarohin, A., Woodford, O. J., Ren. J., Chai, M. & Tulyakov, S. (2021). Motion Representations for Articulated Animation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 13653-13662).

Srivastava, N., Mansimov, E. & Salakhutdinov, R. (2015). Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning* (pp. 843-852).

Tian, Z., Shen, C., Chen, H. & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *Proceedings of the International Conference on Computer Vision* (pp. 9627-9636).

Tulyakov, S., Liu, M.-Y., Yang, X. & Kautz, J. (2017). MoCoGAN: Decomposing motion and content for video generation, *arXiv:1707.04993.*

Vondrick, C., Pirsiavash, H. & Torralba, A. (2016). Generating videos with scene dynamics. In *Neural Information Processing Systems (NIPS) Conference* (pp. 613-621).

Wiest, J., Höffken, M., Kreßel, U. & Dietmayer, K. (2012). Probabilistic trajectory prediction with Gaussian mixture models. In *IEEE Intelligent Vehicles Symposium* (pp. 141-146).

Wu, J., Huang, Z., Dinesh, A., Li, W., Thoma, J., Paudel, D. P. & Van Gool, L. (2017). Sliced Wasserstein generative models, *arXiv:1706.02631.*

Yang, H.-K. & Chung W.-H. (2018). *Semantic segmentation tensorflow*. Available at: <https://github.com/hellochick/semantic-segmentation-tensorflow>, last accessed: 17th of February, 2022.

Yang, T.-J., Collins, M. D., Zhu, Y., Hwang, J., Liu, T., Zhang, X., Sze, V., Papandreou, G. & Chen, L.-C. (2019). DeeperLab: Single-shot image parser*, arXiv:1902.05093.*

Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J (2017). Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 6230-6239).

Zheng, C., Fan, X., Wang, C. & Qi, J. (2020). GMAN: A graph multi-attention network for traffic prediction. In *Proceedings of AAAI Conference on Artificial Intelligence, 34*(01), (pp. 1234-1241).