Deep Long Term Prediction for Semantic Segmentation in Autonomous Driving

Bidya Dash[1,2][0000−0002−8043−6669], Shreyas Bilagi[1][0009−0008−7999−1850], Jasmin Breitenstein[2][0000−0002−6427−3660], Volker Schomerus[1][0009−0009−5957−9989], Thorsten Bagdonat[1][0000−0002−4008−6721], and Tim Fingscheidt[2][0000−0002−8895−5041]

[1] Group Innovation, Volkswagen AG, Wolfsburg, 38440, Germany
{bidya.binayam.dash, shreyas.basavaraj.bilagi, volker.patricio.schomerus, thorsten.bagdonat}@volkswagen.de
[2] Institute for Communications Technology, Technische Universität Braunschweig, Schleinitzstraße 22, Braunschweig, 38106, Germany
{j.breitenstein, t.fingscheidt}@tu-bs.de

Abstract. Temporal prediction is an important function in autonomous driving (AD) systems as it forecasts how the environment will change and transform in the next few seconds. Humans have an inherited prediction capability that extrapolates a present scenario to the future. In this paper, we present a novel approach to look further into the future using a standard semantic segmentation representation and time series networks of varying architectures. An important property of our approach is its flexibility to predict an arbitrary time horizon into the future. We perform prediction in the semantic segmentation domain where inputs are semantic segmentation masks. We present extensive results and discussion on different data dimensionalities that can prove beneficial for prediction on longer time horizons (up to $2s$). We also show results of our approach on two widely employed datasets in AD research, i.e., Cityscapes and BDD100K. We report two types of mIoUs as we have investigated with self generated ground truth labels (mIoU^{seg}) for both of our dataset and actual ground truth labels (mIoU^{gt}) for a specific split of the Cityscapes dataset. Our method achieves 57.12% and 83.95% mIoU^{seg}, respectively, on the validation split of BDD100K and Cityscapes, for short-term time horizon predictions (up to $0.2s$ and $0.06s$), outperforming the current state of the art on Cityscapes by 13.71% absolute. For long-term predictions (up to $2s$ and $0.6s$), we achieve 37.96% and 63.65% mIoU^{seg}, respectively, for BDD100K and Cityscapes. Specifically on the validation split of Cityscapes with perfect ground truth annotations, we achieve 67.55% and 63.60% mIoU^{gt}, outperforming current state of the art by 1.45% absolute and 4.2% absolute with time horizon predictions up to $0.06s$ and $0.18s$, respectively.

Keywords: Cityscapes · BDD100K · forecasting · prediction · long term prediction · semantic segmentation.

1 Introduction

Temporal prediction and forecasting has been an important task in intelligent systems and robotic decision making [2,5]. Simple tasks like object detection and tracking have been quite well investigated with deterministic approaches such as the Kalman filter [9] and dense optical flow techniques [14]. Non-deterministic learning based approaches [1, 19, 20, 24] have proved to be better and more adapted to these tasks in the long run.

Recent advancements using image-based prediction [4,10] and reconstruction have captured attention as an integral part in intelligent autonomous driving (AD) systems. Image-based prediction means forecasting the RGB pixels of a frame in a video sequence to their anticipated future positions in a future video frame in the pixel space. However, there are certain limitations to image-based prediction as it becomes more of a reconstruction task [4,16] (where positions of objects are regenerated) than prediction of actual motion when using deep neural representation models. Also, the predicted RGB pixel domain provides much irrelevant information for decisions in AD scenarios, whereas, the trajectory planner in an AD system can benefit from more information than just RGB pixels for determining possible obstacles and freespace.

Here, semantic segmentation provides a relevant representation as semantically segmented maps provide concise but relevant information not only about the possible obstacles and freespace; but also an extensive distribution of semantically discrete objects with respect to their pixel occupancy in a video frame. As a result, the trajectory planner in an AD system can process more relevant information for efficient decision making. Hence, prediction of semantically segmented frames into the future proves to be much more of a sensible task than temporal prediction of RGB frames. An interesting investigation in the field of prediction of semantic segmentation masks is estimating the model's performance for longer time horizons, because it is essential for AD systems to accommodate the length that these prediction models can forecast without significant deprecation in performance.

We can summarize our main contribution as follows. First, we propose an efficient time series network with an auto regressive gradient accumulation technique for forecasting of semantic segmentation maps. The time horizon prediction during inference is independent of the training time horizon range unlike Nabavi et al. [18] where they input 4 frames as a group concurrently. Secondly, we investigate using the semantic segmentation masks only as input to our prediction method while determining which semantic segmentation representation provides the most promising results. Thirdly, we show that our prediction method outperforms the current state of the art on the Cityscapes dataset [3], and additionally we are the first to report our results on BDD100K [26] which contains scenes from all types of weather conditions and time of the day. The remainder of this work is structured as follows: We present related works in section 2, followed by explaining the approaches in section 3. Section 4 introduces the experimental setup including architectural details, dataset, metrics followed by implementation details. Section 5 reports the experimental results along with

detailed discussion before concluding the paper in Section 6. We also provide a supplementary material with detailed analysis of qualitative results along with a special investigation of the proposed method's behavior.

2 Related Work

Video Prediction: By introducing a 3D optical flow representation across spatial and temporal dimensions along with trilinear interpolation, Liu et al. [11] proposed an unsupervised model to predict frames for video predictions. PredNet [12] employs predictive coding with local predictions to learn future frames and also propagating the deviations from subsequent layers in an unsupervised fashion. Similarly, Walker et al. [23] propose a PixelCNN approach in addition to discretizing the hierarchy of spatiotemporal self-attention latent space in video data using VQ-VAE. Mathieu at al. [17] propose a multi-scale architecture and an adversarial training strategy along with a novel image gradient divergence loss function to enhance frame predictions over longer time horizons. Using adversarial training, retrospective cycle GANs [10] have proved to be useful for predicting video frames while enforcing the consistency of bi-directional time horizons. Guen at al. [6] propose a method where they leverage the physical knowledge described by partial differential equations dynamics to disentangle unknown complementary information in video sequences. Our work follows along the domain of prediction of frames in a video sequence using a time series network. However, in this work, we solely focus on prediction in the semantic segmentation domain for an enriched information processing that can be useful for planning future scenarios in AD systems.

Prediction in Semantic Segmentation: Luc et al. [13] came up with an autoregressive multi scale region proposal CNN based on Mathieu et al.'s [17] backbone architecture using a generative adversarial loss combined with an image gradient difference loss to predict future scenes that are semantically segmented which proves to be a reconstruction technique rather than an actual prediction. In this method, they use varying combinations of RGB frames and semantic segmentation maps together interchangeably as input rendering the input representation highly complex whereas, we investigate using the semantic segmentation masks only as input to our prediction method while determining which semantic segmentation representation provides the most promising results. Nabavi et al. [18] use a `PSPNet` backbone [28] and bi-directional convolutional LSTMs [22] to predict latent space embeddings in the residual layers. With increasing time horizon predictions (i.e., $1s...2s$) into the future, these models reveal significant deviations to the ground truth. This can be attributed to the fact that the time series network used by Nabavi et al. [18] accumulates information only for a limited time span for highly dense class distributions that can be safely labeled as background (i.e., road, buildings, sidewalk, vegetation) and completely overturns the underrepresented classes distributions (i.e., pedestrian, bicycle, traffic lights). Exploiting the mutual benefits of predicting pixel

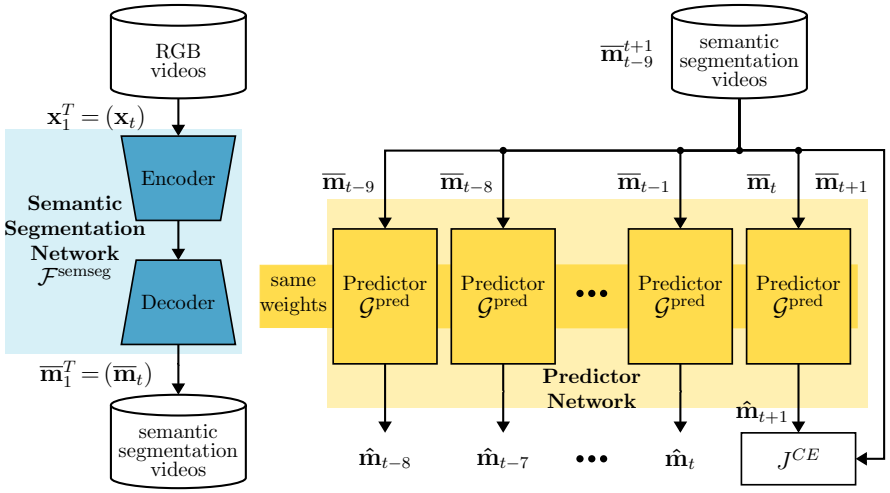


Fig. 1. The left side (in blue) depicts the step of generating pseudo ground truth semantic segmentation masks $\overline{\mathbf{m}}_1^T = \{\overline{\mathbf{m}}_1, \overline{\mathbf{m}}_2, ..., \overline{\mathbf{m}}_T\}$ from raw RGB videos as using a strong semantic segmentation network, i.e., `PSANet`. The right side depicts the prediction step, where a time series network receives a ten-frame mask sequence $\overline{\mathbf{m}}_{t-9}^t$ predicting $\hat{\mathbf{m}}_{t+1}$ from $\overline{\mathbf{m}}_t$ directly step by step in the semantic segmentation domain. The two steps are independent of each other and are performed one after the other with different loss representations.

annotations and dense optical flow estimations, Jin et al. [8] attempt to simultaneously model future semantic segmentation masks along with optical flow representations which proves to be quite useful for different time horizons and input resolutions. While we build on Nabavi et al.'s work [18], we get rid of the dependence of input sequence length and introduce an autoregressive frame prediction technique during inference along with predicting for longer time horizons in the future. Additionally, we report our results on BDD100K [26] dataset which is more challenging for the task of prediction.

3 Method

3.1 Prediction of Semantic Segmentation Masks

In Fig 1, we can see the details of our training methodology. There are two separate integral steps. For the first step (left, in blue), we feed in the raw RGB frames $\overline{\mathbf{x}}_t \in [0,1]^{H \times W \times C}$ where $t \in \mathcal{T} = \{1, 2, ..., T\}$, with T being the the total number of frames in the input video and t denoting the temporal frame index of the video sequence. This is fed to a standard semantic segmentation network $\mathcal{F}^{\text{semseg}}$, i.e., `PSANet` [29]. In Fig. 1 (left, in blue), we can see the semantic segmentation network $\mathcal{F}^{\text{semseg}}(\mathbf{x}_t; \boldsymbol{\theta}^{\text{semseg}})$ whose output is $\overline{\mathbf{y}}_t = (y_{t,i,s}) \in [0,1]^{H \times W \times S}$ where $\boldsymbol{\theta}^{\text{semseg}}$ denotes the semantic segmentation

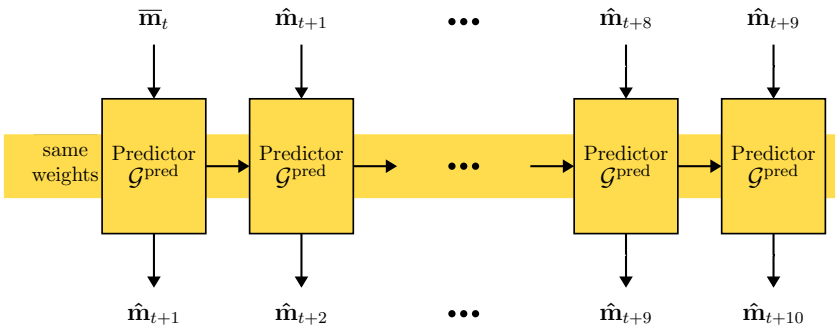


Fig. 2. This figure depicts the inference process for $\Delta t = 10$ time steps ahead using our approach. The predictor starts with the present time instant input $\overline{\mathbf{m}}_t$, generates a prediction $\hat{\mathbf{m}}_{t+1}$, uses this prediction as input to the next time step to produce $\hat{\mathbf{m}}_{t+2}$ and so on. Hence, to predict $\hat{\mathbf{m}}_{t+10}$, the model would use its own predictions $\hat{\mathbf{m}}_{t+1}$, $\hat{\mathbf{m}}_{t+2}$, ..., $\hat{\mathbf{m}}_{t+9}$ as input sequentially.

network's trainable parameters, $s \in \mathcal{S}$ represents the class in dataset with a total of S semantic classes and $i \in \mathcal{I} = \{1, 2, ..., H \cdot W\}$ represents the pixel indices for frames with height H and width W. The semantic segmentation network is entirely responsible for generating the semantic segmentation masks $\overline{\mathbf{m}}_t = (\overline{m}_{t,i})$ where $\overline{m}_{t,i} = \underset{s \in \mathcal{S}}{\text{argmax}}\ \overline{y}_{t,i,s}$. We save these pseudo ground truth semantic segmentation masks $\overline{\mathbf{m}}_1^T$ to train our predictor network. In Fig. 1 (right, in yellow), we can see that $\overline{\mathbf{m}}_{t-9}^t = \{\overline{\mathbf{m}}_{t-9}, ..., \overline{\mathbf{m}}_{t-1}, \overline{\mathbf{m}}_t\}$ is the input sequence fed sequentially as $\overline{\mathbf{m}}_{t-9}$, $\overline{\mathbf{m}}_{t-8}$, ..., $\overline{\mathbf{m}}_t$ to the predictor network, $\mathcal{G}^{\text{pred}}$ whose internal hidden states and cell states are $\mathbf{H}_t, \mathbf{C}_t$ at current time-step t. The predictor network is represented as $\mathcal{G}^{\text{pred}}(\overline{\mathbf{m}}_t, \mathbf{H}_t, \mathbf{C}_t; \boldsymbol{\theta}^{\text{pred}})$ whose output is $\hat{\mathbf{m}}_{t+1} = (\hat{m}_{t+1,i})$ where $\hat{m}_{t+1,i} = \underset{s \in \mathcal{S}}{\text{argmax}}\ \hat{y}_{t+1,i,s}$ where $\hat{\mathbf{y}}_{t+1} \in [0,1]^{H \times W \times S}$ denotes the normalized probabilistic output of the predictor network at timestep $t+1$. Here, $\boldsymbol{\theta}^{\text{pred}}$ denotes the predictor's trainable parameters.

The generated semantic segmentation masks from $\mathcal{F}^{\text{semseg}}$ are fed as 10-frame long sequences $\overline{\mathbf{m}}_{t-9}^t = \{\overline{\mathbf{m}}_{t-9}, ..., \overline{\mathbf{m}}_{t-1}, \overline{\mathbf{m}}_t\}$ to our predictor one by one which update the intermediate hidden state representations. The loss representation $\mathcal{J}^{CE}$ (categorical cross entropy loss) is always calculated with the 10th frame prediction i.e., $\hat{\mathbf{m}}_{t+1}$ and corresponding pseudo ground truth $\overline{\mathbf{m}}_{t+1}$. Note that both training steps are performed independently of each other with different loss representations.

3.2 Inference Processing

We can see the inference step of our method in Figure 2 which is at the core of our work. Hence, understanding the inference approach would give us a better idea of the proposed model's ability to look efficiently into the future. We incorporate

the auto regressive approach to predict longer time horizons. The input sequence is 10 frames long represented as $\overline{\mathbf{m}}_{t-9}^{t}$ and predicts $\hat{\mathbf{m}}_{t+1}$. To predict for more than one time step ahead, this prediction $\hat{\mathbf{m}}_{t+1}$ is once again fed as input to the predictor to produce $\hat{\mathbf{m}}_{t+2}$ that is $\Delta t = 2$ time steps ahead. We choose an arbitrary length of 10 time steps to be predicted in the future and hence, this process is repeated until $\Delta t = 10$ time steps ahead.

Note that, our method only looks at the first 10 pseudo ground truth frames of a sequence ($\overline{\mathbf{m}}_{t-9}^{t}$) and can predict up to an arbitrary number of frames, Δt, into the future. This is specifically useful in AD applications where we can set the ego vehicle to warm up for a certain length of sequences and then predict into the future arbitrarily.

4 Experimental Setup

In this section, we introduce the settings for our experiments including a new architecture for defining the predictor $\mathcal{G}^{\text{pred}}$ in subsection 4.1. We investigate different arrangements of temporal blocks along with introducing the datasets used for our experiments and the most important metrics.

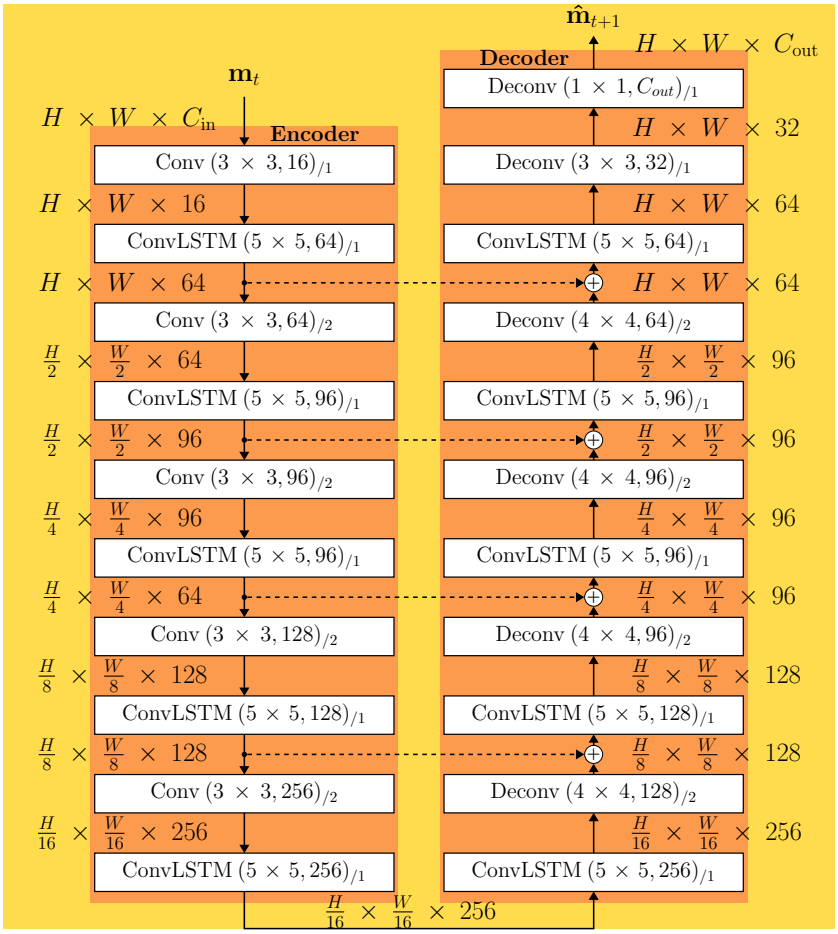
4.1 Predictor Architecture

For our predictor network $\mathcal{G}^{\text{pred}}$, we use convolutional LSTM [22] blocks along with a normalization layer to keep the training stable. We investigate three different arrangements of the convolutional LSTM blocks within $\mathcal{G}^{\text{pred}}$, which are shown in Figure 3.

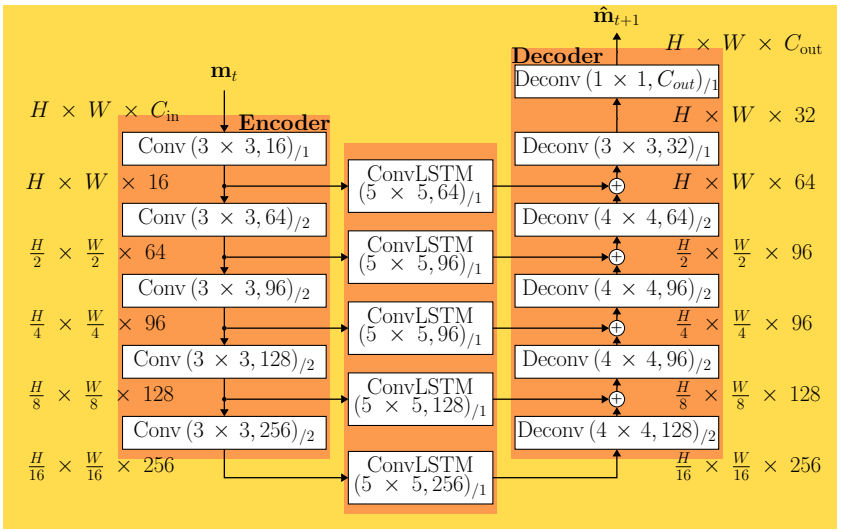
The first architecture, defined as `PRED` can be seen in Fig. 3(a) without the residual connections.`PRED` stacks up the convolutional LSTM blocks with a group normalization layer for normalizing the four intermediate activations: input gate, forget gate, cell gate and output gate. As described in Shi et al. [22], we follow the standard definition of a convolutional LSTM block which replaces the matrix-vector multiplications in the input-hidden and hidden-hidden mappings of a fully-connected LSTM [7] with convolutions, whereas keeping the general structure of the LSTM cell unchanged. The governing equations can be described as

$$
\begin{aligned}
\mathbf{I}_t &= \boldsymbol{\sigma}(\mathbf{W}^{I,ih} * X_t + \mathbf{W}^{I,hh} * \mathbf{H}_{t-1} + \mathbf{B}^{I}) \\
\mathbf{F}_t &= \boldsymbol{\sigma}(\mathbf{W}^{F,ih} * X_t + \mathbf{W}^{F,hh} * \mathbf{H}_{t-1} + \mathbf{B}^{F}) \\
\mathbf{O}_t &= \boldsymbol{\sigma}(\mathbf{W}^{O,ih} * X_t + \mathbf{W}^{O,hh} * \mathbf{H}_{t-1} + \mathbf{B}^{O}) \\
\mathbf{C}_t &= \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tanh(\mathbf{W}^{C,ih} * X_t + \mathbf{W}^{C,hh} * \mathbf{H}_{t-1} + \mathbf{B}^{C}) \\
\mathbf{H}_t &= \mathbf{O}_t \odot \tanh(\mathbf{C}_t).
\end{aligned}
\tag{1}
$$

where $\odot$ denotes element-wise multiplication, $\sigma(\cdot)$ denotes the element-wise applied sigmoid activation. $\mathbf{I}_t, \mathbf{F}_t, \mathbf{O}_t, \mathbf{C}_t, \mathbf{H}_t$ and $\mathbf{X}_t$ are tensors for values of input gate, forget gate, output gate, cell state, hidden state and input respectively, at



(a)



(b)

Fig. 3. (a): This figure depicts the `PRED+R` architecture for the predictor. By removing the residual connections (dashed lines), we obtain the `PRED` architecture. (b): This figure depicts the `PRED+RS` architecture for the predictor. Here, the ConvLSTM units are placed in the residual feature space connecting the encoders and decoders.

time step t. The tensors $\mathbf{W}^{Z,ih}$, $\mathbf{W}^{Z,hh}$ and $\mathbf{B}^{Z}$ where $Z \in \{I, F, O, C\}$ contain the kernel weights for input-hidden ($\mathbf{W}^{Z,ih}$), hidden-hidden ($\mathbf{W}^{Z,hh}$) mappings and bias values ($\mathbf{B}^{Z}$) for the input gate ($Z = I$), forget gate ($Z = F$), output gate ($Z = O$) and cell state ($Z = C$) computations respectively. For the first time step, the hidden states ($\mathbf{H}_t, \mathbf{C}_t$) are always set to zero tensors.

Every convolution layer consists of a leaky ReLU activation [15] function with a slope of -0.2. The architecture consists of an encoder part which extracts the spatio-temporal fashion with increasing receptive size for each consecutive layer. As shown in Fig 3(a), the input and output data resolutions are denoted as $H \times W \times D$ where H, W denote the spatial resolution and D represents the channel depth of the output. Similarly, $\text{Conv}(K \times K, D_{out})_{/M}$ represents a convolutional layer with kernel size of $K \times K$, output depth of channels as D_{out} with a stride of M. The convolutional LSTM layers are denoted as $\text{ConvLSTM}(K \times K, D_{out})_{/M}$ where the denotions are same as defined before. For the decoder part in Fig. 3(a), there are corresponding deconvolutional layers with transpose convolutional LSTMs. $\text{Deconv}(K \times K, D_{out})_{/M}$ denotes a deconvolutional layer with kernel size of $K \times K$, input and output depth of channels as D_{out} with a stride of M. The convolutional LSTM layers of the decoder are denoted in the same way.

We investigate a second type of predictor architecture, `PRED+R`, which also can be seen in Fig. 3(a). It is similar to `PRED` but with the residual connections (dashed lines) after every ConvLSTM block connected in an U-Net [21] fashion with the respective group of Deconv-ConvLSTM block. This is done to overcome the generic problem of vanishing gradients as well as facilitate better flow of gradients during training phase.

Fig. 3(b) shows our third proposed predictor architecture, `PRED+RS`. It is a more intuitive method with convolutional LSTM blocks present in between the corresponding convolution - deconvolution latent space. This can be seen in Fig 3(b) where the convolutional LSTM units are predicting the hidden state representations in the residual connections.

4.2 Datasets

Cityscapes: The Cityscapes [3] dataset $\mathcal{D}^{\text{CS}}$ is specifically tailored for AD scenes in urban setting. The dataset contains 5000 images for semantic segmentation. The dataset is split into 2975 images for training ($\mathcal{D}^{\text{CS}}_{\text{train}}$), 500 images for validation ($\mathcal{D}^{\text{CS}}_{\text{val}}$) and 1525 images for testing ($\mathcal{D}^{\text{CS}}_{\text{test}}$). The training split and the validation split have corresponding perfect ground truths available. For each image in $\mathcal{D}^{\text{CS}}$, there exists a 30 frame long video sequence whose 20^{th} frame annotations are available. To distinguish, we denote the dataset consisting of the entire videos as $\mathcal{D}^{\text{CS}-\text{vid}}$. This dataset accordingly contains 2975 videos for training ($\mathcal{D}^{\text{CS}-\text{vid}}_{\text{train}}$), 500 videos for validation ($\mathcal{D}^{\text{CS}-\text{vid}}_{\text{val}}$) and 1525 videos for testing ($\mathcal{D}^{\text{CS}-\text{vid}}_{\text{test}}$). Each video sequence contains 30 frames lasting 1.8s (16.67fps) long with a resolution of 1024×2048. Given the 16.67fps frame rate of Cityscapes and our arbitrary choice of predicting $\Delta t = 10$ time steps in the future, our setup predicts $0.6s$ into the future. There are $S = 19$ semantic class categories.

BDD100K: The BDD100K [26] dataset $\mathcal{D}^{\text{BDD}}$ contains randomly sampled images from 10,000 video clips with perfect semantic segmentation ground truths. This subset is split into 7000 images for training ($\mathcal{D}^{\text{BDD}}_{\text{train}}$), 1000 images for validation ($\mathcal{D}^{\text{BDD}}_{\text{val}}$) and 2000 images for testing ($\mathcal{D}^{\text{BDD}}_{\text{test}}$). For the purpose of prediction, we use the video dataset from multiple object tracking and segmentation $\mathcal{D}^{\text{BDD-MOTS}}$ subset containing 223 videos in total with 154 training $\mathcal{D}^{\text{BDD-MOTS}}_{\text{train}}$, 32 validation $\mathcal{D}^{\text{BDD-MOTS}}_{\text{val}}$ and 37 testing videos $\mathcal{D}^{\text{BDD-MOTS}}_{\text{test}}$. Each video contains about 200 frames (5fps) lasting 40s with a resolution of 720×1280. Given the 5fps frame rate of $\mathcal{D}^{\text{BDD-MOTS}}$ and our arbitrary choice of predicting $\Delta t = 10$ time steps in the future, our setup predicts $2s$ into the future. There are $S = 19$ semantic class categories.

4.3 Input Representations

Different types of input modalities were exploited by the model to predict future scenes using hidden state representations. Usually, most works focus only on the raw semantic segmentation masks $\overline{\mathbf{m}}_t \in \mathcal{S}^{H \times W}$ which can be interpreted as 1-channel input [8, 13, 18], i.e., $\overline{\mathbf{m}}_t \in [0, 1]^{H \times W \times 1}$. This is the most prevalent input representation for prediction tasks because it is computationally less expensive and independent of predefined semantic classes in the dataset. We also conduct some investigations with 1-channel input $\overline{\mathbf{m}}_t$ to prove our model's robustness, which are explained in detail in the supplementary material. However, in the process we often ignore the amount of information contained in the normalized probabilistic outputs of the semantic segmentation network, i.e., softmax activations just before the argmax function. Hence, we also conduct extensive experiments on these softmax activations as input data representations $\overline{\mathbf{y}}_t \in [0, 1]^{H \times W \times S}$, i.e., S-channel input. One advantage of using the S-channel input $\overline{\mathbf{y}}_t$ is that all the semantic classes are equidistant to each other in the latent space. In the process, we do prove that the S-channel softmax activation $\overline{\mathbf{y}}_t$ as input representation leads to better performance in prediction for both short-term and long-term time horizons ($\Delta t = 1, 2, 5, 10$).

4.4 Metrics

For reporting the quantitative performance of our experiments, we resort to estimating the mean intersection over union or Jaccard index. This metric is usually an indispensable estimation technique for most semantic segmentation and object detection models [25, 27]. As indicated by the name, the intersection over union (IoU) is computed as area of intersection (overlap) divided by the area of union. For the application of semantic segmentation and prediction, it is more appropriate to express IoU based on sensitivity and specificity indicators. Given the amount of true positives TP_s, false positives FP_s and false negatives FN_s for class s respectively, the IoU can be defined as

$$\text{IoU}_s = \frac{TP_s}{TP_s + FP_s + FN_s} \tag{2}$$

The overall mean performance over all the classes $\mathcal{S} = \{1, 2, ..., S\}$, i.e., mIoU can be represented as

$$\text{mIoU} = \frac{1}{S} \sum_{s=1} IoU_s \tag{3}$$

From now on, we will report two types of mIoUs: $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$, which is the mIoU between the predictions and self generated ground pseudo truth semantic segmentation masks. And $\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$, which is the mIoU between the predictions and human annotated (perfect) ground truths of $\mathcal{D}^{\text{CS-vid}}_{\text{val}}$ for the 20^{th} frame.

4.5 Implementation Details

Generating ground truth annotations: As both $\mathcal{D}^{\text{BDD-MOTS}}$ and $\mathcal{D}^{\text{CS-vid}}$ do not have off-the-shelf semantically annotated masks, we create our own ground truth labels using a standard semantic segmentation model. The semantic segmentation network $\mathcal{F}^{\text{semseg}}$, i.e., `PSANet` [29] is trained on $\mathcal{D}^{\text{BDD}}_{\text{train}}$ and $\mathcal{D}^{\text{CS}}_{\text{train}}$ dataset first. The annotations are produced via supervised learning using human annotated ground truth semantic segmentation masks with a categorical cross entropy loss representation. For $\mathcal{D}^{\text{BDD}}_{\text{train}}$, the original image resolution is of size 720×1280 whereas for $\mathcal{D}^{\text{CS}}_{\text{train}}$, it is 1024×2048. To arrive at a unified input resolution for training, we downsample the original images to 513×1025 which is a requirement to the choice of our semantic segmentation network due to affine transformations [29]. The output of this network is again downsampled via bilinear interpolation to produce semantic segmentation masks of size 128×256. The semantic segmentation network $\mathcal{F}^{\text{semseg}}$ is trained for 1000 epochs with a batch size of 4, learning rate of 1×10^{-2} and an auxiliary weight of 0.4. This setup uses a SGD optimizer with a weight decay of 5×10^{-4}. Our implementation of the semantic segmentation network, i.e., `PSANet` with a `ResNet50` backbone achieves an mIoU of 73.44% on $\mathcal{D}^{\text{CS}}_{\text{val}}$ and 59.84% on $\mathcal{D}^{\text{BDD}}_{\text{val}}$. The trainings were carried out on a single Tesla V100 GPU for 7 days.

Training the predictor: The three predictor architectures (`PRED`, `PRED+R`, `PRED+RS`) have been trained with the same hyperparameters for 100 epochs with an early stopping for plateau conditions. The input resolution of the semantic segmentation masks is 128×256 as mentioned above as this resolution proves to be the most optimal for our predictor architecture in terms of training time, memory consumption and storage constraints. As shown in Fig. 1, the inputs are 10 frame long sequences. The sequences are fed as single mask inputs at each timestep and the predictor's hidden state representations are updated sequentially according to Eq 1. The output is of the same size as input with a prediction time horizon of $\Delta t = 1$ timestep ahead. The batch size is set to 4 with a base learning rate of 1×10^{-3} and a learning rate scheduler with a factor of 0.5. Here, we use the Adam optimizer with a weight decay of 5×10^{-4}. The trainings were carried out on a single Tesla V100 GPU for 3 days. It is to be noted that all the reported numbers with mean and standard deviation have been averaged over 3 different seeds.

Table 1. $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ performance for different predictor architectures on $\mathcal{D}^{\text{BDD-MOTS}}_{\text{val}}$ showing different input types as well as time horizons into the future. The best architecture per time horizon is denoted in **bold** font. Also, the best input type can be seen underlined. All the experiments have been averaged over 3 seeds except `CP` and `OF` as they are deterministic, n/a - not applicable.

Input Type	Time Horizon Δt	Methods				
		`CP`	`OF`	`PRED`(ours)	`PRED+R`(ours)	`PRED+RS`(ours)
1-channel	$1(0.2s)$	49.32	51.75	54.96 ± 0.15	$\mathbf{55.20} \pm 0.23$	51.66 ± 0.39
	$2(0.4s)$	45.39	47.15	51.62 ± 0.18	$\mathbf{51.79} \pm 0.16$	48.32 ± 0.43
	$5(1.0s)$	38.93	38.89	41.70 ± 0.14	$\mathbf{44.91} \pm 0.11$	41.82 ± 0.53
	$10(2.0s)$	33.59	31.33	$\mathbf{38.41} \pm 0.06$	38.11 ± 0.09	35.36 ± 0.45
S-channel	$1(0.2s)$	49.19	n/a	$\underline{57.02} \pm 0.17$	$\mathbf{\underline{57.12}} \pm 0.36$	$\underline{55.33} \pm 0.16$
	$2(0.4s)$	45.17	n/a	53.06 ± 0.23	$\mathbf{53.15} \pm 0.29$	51.29 ± 0.09
	$5(1.0s)$	38.52	n/a	45.32 ± 0.37	$\mathbf{45.40} \pm 0.07$	43.40 ± 0.34
	$10(2.0s)$	32.98	n/a	$\mathbf{38.16} \pm 0.54$	37.96 ± 0.10	36.13 ± 0.38

5 Experimental Results and Discussion

5.1 Performance on BDD100K

Comparison of the proposed architectures: As mentioned above, there are no semantic segmentation ground truth labels available for $\mathcal{D}^{\text{BDD-MOTS}}$. Hence, we generate our own pseudo ground truth annotations $\overline{\mathbf{m}}_t$ using `PSANet`. The prediction performance for the $\mathcal{D}^{BDD-MOTS}_{\text{val}}$ with the pseudo ground truths can be seen in Table 1 in terms of $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$. We compare the three proposed predictor architectures (`PRED`, `PRED+R` and `PRED+RS`) along with two baseline forecasting techniques based on copy-paste (`CP`) and optical flow (`OF`). Copy-paste, (`CP`) is simply using the last input as prediction, i.e., $\hat{\mathbf{m}}_{t+1} = \overline{\mathbf{m}}_t$. For optical flow (`OF`), we use Lucas et al.'s [14] dense optical flow algorithm to estimate the optical flow motion matrix $\mathbf{f}$ for each pixel which is warped to predict next frame, i.e., $\hat{\mathbf{m}}_{t+1} = \mathbf{warp}(\overline{\mathbf{m}}_t, \mathbf{f})$. The dense optical flow for S-channel inputs does not make sense as the values are simply normalized probabilistic outputs and not pixels in conventional sense. Hence, the values for dense optical flow with S-channel masks were not calculated. It can be observed that `PRED+R` performs consecutively better for all time horizons in both types of input representations achieving a mean $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ performance of 57.12%, 53.15% and 45.40% for $t+1$, $t+2$ and $t+5$ future time steps respectively with S-channel input. This can be explained by the fact that this model extracts the semantic information from the input semantic segmentation masks at different spatial resolutions and passes this information to the convolutional LSTM layer ahead which performs a time series probabilistic prediction. This information gets saved in the respective hidden state representation that can be used when the next frame's embeddings come into picture in the further time steps. This process is repeated until the embeddings are downsampled by 16 times. There are corresponding de-convolution layers which upsample each encoded latent feature by a factor of 2 and thus, rendering an output whose spatial resolution is exactly the same as its input.

Table 2. $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ performance of PRED+R on $\mathcal{D}^{\text{CS}-\text{vid}}_{\text{val}}$ showing different input types as well as time horizons into the future. The most optimal input type with the best performance can be seen in **bold font**. The experiments have been averaged over 3 seeds.

Input Type	Time Horizon Δt	Method PRED+R
1-channel	$1(0.06s)$	83.37 ± 0.05
	$2(0.12s)$	80.43 ± 0.08
	$5(0.3s)$	72.37 ± 0.22
	$10(0.6s)$	62.44 ± 0.51
S-channel	$1(0.06s)$	$\mathbf{84.95} \pm 0.03$
	$2(0.12s)$	81.88 ± 0.07
	$5(0.3s)$	73.52 ± 0.12
	$10(0.6s)$	63.65 ± 0.09

Also, it is worth mentioning that there are convolutional LSTM layers in between two consecutive de-convolution layers to extract the temporal information even while de-convolution. There are skip connections with corresponding convolutional LSTM layers of equal spatial resolution to facilitate better flow of gradients between the layer representations. These factors further enhance the performance of the predictor architecture compared to PRED+RS where the convolutional LSTM layers are placed in between corresponding convolution and de-convolution layers. Also, the PRED architecture performs quite similar to the best architecture PRED+R with slight deprecation because of the absence of residual gradient flow during backpropagation.

It can be seen that the model performs well on the $\mathcal{D}^{\text{BDD}-\text{MOTS}}_{\text{val}}$ in Table 1 which is quite a challenging dataset given the unnormalized raw images, absence of proper lighting conditions, varying weather and diverse scenarios. In Table 1, we can see the comparison between standard baselines and our architecture on $\mathcal{D}^{\text{BDD}-\text{MOTS}}_{\text{val}}$. We can observe that PRED+R outperforms copy-paste (CP) and optical flow (OF) by 5.88% and 3.45% absolute $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ respectively for 1-channel input and 7.93% absolute $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ for S-channel input with copy-paste (CP) only.

Input modalities: In Table 1, we can see that the S-channel input representation achieves an increase of 1.92%, 1.36%, 0.49% $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ when compared to 1-channel inputs for PRED and PRED+R with future time horizons for $\Delta t = 1, 2, 5$, respectively. For PRED+RS, the S-channel input outperforms the 1-channel counterpart for $\Delta t = 1, 2, 5$, respectively, i.e., on an average of 3.67%, 2.97%, 1.58% $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$. Hence, it can be inferred that S-channel input representation performs better than the 1-channel input representation for all three predictor definitions. This can be attributed to the fact that the softmax activations, $\overline{\mathbf{y}}_t$ (S-channel input), indeed capture better semantic sense of the scene along with the boundary definitions and probabilistic pixel motion dependencies as these are the normalized probabilistic outputs of the semantic segmentation network. The softmax activations not only contain the most likely class per pixel probability but also

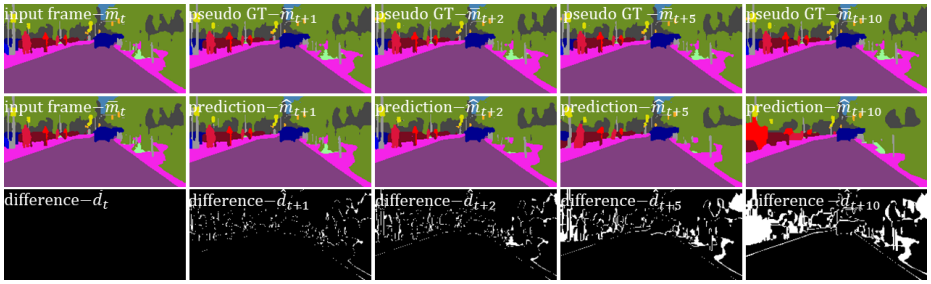


Fig. 4. Output predictions for a sequence of the Cityscapes validation split, $\mathcal{D}_{val}^{CS-vid}$. The top row depicts the pseudo ground truth $\overline{m}_t$, $\overline{m}_{t+1}$, $\overline{m}_{t+2}$, $\overline{m}_{t+5}$, $\overline{m}_{t+10}$ generated by `PSANet`. In the middle row, we show the input semantic segmentation $\overline{m}_t$ along with the predictions $\hat{m}_{t+1}$, $\hat{m}_{t+2}$, $\hat{m}_{t+5}$, $\hat{m}_{t+10}$ from the prediction network. The bottom row portrays the absolute difference $\hat{d}_{t+1}$, $\hat{d}_{t+2}$, $\hat{d}_{t+5}$, $\hat{d}_{t+10}$, between the ground truth and prediction frames. We also show additional qualitative results in the supplementary material.

the less likely and false semantic-pixel information. This in turn proves to be a better alternative compared to the raw pixel class annotations (1-channel input) where the less likely pixel connotations are completely shut off in a discrete and condensed output representation.

5.2 Performance on Cityscapes

In Table 2, we report our proposed method's performance on $\mathcal{D}_{\text{val}}^{\text{CS}-\text{vid}}$. We report performance only on `PRED+R` to save time and computation cycles, as it was clear from our experiments with the $\mathcal{D}^{\text{BDD}-\text{MOTS}}$ dataset that `PRED+R` easily outperforms other model definitions. We achieve a mean $\text{mIoU}_{\Delta\text{t}}^{\text{seg}}$ performance of 84.95% and 83.37% respectively for S-channel and 1-channel input types. Here, also the S-channel input representation proves to be a better choice than the 1-channel input type for all time horizons. In Figure 4, we also show the qualitative results of our prediction method on a sequence of Cityscapes validation split $\mathcal{D}_{\text{val}}^{\text{CS}-\text{vid}}$ with pseudo ground truths generated by `PSANet`.

5.3 Comparison with the state of the art

Most prior work report their performance only on the benchmark Cityscapes dataset $\mathcal{D}_{\text{val}}^{\text{CS}-\text{vid}}$. Hence, to make our work comparable with previous approaches, we adapt their training styles like input resolution, input frame sequence length and re-perform the experiments to calculate performances. We input 4 frame long sequences $\overline{\mathbf{m}}_{t-3}^{t} = \{\overline{\mathbf{m}}_{t-3}, \overline{\mathbf{m}}_{t-2}, \overline{\mathbf{m}}_{t-1}, \overline{\mathbf{m}}_{t}\}$ and predict one time step into the future. The input masks have a spatial resolution of 256×512. This prediction is compared with the 20^{th} human annotated ground truth frame from $\mathcal{D}_{\text{val}}^{\text{CS}-\text{vid}}$ and $\text{mIoU}_{\Delta\text{t}}^{\text{gt}}$ is calculated.

Table 3. $\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$ and $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ performance of `PRED+R` and other state of the art works in the field of prediction of semantic segmentation masks on $\mathcal{D}^{\text{CS}-\text{vid}}_{\text{val}}$. Note that here, the one time step and three time step ahead ($\Delta t = 1, 3$) prediction is being compared and our performances are highlighted in **bold**. *Taken from Jin et al. [8], **taken from Nabavi et al. [18], − are not reported.

Model	$\Delta t = 1$		$\Delta t = 3$	
	$\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$	$\text{mIoU}^{seg}_{\Delta t}$	$\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$	$\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$
S2S, Luc et al. [13]	62.60*	-	59.40*	-
Pred. Scene Parsing, Jin et al. [8]	66.10*	-	-	-
Future Sem. Seg., Nabavi et al. [18]	-	70.24**	-	58.90**
1-channel input (ours)	**67.55**	**83.95**	**63.60**	**78.28**

We can see in Table 3 for $\Delta t = 1$, our approach consistently outperforms Luc et al. [13] by 4.95% absolute and Jin et al. [8] by 1.45% absolute in terms of $\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$. Also, our model beats Nabavi et al. [18] by 13.71% absolute in terms of $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ when pseudo ground truth labels are taken into account. Similarly, for $\Delta t = 3$, our model beats Luc et al. [13] by 4.20% absolute in terms of $\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$ and outperforms Nabavi et al. [18] by 19.38% absolute in terms of $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$.

6 Conclusion

We present a time series network using LSTM units in the convolution domain for predicting semantically segmented scenarios in the future. This would help the ego vehicle to have an excellent understanding of its maneuverability decision space in good time. Our method of using convolutional LSTMs in between feature extraction layers with residual connection proves to be a better approach for predicting dynamic and static object categories on BDD100K [26] and Cityscapes [3] datasets with the freedom to use arbitrary input sequence length and output prediction time horizons. Also, we demonstrate the usefulness of employing the S-channel input representation over 1-channel input representation for improvement in semantic segmentation forecasting. We show results proving that our prediction method outperforms the current state of the art on the Cityscapes [3] dataset by 1.45% and 4.2% absolute $\text{mIoU}^{\text{gt}}_{\Delta\text{t}}$ with time horizon predictions up to $0.06s$ and $0.18s$, respectively and also outperforming the current state of the art on Cityscapes [3] by 13.71% and 19.38% absolute $\text{mIoU}^{\text{seg}}_{\Delta\text{t}}$ with time horizon predictions up to $0.06s$ and $0.18s$, and additionally we are the first to report our results on BDD100K dataset [26].

Disclaimer: The results, opinions and conclusions expressed in this publication are not necessarily those of Volkswagen Aktiengesellschaft.

References

1. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple Online and Realtime Tracking. In: Proc. of ICIP. pp. 3464–3468. Melbourne, VIC, Australia (Sept 2016)
2. Breitenstein, J., Termöhlen, J.A., Lipinski, D., Fingscheidt, T.: Systematization of Corner Cases for Visual Perception in Automated Driving. In: Proc. of IV. pp. 986–993. Las Vegas, NV, USA (Oct 2020)
3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Proc. of CVPR. pp. 3213–3223. Las Vegas, NV, USA (Jun 2016)
4. Duwek, H.C., Shalumov, A., Tsur, E.E.: Image Reconstruction from Neuromorphic Event Cameras using Laplacian-Prediction and Poisson Integration with Spiking and Artificial Neural Networks. In: Proc. of CVPR - Workshops. pp. 1333–1341. Virtual (Jun 2021)
5. Fingscheidt, T., Gottschalk, H., Houben, S. (eds.): Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety. Springer International Publishing, Cham (2022)
6. Guen, V.L., Thome, N.: Disentangling Physical Dynamics From Unknown Factors for Unsupervised Video Prediction. In: Proc. of ICCV. pp. 11471–11481. Los Alamitos, CA, USA (Jun 2020)
7. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (Nov 1997)
8. Jin, X., Xiao, H., Shen, X., Yang, J., Lin, Z., Chen, Y., Jie, Z., Feng, J., Yan, S.: Predicting Scene Parsing and Motion Dynamics in the Future. In: Proc. of NeurIPS. Long Beach, CA, USA (Dec 2017)
9. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering **82**(Series D), 35–45 (1960)
10. Kwon, Y.H., Park, M.G.: Predicting Future Frames Using Retrospective Cycle GAN. In: Proc. of CVPR. pp. 1811–1820. Long Beach, CA, USA (Jun 2019)
11. Liu, Z., Yeh, R., Tang, X., Liu, Y., Agarwala, A.: Video Frame Synthesis using Deep Voxel Flow. In: Proc. of ICCV. pp. 4463–4471. Venice, Italy (Oct 2017)
12. Lotter, W., Kreiman, G., Cox, D.D.: Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. arXiv (Aug 2016)
13. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting Deeper Into the Future of Semantic Segmentation. In: Proc. of ICCV. pp. 648–657. Venice, Italy (Oct 2017)
14. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proc. of IJCAI. p. 674–679. Vancouver, BC, Canada (Aug 1981)
15. Maas, A., Hannun, A., Ng, A.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: Proc. of ICML. Atlanta, Georgia (2013)
16. Mahjourian, R., Wicke, M., Angelova, A.: Geometry-based next frame prediction from monocular video. Proc. of IV pp. 1700–1707 (Jun 2017)
17. Mathieu, M., Couprie, C., LeCun, Y.: Deep Multi Scale Video Prediction Beyond Mean Square Error. In: Proc. of ICLR. pp. 1–14. San Juan, Puerto Rico (May 2016)
18. Nabavi, S.S., Rochan, M., Wang, Y.: Future Semantic Segmentation With Convolutional LSTM. In: Proc. of BMVC. pp. 1–12. Newcastle, UK (Sep 2018)

19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proc. of CVPR. pp. 779–788. Las Vegas, NV, USA (Jun 2016)
20. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: Proc. of CVPR. Honolulu, HI, USA (Jul 2017)
21. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Proc. of MICCAI. pp. 234–241. Munich, Germany (Oct 2015)
22. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In: Proc. of NIPS. pp. 802–810. Montreal, QC, Canada (Dec 2015)
23. Walker, J., Razavi, A., van den Oord, A.: Predicting video with VQVAE. CoRR **abs/2103.01950** (2021), `https://arxiv.org/abs/2103.01950`
24. Wang, Z., Zheng, L., Liu, Y., Wang, S.: Towards Real-Time Multi-Object Tracking. Proc. of ECCV pp. 107–122 (Aug 2020)
25. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In: Proc. of NeurIPS. pp. 12077–12090. Virtual Conference (Dec 2021)
26. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: Proc. of CVPR. pp. 1–14. Seattle, WA, USA (Jun 2020)
27. Zhao, H., Zhang, S., Wu, G., Moura, J.M.F., Costeira, J.P., Gordon, G.J.: Adversarial Multiple Source Domain Adaptation. In: Proc. of NeurIPS. pp. 8568–8579. Montréal, QC, Canada (Dec 2018)
28. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid Scene Parsing Network. In: Proc. of CVPR. pp. 2881–2890. Honulu, HI, USA (Jul 2017)
29. Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C.C., Lin, D., Jia, J.: PSANet: Point-wise Spatial Attention Network for Scene Parsing. In: Proc. of ECCV. pp. 267–283. Munich, Germany (Sep 2018)