An Improved Vehicle Trajectory Prediction Model Based on Video Generation

David-Traian IANCU, Adina-Magda FLOREA

University Politehnica of Bucharest, 313 Splaiul Independenței, Bucharest, 060042, Romania
david_traian.iancu@upb.ro (**Corresponding author*), adina.florea@upb.ro

Abstract: The trajectory prediction task is one of the current challenges regarding computer vision and autonomous driving. However, unlike more common tasks like object detection or semantic segmentation, the problem is far to be resolved and there are still a lot of limitations for the prediction of the trajectory regarding the surrounding vehicles. Only a few models try to make trajectory prediction for a real-life application with live responses and there are only a few datasets to work with, the trajectory being hard to annotate. At least the limitation regarding the dataset annotation can be overcome by using a video generation-based model for trajectory prediction, considering that the video generation task does not require any special annotations. Following a recent study (Iancu et al., 2022), this work proposes some modifications to the PredNet architecture with better results for the trajectory prediction task.

Keywords: Trajectory prediction, Video generation, Autonomous driving, Convolutional neural networks, Long short-term memory networks, PredNet.

1. Introduction

One of the most interesting challenges in the latest years is to develop an autonomous car. There are five levels of car autonomy, ranging from only a simple assistance from the car regarding breaking, parking etc, up to the fifth level where the car could drive without any human assistance (even without a steering wheel or pedals, like the car developed by Google, Waymo). However, at this moment, only cars up to the third level of autonomy have been developed (Reyes, 2022). There are many problems that the research community must handle when designing an autonomous car. There are two different approaches for an autonomous car. The first one is an end-to-end approach, where ideally the network would take only driving images as input, along with some details regarding the acceleration, steering angle and other sensors, and give an immediate answer regarding the steering angle and the braking or acceleration level. However, given the complexity of this task, the preferred approach is the second one, with many different networks and components that work together and are supervised by the main algorithm. The system can be seen as a multi-layer architecture. The first layer, the perception layer, is responsible for scene understanding, considering the tasks of object detection and tracking, semantic and instance segmentation, lane detection, depth estimation. After the car has a complete representation of the environment, the next step is to decide the right actions that must be made - the decision layer. The software has to know the GPS coordinates of the car and the destination, and it makes a route based on this information. There is a global route planning and a local route planning and also a behaviour planning, regarding the manoeuvres hat have to be made. The process of following that route is called path following. The last component is responsible with the actual movement of the car – giving the steering angle and the braking/ acceleration to the physical components of the car.

In the path following process, an important element is represented by the prediction of the trajectory for the surrounding cars. This task is especially important for two reasons – avoiding a collision with another car and also following the car in front of the vehicle. One of the simplest, yet efficient algorithms for path following is to follow the first vehicle that is in front of the autonomous car and has the same behaviour – for example, if the next vehicle is going to break, the autonomous car should also break (with only one exception, where the next vehicle is going to stop, in this case it should be overtaken). However, the task of trajectory prediction is not a simple one and currently has a lot of limitations (Bahari et al., 2022). One of the biggest problems is that the annotation of the ground truth is very costly and implies the manual identification and annotation of the trajectories of the surrounding cars. As it can be seen in a recent study (Iancu et al., 2022), this limitation can be overcome if a video prediction model is used for finding the future positions of the cars, because a video prediction network can be trained using any existing driving video. However, the video prediction task is even harder than the task of trajectory prediction, therefore the results of a dedicated trajectory prediction network are better, for the moment.

https://doi.org/10.24846/v32i1y202303

In this work, an existing video generation model, PredNet (Lotter, Kreiman & Cox, 2016) is modified, taking into account the task of video generation, and the results are compared to those of the standard model in a trajectory prediction architecture composed of an object detection network, a semantic segmentation architecture for road detection and also a depth estimation network used in the evaluation process of the network performance.

The rest of the paper is organized as follows. In Section 2 presents the works from the specialized literature related to video prediction with a focus on the network used. Section 3 displays the material and methods. Discussions regarding the results of the present work are provided in Section 4, while Section 5 presents the conclusion and prospects for the development of the proposed research in some future works.

2. Related Work

The most relevant architectures regarding the task of predicting new frames for a given video are described in this section. The first distinction that has to be made is that there are actually two different tasks regarding the video prediction – the video generation and the video prediction. While the two tasks are related, there is an important distinction – a video generation network will make videos based on a training data that could be considered real videos from an observer and a video prediction network will generate the next frame(s) based on a real video. Regarding autonomous driving, the useful task is the later one. This section analyzes architectures from both tasks, considering the fact that the tasks are relatable and share the same architectures. A good review regarding video prediction can be found in a recent article (Iancu et al., 2022) and also in (Oprea et al., 2020).

The main video prediction architectures used in the present work are discussed in this section. Mainly, there are architectures based on Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), a combination of LSTM with CNN, Recurrent Neural Networks (RNN), Variational Autoencoders (VAE), Generative Adversarial Networks (GAN) or even o combination of GAN with VAE. Regarding trajectory prediction, there are different approaches that can be used with this purpose, either by using traditional architectures (CNNs, RNNs, LSTMs), sometimes in combination with a generic encoder-decoder architecture (Deo & Trivedi, 2018), or by using more recent architectures like GAN or VAE (Lee et al., 2022). Also, a common technique is to make only socially acceptable predictions (Kosaraju et al., 2019).

For reference, this work compares the results with TraPHic (Chandra et al., 2019), a convolutional LSTM architecture specialized in trajectory prediction.

2.1 Video Prediction using CNN

Even if the solely usage of convolutional neural networks (CNNs) is not common for video prediction, convolutional layers are widely used in combination with other architectures, especially in combination with LSTMs. However, there are some architectures that resemble the idea of CNNs. One relevant network is Dynamic Filter Network (De Brabandere et al., 2016). It is a new architecture which somehow resembles the transformer network – a transformation is ap plied to a feature map, by conditioning it on the input image. Their architecture consists of two different modules: a filter-generating network and a dynamic filter layer. The first module generates filters, based on some input, that are then applied on the image. This module is implemented using a convolutional network architecture, hence the relationship with the CNNs. The second module takes images and the previously generated filter, which is different regarding the location (dynamic filter). The filter, also, is similar to a classical convolutional filter, but is dynamically generated using the first module.

Another interesting work can be found in (Reda et al., 2018). The prediction is made using information from past frames and also past optical flows. Their method is called "spatially displaced convolution" and involves a convolution at a displaced location in the image. They use a fully convolutional neural network for the optical flow and for learning the parameters of the spatially displaced convolutions.

2.2 Video Prediction using LSTM

The Long Short-Term Memory Networks (LSTMs) are one of the favourite architectures for video prediction, due to their capacity of incorporating time changes. One of the oldest models for frame prediction can be found in (Srivastava, Mansimov & Salakhutdinov, 2015). The authors used an encoder-decoder LSTM. The encoder makes a representation of the input, while the decoder will decode this representation in order to make future frame predictions. They also vary the decoder by being conditioned on the previous frames or not. However, most of the networks also use convolutions, besides an LSTM network, due to the fact that the CNNs can model space-related features and the LSTMs can model time-related features. In (Wang et al., 2019) it can be seen a spatiotemporal LSTM that has been modified to use 3d-convolutions. Another example is Video pixel networks (Kalchbrenner et al., 2017). The architecture is based on an encoder decoder structure. Some convolutional operators are used for the encoder, whose outputs go into a convolutional LSTM. For the decoder they use an architecture called PixelCNN, which uses masked convolutions. In (Finn, Goodfellow & Levine, 2016) it can be seen an architecture for predicting future frames. They use different approaches for the video frame prediction, including spatial transformer predictions and also their own module called convolutional dynamic neural advection, which tries to output the locations in the next frame by using a distribution of the locations for the previous frame and also by using convolutions. They also used stacked convolutional LSTMs for their final model, which does action-conditioned prediction in videos. There are also more recent works that use convolutional LSTMs. For example, in (Desai et al., 2022) it can be found a model that uses an encoder decoder LSTM architecture for next frame prediction. The model used in the current experiments, PredNet (Lotter, Kreiman & Cox, 2016) is also a convolutional LSTM architecture with four components - a representation layer, which is a convolutional LSTM architecture, a convolutional target layer, a convolutional prediction layer and also an error layer. However, the architecture is discussed in detail in section 3, where the modifications made on the original architecture are also described.

2.3 Video Prediction using RNN

Another popular architecture used for video prediction is the recurrent neural network (RNN). One of the most cited architectures is the one from (Oliu, Selva & Escalera, 2017). The authors updated a traditional Gated Recurrent Unit, considering that the gate is, in fact, another recurrent unit. They applied this modification to recurrent auto-encoders (thus, they combined the RNN with the Autoencoders), making their modified GRU (Gated Recurrent Unit) as the shared state between the two components of the autoencoder (the encoder and the decoder), and claimed that the modification created a bijection between each input and output in a multi-layer architecture. They applied this architecture for video prediction on MNIST (Modified National Institute of Standards and Technology database) and KTH (KTH Royal Institute of Technology dataset).

Another interesting research work can be found in (Villegas et al., 2019). They questioned the traditional approaches for video prediction, which involved many components, such as segmentation, optical flow, etc, and proposed a stochastic approach, which considered that for some given past frames there are many possible future frames. They also added convolutional LSTM layers to an encoder-decoder architecture.

2.4 Video Prediction using VAE

Even if there are RNN networks for the video prediction task, they usually pair up the recurrent units with an encoder-decoder or autoencoder architecture. A more suitable architecture for generation tasks is the Variational Autoencoder (VAE). An example is Seg2Vid (Pan et al., 2019), which, as opposed to the previous network, adds an intermediate step for the optical flow and use a conditional VAE architecture (Xue et al., 2016). The authors made a video prediction from a single frame only, which is a method that can lead to more errors than including more frames in the prediction.

Another stochastic architecture is SAVP (Lee et al., 2018), which combines a generative adversarial network (GAN) with VAE. The traditional GAN model, with a generator and a discriminator, is modified to have two discriminators - a proper GAN discriminator and also a discriminator that learns from an encoder in a VAE model. The authors predicted future frames given two initial frames, however their model is tested on very simple datasets, like KTH (Schuldt, Laptev & Caputo, 2004) and moving robot arm (Ebert et al., 2017).

2.5 Video Prediction using GAN

One of the most popular architectures regarding different generation tasks is the generative adversarial network (GAN). The main issue regarding GANs is that there are usually used only for generation, not for predicting conditional frames given some past frames. There are many recent architectures that use this approach, for example Imaginator (Wang et al., 2020), which generates realistic videos given only a frame and a label. However, this approach is an improvement of two older models, VGAN (Vondrick, Pirsiavash & Torralba, 2016) and MoCoGan (Tulyakov et al., 2018). The first one, VGAN, is a very popular architecture for generating realistic videos. It uses a generator of two independent stacked up-sampled convolutional layers – one for the foreground and the other stream for the background. At the end, the two streams are combined to obtain the generated frame. The discriminator is a five-layer convolutional neural network. The layers generate small, realistic videos for different themes, for example a golf course or a train station. MoCoGan (Tulyakov et al., 2018) is a more recent architecture, which is cited by many research papers involving video prediction, due to its results. It involves a recurrent neural network, a generator used for the future frames, and two discriminators – one for a single frame and one for the video itself. It also generates simple clips, like facial expressions or human actions. However, one of the oldest architectures is the one proposed in (Mathieu, Couprie & LeCun, 2015). It is an influential work, by introducing the combination of a GAN with a multi-scale convolutional architecture for both the generator and the discriminator. Again, the network is tested on small clips. Even if the GAN could be considered the best architecture for generation, the current works obtains better results in real-life driving application by using convolutional LSTMs, which is also the architecture involved in this study.

3. Material and Methods

The most relevant methods used for video prediction were described in the previous network. The preferred approach for video prediction is done via neural networks, with most of the models using either a combination of recurrent layers (generally LSTMs) with convolutional layers or more recent approaches, like the GANs or the VAE. In this work, the network used is PredNet, a convolutional LSTM network, being in the first category of the described approaches.

For the trajectory prediction, this work also uses a popular convolutional neural network for object detection, YOLO v4 (Bochkovskiy, Wang & Liao, 2020), in order to detect the cars from the predicted frames. The detections are further improved with the help of the segmentation of another convolutional neural network, Fully Convolutional Networks (FCN) (Long, Shelhamer & Darrell, 2015).

The last network is used for depth estimation, namely it is used to compute some metrics regarding the predicted depth of the surrounding cars. The network used is Monodepth2 (Godard et al., 2018). For each model, the implementation was taken from its associated Github page. The training and the testing were made using an Nvidia DGX server.

4. Results

This section describes the results obtained regarding the new trajectory prediction model.

The first subsection describes the proposed architectures used for the video prediction task, whose output is later used to predict the trajectories of the surrounding vehicles. For the video prediction task, the experiments use three modified versions of the PredNet architecture and a convolutional LSTM architecture for video prediction. After the description of the architecture, the workflow regarding the trajectory prediction task is briefly discussed.

The second subsection describes the trajectory prediction workflow.

The third subsection describes the dataset used, the training and the experiments made for the trajectory prediction task.

The last subsection presents the experiments and also a qualitative analysis regarding the results.

4.1 Main Architecture

On a higher level, PredNet can be seen as multiple recurrent convolutional layers, whose output goes through a rectified linear unit (ReLU) activation and a max-pooling layer with stride 2. Now, regarding the convolutional recurrent layers, they consist of four different layers of convolutions. The first one is a representation layer, which is a recurrent layer that makes a prediction based on the current representation input. The input and the prediction represent another two layers of convolutions. The last layer is an error layer which is computed based on the input and the prediction and it becomes the next input layer. The representation layer at a given step is based on the representation layer at the previous step, the error layer at the previous step and also on the representation layer at the next step (which can be obtained initially by using upsampling). The main network is made in order to predict only a single future frame given an input video. However, the network can also be fine-tuned in order to predict up to five frames into the future. For the current experiments, the architectures were also fine-tuned to predict five future frames given only the initial video.

This research proposes three different versions of the internal representation of the convolutional layers. The standard version uses a four layer model with 3x3 convolutions for the prediction of driving images, as it can be seen in the public repository of Lotter, Kreiman & Kox (2016). The proposed models are the following:

The P_5_5 simply replaces the 3x3 convolutions with 5x5 convolutions, without adding any additional layers.

The P_3_5 is a 6-layer model with two extra 3x3 convolutional layers, considering the previous model, P_5_5. It also replaces the ReLU activation with PReLU, which instead of zeroing negative values it learns a parameter which is multiplied with the value for the response, acording to the following equation:

$$f(x) = \left\{ \begin{matrix} x, x >= 0 \\ a * x, otherwise \end{matrix} \right\} \qquad (1)$$

Finally, the P_full is also a 6-layer model consisting of only 3x3 convolutional layers and also using the PRELU activation function. The modifications can be seen in Figure 1.

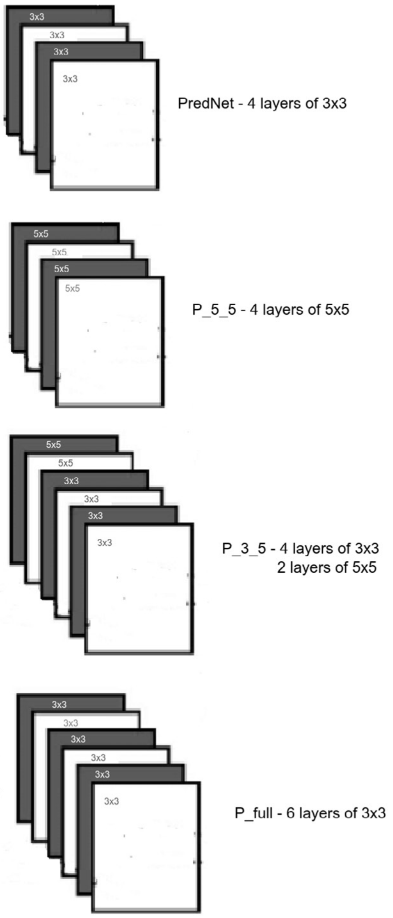


Figure 1. Proposed convolutional structures

4.2 Trajectory Prediction Workflow

The previous described architectures are used for the video generation task. The workflow for the trajectory prediction is presented in this subsection.

The input frames are going through one of the architectures described earlier after an additional pre-processing step, where the frame dimensions are adjusted to those required by the model. The output goes further through YOLO v4 (Bochkovskiy, Wang & Liao, 2020), to detect the objects in the frames. Together with the information obtained by a segmentation network, Fully Convolutional Networks (FCN) (Long, Shelhamer & Darrell, 2015), there are made predictions regarding the positions of the cars in the future frames. These positions are compared to the original ones, manually annotated, after which two errors are computed – a location dependent Normalized Root Mean Square Error (NRMSE) and a depth dependent NRMSE. The depth error is computed with the help of a depth estimation network, Monodepth2 (Godard et al., 2018). The full architecture can be seen in Figure 2.

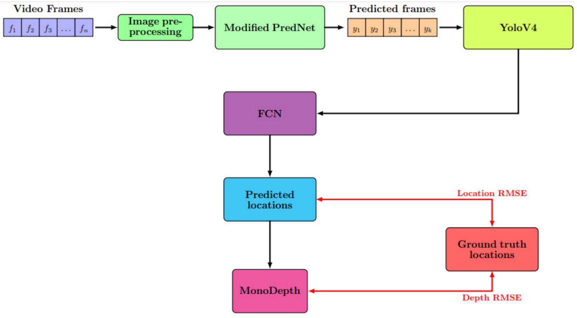


Figure 2. Trajectory prediction workflow

4.3 Training, Dataset and Experiments

The networks were trained on the KITTI dataset, in a similar way to the training that was made for the original network. The P_full and the P_3_5 architectures were trained for 150 epochs, the same number used for the basic PredNet network. However, the P_5_5 network was trained using 100 epochs, due to the long time required for the training. The P_5_5 network learned faster than the P_3_5, the final loss, 0.0229, being smaller than the final loss of the P_3_5 network, which was 0.0241. Initially, each network was trained to predict one frame, then it was trained again, using fine-tuning, in order to predict five frames in the future.

The dataset is the same one used in (Iancu et al., 2022) and it consists of over 4000 manually annotated cars in 945 images. The data is divided into small clips of 35 frames – 30 for training and 5 for testing.

Different experiments are made taking into account the data used for the segmentation. There are three different categories: the segmentation data is not included at all, the segmentation is obtained by the FCN network or the segmentation is taken from the ground truth images, which were manually annotated. The segmentation is used only for the detection of the road, which improves the prediction by making slight modification of the detected cars, in the case that they are outside the detected road. The segmentation is made only for the last real frame, not for the detected frames. Each of these three categories is further used in two different setups – the detected car locations are made considering the ground truth annotations of the cars in the predicted frames and the detected locations are based on the results from YOLO. Furthermore, for the depth there are two different setups – considering the estimated depth from the predicted frames and considering the depth from the real frames. For the segmentation, there are two setups – considering the ground truth or considering the results from FCN. In total, there are eighteen types of different experiments made for a given architecture and for each experiments the metrics are computed independently for the images recorded during the day, during the dusk and during the night, but also an average over the whole dataset is shown.

Unlike the previous study, which computes the RMSE for the location, taking into account the four corners of an object, and the RMSE for the depth, considering the average depth of the pixels in the predicted location compared to the average depth of the pixels in the actual location, in this research the RMSE was changed with normalized RMSE (NRMSE). This change was made for a better understanding of the metrics. In the following formula, the NRMSE is computed regarding the square root of the mean of the squared differences between the actual x and y coordinates (x_{ip} = predicted x coordintate, x_{ir} = real y coordinate, same for the y) for all the N detected objects. Then, the result is divided by the maximum possible value of the RMSE, which is computed by considering x_{ip}=y_{ip}=0, x_{ir}=640, y_{ir}=360 (given that the images are 640x360).

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

Regarding the depth the error was computed by taking into account the normalized value of the RMSE, too. The NRMSE for the depth is computed by taking the square root of the mean between the squared value of the difference between the mean value of the predicted N1 pixels and the real N2 pixels, considering all the N detected objects. Then, the result is divided by the maximum possible value of the depth RMSE, which is 256 (the value of the predicted pixels is 0, the value of the real pixels is 256). The NRMSE_depth has the advantage of estimating how close was the network to understand the actual distance from the ego vehicle to the surrounding cars.

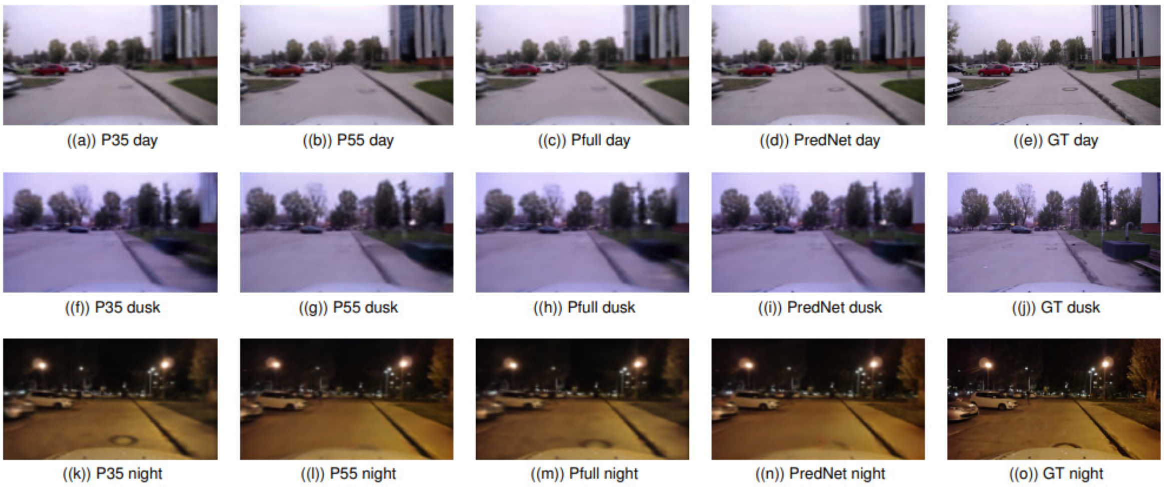
$$NRMSE_{Depth} = \frac{\sqrt{\frac{\sum_{j=1}^{N}\left(\left(\frac{\sum_{i=1}^{N1} PredictedDepthPixel_i}{N1}\right)-\left(\frac{\sum_{i=1}^{N1} RealDepthPixel_i}{N2}\right)\right)^2}{N}}}{Max\_RMSE_{Depth}} \quad (3)$$

4.4 Result Analysis

This subsection presents the most relevant results regarding the performances of the three architectures P_3_5, P_5_5 and P_full. The focus is put on the difference between these architectures and the basic PredNet network, not necessarily on the absolute values on the results, which are not very relevant regarding the comparison. In Figure 3 there are some images with the second predicted frame in different scenarios from each architecture, including the original PredNet architecture and the ground truth. The differences are not so obvious, given that the resolution of the images is small, but it can be seen that there are some small variations from the ground truth regarding different models. Also, the ground truth image is clearer. The images shown in this figure were taken from the same place, for a fair comparison regarding the time of the day. The results are presented in Table 1, Table 2 and also in Figure 4. The first table contains the NRMSE regarding the location for each of the eighteen setups and regarding the time of the day. The second table contains the NRMSE regarding the depth for the same setups as in the first table. Lastly, in Figure 4 it can be seen a plot for the NRMSE taking into account the car size, considering the last setup - the detections from YOLO and the segmentation from the FCN network.

Table 1. Location NRMSE

Model	Det.	Segm.	Day	Dusk	Night	Avg.
P.O.	GT	No	0.217	0.049	0.064	0.169
P.O.	GT	GT	0.216	0.048	0.064	0.168
P.O.	GT	FCN	0.216	0.049	0.064	0.168
P.O.	YOLO	No	0.191	0.132	0.032	0.184
P.O.	YOLO	GT	0.188	0.108	0.032	0.18
P.O.	YOLO	FCN	0.188	0.108	0.032	0.18
P35	GT	No	0.208	0.056	0.061	0.164
P35	GT	GT	0.207	0.056	0.061	0.17
P35	GT	FCN	0.207	0.056	0.061	0.163
P35	YOLO	No	0.218	0.238	0.063	0.216
P35	YOLO	GT	0.215	0.157	0.065	0.21
P35	YOLO	FCN	0.211	0.156	0.062	0.206
P55	GT	No	0.208	0.056	0.061	0.164
P55	GT	GT	0.207	0.056	0.061	0.17
P55	GT	FCN	0.207	0.056	0.061	0.163
P55	YOLO	No	0.218	0.238	0.063	0.216
P55	YOLO	GT	0.215	0.157	0.065	0.21
P55	YOLO	FCN	0.211	0.156	0.062	0.206
Pfull	GT	No	0.209	0.057	0.05	0.164
Pfull	GT	GT	0.209	0.057	0.049	0.164
Pfull	GT	FCN	0.209	0.057	0.05	0.164
Pfull	YOLO	No	0.187	0.207	0.033	0.186
Pfull	YOLO	GT	0.177	0.121	0.028	0.179
Pfull	YOLO	FCN	0.177	0.122	0.028	0.172
TP			0.057	0.037	0.078	0.059


((a)) P35 day ((b)) P55 day ((c)) Pfull day ((d)) PredNet day ((e)) GT day

((f)) P35 dusk ((g)) P55 dusk ((h)) Pfull dusk ((i)) PredNet dusk ((j)) GT dusk

((k)) P35 night ((l)) P55 night ((m)) Pfull night ((n)) PredNet night ((o)) GT night

Figure 3. Prediction results

Table 2. Depth NRMSE

Model	Det.	Segm.	Depth	Day	Dusk	Night	Avg.
P.O.	GT	NO	Real	0.555	0.034	0.092	0.409
P.O.	GT	NO	Pred	0.559	0.055	0.101	0.413
P.O.	GT	GT	Real	0.567	0.033	0.089	0.434
P.O.	GT	GT	Pred	0.569	0.051	0.094	0.436
P.O.	GT	FCN	Real	0.569	0.033	0.092	0.435
P.O.	GT	FCN	Pred	0.572	0.054	0.1	0.438
P.O.	YOLO	NO	Real	0.272	0.079	0.027	0.254
P.O.	YOLO	NO	Pred	0.659	0.097	0.003	0.613
P.O.	YOLO	GT	Real	0.243	0.076	0.025	0.23
P.O.	YOLO	GT	Pred	0.647	0.064	0.002	0.604
P.O.	YOLO	FCN	Real	0.255	0.076	0.023	0.241
P.O.	YOLO	FCN	Pred	0.622	0.076	0.002	0.583
P35	GT	NO	Real	0.555	0.041	0.083	0.418
P35	GT	NO	Pred	0.552	0.046	0.076	0.415
P35	GT	GT	Real	0.559	0.04	0.083	0.436
P35	GT	GT	Pred	0.556	0.045	0.076	0.433
P35	GT	FCN	Real	0.557	0.041	0.083	0.434
P35	GT	FCN	Pred	0.555	0.046	0.076	0.432
P35	YOLO	NO	Real	0.293	0.11	0.055	0.263
P35	YOLO	NO	Pred	0.699	0.168	0.087	0.623
P35	YOLO	GT	Real	0.264	0.13	0.065	0.246
P35	YOLO	GT	Pred	0.682	0.083	0.066	0.614
P35	YOLO	FCN	Real	0.241	0.116	0.05	0.241
P35	YOLO	FCN	Pred	2.401	0.073	0.07	0.577
P55	GT	NO	Real	0.561	0.046	0.063	0.416
P55	GT	NO	Pred	0.561	0.048	0.072	0.416
P55	GT	GT	Real	0.565	0.044	0.063	0.432
P55	GT	GT	Pred	0.562	0.046	0.072	0.43
P55	GT	FCN	Real	0.568	0.046	0.063	0.433
P55	GT	FCN	Pred	0.566	0.048	0.072	0.432
P55	YOLO	NO	Real	0.271	0.096	0.045	0.246
P55	YOLO	NO	Pred	0.633	0.092	0.052	0.568
P55	YOLO	GT	Real	0.254	0.095	0.041	0.235
P55	YOLO	GT	Pred	0.63	0.059	0.052	0.567
P55	YOLO	FCN	Real	0.257	0.095	0.041	0.237
P55	YOLO	FCN	Pred	0.593	0.063	0.052	0.536
Pfull	GT	NO	Real	0.544	0.036	0.071	0.41
Pfull	GT	NO	Pred	0.541	0.048	0.089	0.409
Pfull	GT	GT	Real	0.557	0.035	0.069	0.434
Pfull	GT	GT	Pred	0.553	0.045	0.083	0.432
Pfull	GT	FCN	Real	0.555	0.036	0.071	0.433
Pfull	GT	FCN	Pred	0.552	0.048	0.087	0.432
Pfull	YOLO	NO	Real	0.293	0.11	0.055	0.263
Pfull	YOLO	NO	Pred	0.66	0.129	0.048	0.584
Pfull	YOLO	GT	Real	0.272	0.116	0.051	0.25
Pfull	YOLO	GT	Pred	0.68	0.083	0.054	0.61
Pfull	YOLO	FCN	Real	0.237	0.112	0.046	0.237
Pfull	YOLO	FCN	Pred	0.64	0.069	0.066	

The tables are easy to understand. The first column contains the name of the model (P.O. stands for PredNet original and TP stands for TraPHic). The second column contains the method used for object detection – ground truth (GT) or YOLO. In the second table, an additional column contains the depth information used for the images. Real means that the depth values are taken from the depth estimation networks on the real images and "pred." means that the estimation was taken from the predicted frames. The next columns contain the results for day, dusk, night and an average for the whole dataset (avg.). The performances of each of the three proposed architectures are analyzed in the following paragraphs.

The first architecture proposed, P_3_5, obtains slightly better results than the basic model for the NRMSE considering the location and the ground truth detections.

However, for the YOLO detections the error is higher, which can be due to the errors introduced by YOLO.

The YOLO network obtained only 133 objects for this architecture, compared to 162, the number of the detections made in the original architecture, so this could partially explain the smaller results.

The P_5_5 architecture obtained 157 detected objects and P_full obtained 190 detections. The results don't take into account the recall, only the mean for the detected cars.

Considering the NRMSE for the depth, the numbers are almost similar with the ones of the basic model, with two exceptions only a little bit higher. The only exception is for the predicted depth, with FCN segmentation, regardless the way the detected objects are considered (ground truth or YOLO). Regarding the differences between the time of the day, all the networks are similar to the basic model – the worst results are during the day, the best during the dusk. However, this is due to a much bigger number of cars during the day which are harder to identify.

The second architecture proposed, P_5_5, obtains even better results for the NRMSE location than the P_3_5 considering the ground truth annotations of the cars, therefore better results than those of the the basic model.

The results regarding the YOLO detection are slightly better than the results from the P_3_5 architecture but unfortunately still worse than the results of the basic model.

Regarding the depth NRMSE, the network obtains comparable results, but this time the majority of the results are better than the ones of the basic model, including the most important experiment, which is the last one, because it can be used in real-life scenarios, without taking into account any ground truth or pre-existing frames.

The last architecture, P_full, consisting of 6 layers of 3x3 convolutions and with the PReLU activation, obtains the best results compared to all the previous ones, including the ones of the basic model. The results are better in all the experiments, with only one exception, for the experiment without segmentation and considering the YOLO detection. However, this result shows that taking into account the segmentation in order to adjust the positions of the predicted cars worked better for this architecture. Regarding the depth the results are slightly better than the ones of the basic model in almost all the experiments made, including the most relevant one with no ground truth. There are a few experiments, though, with a slightly better result for the basic model, but the difference is too small to be relevant.

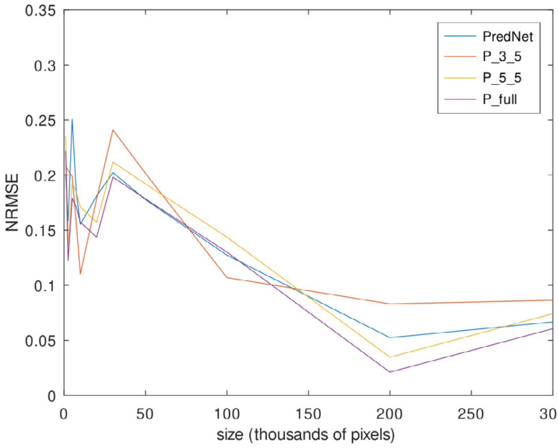


Figure 4. NRMSE with regard to car size

5. Discussion

As a conclusion regarding the experiments for the proposed architectures, P_full obtains the best results in almost all the experiments, P_5_5 obtains better results than the ones of the original networks regarding the location considering the ground truth detection and similar results regarding the depth NRMSE while P_3_5 obtains similar or even worse results in comparison with the ones of the PredNet, in all the experiments made. The rational conclusion for these experiments is that mixing 3x3 convolutions with 5x5 convolutions is not a good idea, however more convolutions mean better results and for the same number of layers. 5x5 convolutions were generally better for some scenarios than 3x3 convolutions.

Regarding the inference time, given that all the presented models are variations of the PredNet, the differences between them are not relevant. The speed for all the models is about 5 FPS, which can be used in real-life applications if one doesn't want to predict the trajectories in every frame. Also, the model depends on the speed of the segmentation and on the speed of the detection, but generally these operations will require less time and can be done in parallel.

Considering that autonomous driving is a complex topic, the results can be highly influenced by the dataset used. For this reason, this work uses frames involving different times of the day (day, dusk and night). However, the frames consist of images recorded in the campus of University POLITEHNICA of Bucharest.

A better approach would consist in testing many frames from different cities and in traffic conditions. However, this will lead to a big effort regarding the manual annotation of the data.

Regarding the trajectory prediction task, the current models generally obtain better results compared to the ones of the classical PredNet architecture. For this reason, instead of using the PredNet architecture, a new model for trajectory prediction that uses video prediction with the help of the P_full architecture, for exemple, can be employed.

6. Conclusion and Future Work

This study is a follow up of a previous trajectory prediction research and follows the same logic for the trajectory prediction architecture – it aims to predict future trajectories by using a video prediction network, instead of using a conventional architecture designed especially for the trajectory prediction task.

This approach is a new one and has the advantage that the training data can be easily obtained, basically by taking any driving video and considering some frames as the input and another few frames ahead as the desired output (ground truth). However, unlike the previous study, the current research also proposes three different architecture variations for a classical video prediction network called PredNet, by modifying its internal representation of the convolutional layers and also by using a different activation function, namely PReLU. This research presents an up-to-date analysis of the most important architectures regarding video prediction and generation, along with their advantages and disadvantages. It presents three different variations of the standard PredNet architecture, which are used in a framework made especially for trajectory prediction, which involves object detection, depth estimation and semantic segmentation. Different experiments are made using these architectures on a dataset containing images recorded in the University POLITEHNICA of Bucharest campus with different light conditions. The results are analyzed and compared with those of the standard PredNet architecture and also with those of a proper trajectory prediction network. The results show that the proposed models obtain better results than the classical PredNet architecture at least in some of the experiments made, which can help future researchers to develop better models for trajectory prediction, considering the video generation approach. The models can be updated in the future, considering that a trajectory prediction architecture still has better results.

Acknowledgements

Special thanks are due to the University POLITEHNICA of Bucharest which made this study possible. The research reported in this paper was supported by a grant of the Ministry of Research, Innovation and Digitalization, project CloudPRECIS, MySmis Code: 124812, within POC.

REFERENCES

Bahari, M., Saadatnejad, S., Rahimi, A., Shaverdikondori, M., Shahidzadeh, A. H., Moosavi-Dezfooli, S. M. & Alahi, A. (2022) Vehicle trajectory prediction works, but not everywhere. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, 19-24 June 2022, New Orleans, Louisiana.* Danvers, MA, The Institute of Electrical and Electronics Engineers, Inc. pp. 17123-17133.

Bochkovskiy, A., Wang, C.-Y. & Liao, H. M. (2020) *YOLOv4: Optimal speed and accuracy of object detection.* [Preprint] https://arxiv.org/abs/2004.10934 [Accessed: 20th January 2023].

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, CVPR 2019, 16-20 June 2019, Long Beach, California.* Danvers, MA, The Institute of Electrical and Electronics Engineers, Inc. pp. 8483-8492.

De Brabandere, B., Jia, X., Tuytelaars, T., & Gool, L. V. (2016) Dynamic filter networks. In: Lee, D., von Luxburg, U., Garnett, R., Masashi Sugiyama, M. and Isabelle Guyon, I. (eds.) *Advances in Neural Information Processing Systems 29: Proceedings of the 30th International Conference on Neural Information Processing Systems NIPS 2016, 5-10 December 2016, Barcelona, Spain.* NY, United States, Curran Associates, Inc. pp. 667–675.

Deo, N. & Trivedi, M. M. (2018) Convolutional social pooling for vehicle trajectory prediction. To be published in *CVPR TrajNet Workshop.* [Preprint] https://arxiv.org/abs/1805.06771 [Accessed: 20th January 2023].

Desai, P., Sujatha, C., Chakraborty, S., Ansuman, S., Bhandari, S & Kardiguddi, S. (2022) Next frame prediction using ConvLSTM. *Journal of Physics: Conference Series.* 2161 (1), 12-24.

Ebert, F., Finn, C., Lee, A. X. & Levine, S. (2017) Self-Supervised Visual Planning with Temporal Skip Connections. In: Lawrence, N. (ed.) *1st Annual Conference on Robot Learning: Proceedings of Machine Learning Research, CoRL2017, 13-15 November 2017, Mountain View, California*. Melville, NY, AIP Publishing. pp. 344-356.

Finn, C., Goodfellow, I. & Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In: Lee, D., von Luxburg, U., Garnett, R., Masashi Sugiyama, M. and Isabelle Guyon, I. (eds.) *Advances in Neural Information Processing Systems 29: Proceedings of the 30th International Conference on Neural Information*

Processing Systems, NIPS 2016, 5-10 December 2016, Barcelona, Spain. NY, United States, Curran Associates, Inc. pp. 64–72.

Godard, C., Aodha, O., Firman, M. & Brostow, G. (2018) Digging into self-supervised monocular depth estimation. In: *Proceedings of the 2018 IEEE/CVF International Conference on Computer Vision, ICCV, 18-22 June 2018, Salt Lake City, Utah.* Los Alamitos, California, Curran Associates, Inc. pp. 3828–3838.

Iancu, D. T., Nan, M., Ghita, S. A. & Florea, A. M. (2022) Trajectory Prediction Using Video Generation in Autonomous Driving. *Studies in Informatics and Control*. 31 (1), 37–48. doi:10.24846/v31i1y202204.

Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A. & Kavukcuoglu, K. (2017) Video pixel networks. In: Precup, D. and The, Y. W. (eds.) *Proceedings of the 34th International Conference on Machine Learning PMLR 2017, 06-11 August 2017, Sydney, Australia*. Sydney, NSW, Australia, Curran Associates, Inc. pp. 1771–1779.

Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H. & Savarese, S. (2019) Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. and Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32: Proceedings of the 30th International Conference on Annual Conference on Neural Information Processing Systems NIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. NY, United States, Curran Associates, Inc. pp. 137-146

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., & Levine, S. (2018). *Stochastic adversarial video prediction*. [Preprint] https://arxiv.org/abs/1804.01523 [Accessed: 20th January 2023].

Lee, M., Sohn, S. S., Moon, S., Yoon, S., Kapadia, M. & Pavlovic, V. (2022) MUSE-VAE: Multi-Scale VAE for Environment-Aware Long Term Trajectory Prediction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, 19-24 June 2022, New Orleans, Louisiana.* pp. 2221-2230.

Long, J., Shelhamer, E. & Darrell, T. (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, 7-12 June 2015, Boston, MA, USA*. pp. 3431-3440.

Lotter, W., Kreiman, G. & Kox, D. (2016) *Deep predictive coding networks for video prediction and unsupervised learning*. [Preprint] https://arxiv.org/abs/1605.08104 [Accessed: 20th January 2023].

Mathieu, M., Couprie, C. & LeCun, Y. (2015) *Deep multi-scale video prediction beyond mean square error.* [Preprint] https://arxiv.org/abs/1511.05440v1 [Accessed: 20th January 2023]

Oliu, M., Selva, J. & Escalera, S (2017) *Folded Recurrent Neural Networks for Future Video Prediction*. [Preprint] https://arxiv.org/abs/1712.00311 [Accessed: 20th January 2023]

Oprea, S., Martinez-Gonzalez, P., Garcia A., Castro-Vargas J. A., Orts-Escolano, S., Garcia-Rodriguez, J. & Argyros, A (2020) *A review on deep learning techniques for video prediction*. [Preprint] https://arxiv.org/abs/2004.05214 [Accessed: 20th January 2023]

Pan, J., Wang, C., Jia, X., Shao, J., Sheng, L., Yan, J. & Wang, X. (2019). Video generation from single semantic label map. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, 16–20 June 2019, Long Beach, California*. pp. 3733-3742.

Reda, F. A., Liu, G., Shih, K. J., Kirby, R., Barker, J., Tarjan, D., Tao, A. & Catanzaro, B. (2018) SDC-Net: Video prediction using spatially-displaced convolution. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y. (eds) *Computer Vision – ECCV 2018, Lecture Notes in Computer Science: Proceedings of the European Conference on Computer Vision, ECCV 2018, 8-14 September 2018, Munich, Germany.* Springer, Cham. pp. 718-733.

Reyes, A. (2022) Mercedes-Benz Wins World's First Approval For Level 3 Autonomous Cars: What's That Mean?. *Slash Gear*. https://www.slashgear.com/782536/mercedes-benz-wins-worlds-first-approval-for-level-3-autonomous-cars-whats-that-mean/ [Accessed: 25th July 2022].

Schuldt, C., Laptev, I. & Caputo, B. (2004). Recognizing human actions: a local SVM approach. In: *Proceedings of the 17th International Conference on Pattern Recognition, 26 august 2004, Cambridge, UK.* pp. 32-36.

Srivastava, N., Mansimov, E. & Salakhudinov, R. (2015) Unsupervised Learning of Video Representations using LSTMs. In: Bach, F. and Blei, D. (eds.) *Journal of Machine Learning Research Workshop and Conference Proceedings (JMLR W&CP): Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 7-9 July 2015, Lille, France.* pp. 843-852.

Tulyakov, S., Liu, M. Y., Yang, X. & Kautz, J. (2018) MoCoGAN: Decomposing Motion and Content for Video Generation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, *CVPR 2018, 18-23 June 2018, Salt Lake City, UT, USA*. pp. 1526-1535.

Villegas, R., Pathak, A., Kannan, H., Erhan, D., Le, Q. V. & Lee, H. (2019) High fidelity video prediction with large stochastic recurrent neural networks. In: Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. A. and Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32: Proceedings of the 30th International Conference* *on Annual Conference on Neural Information Processing Systems, NIPS 2019, 8-14 December 2019*, *Vancouver, Canada.* NY, United States, Curran Associates, Inc. pp. 81-91.

Vondrick, C., Pirsiavash, H. & Torralba, A. (2016) Generating videos with scene dynamics. In: Lee, D., von Luxburg, U., Garnett, R., Masashi Sugiyama, M. and Isabelle Guyon, I. (eds.) *Advances in Neural Information Processing Systems 29: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS 2016, 5-10 December 2016*, *Barcelona, Spain*. NY, United States, Curran Associates, Inc. pp. 613-621.

Wang, Y., Bilinski, P., Bremond, F., & Dantcheva, A. (2020) ImaGINator: Conditional Spatio-Temporal GAN for Video Generation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2020, 4–8 January 2022, Waikoloa, Hawaii*. pp. 1160-1169.

Wang, Y., Jiang, L., Yang, M. H., Li, L. J., Long, M., & Fei-Fei, L. (2019) Eidetic 3D LSTM: A model for video prediction and beyond. In: *Proceedings of the International Conference on Learning Representations*, *ICLR 2019, 6-9 May, 2019, New Orleans, United States.*

Xue, T., Wu, J., Bouman, K., & Freeman, B. (2016). Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In: Lee, D., von Luxburg, U., Garnett, R., Masashi Sugiyama, M. and Isabelle Guyon, I. (eds.) *Advances in Neural Information Processing Systems 29: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS 2016, 5-10 December 2016*, *Barcelona, Spain*. NY, United States, Curran Associates, Inc. pp. 91-99.