Video Segmentation Prediction using UNet and SimVP

Cindy Luo
NYU CDS
kl3108@nyu.edu

Miles Pophal
NYU Courant
map10046@nyu.edu

Bella Zhang
NYU CDS
bz2428@nyu.edu

Abstract

We give an approach to modeling video prediction with semantic segmentation using a UNet and SimVP structure. We examine how these models perform on some sample data and provide future directions in light of our current model's strengths and limitations.

1 Introduction

Video segmentation prediction is a critical task in computer vision, with applications in autonomous driving, video surveillance, and human-computer interaction. In recent years, deep learning approaches have demonstrated remarkable success in solving complex computer vision tasks. Among them, the UNet architecture has received considerable attention for its effectiveness in semantic segmentation tasks. In addition, the SimVP framework has provided a powerful means for generating synthetic video data. This paper addresses the challenging problem of predicting future segmentation masks in a given sequence of images by combining the strengths of UNet and SimVP.

Problem Statement The problem at hand involves predicting the segmentation mask for a future frame in a sequence of images. Specifically, the input comprises 11 consecutive frames, and the task is to generate the segmentation mask for the 22nd frame. Each frame in the sequence is composed of an individual image accompanied by its corresponding ground truth segmentation mask.

The dataset used for training and validation consists of synthetic videos featuring simple 3D shapes that interact with each other based on fundamental physics principles. The dataset comprises 1,000 labeled training videos, each containing 22 frames, as well as 1,000 labeled validation videos with the same frame structure.

2 Literature

Several authors have explored video prediction with semantic segmentation. One example is Luc et al. [2017], they performed video prediction using semantic segmentation and compared different structures. They examined models which map image sequences to the next image, segmentation sequences to the next segmentation, and combined sequences to the next combined frame (both mask and image, or individual components) along with different strategy for predicting more than a single frame into the future. They used a pretrained Dilation10 model to generate masks and multi-scale alternating convolutions and ReLUs for the actual prediction. They were able to do short-range predictions, but the model struggled with long range predictions (see section 4.4 of Luc et al. [2017]) because everything blurred as an artifact of the deterministic model predicting multi-modality. They also demonstrated that the Segmentation-to-Segmentation (S2S) model outperformed other models using either image-only or image-with-segmentation input in the future segmentation prediction task, leading us to take a similar approach in our methodology.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3 Methodology

Following the literature review, we divided the problem into two distinct tasks, the segmentation task and the prediction task. The segmentation task takes a single image and maps it to a single segmentation without any temporal shift. The prediction task consists of taking a sequence of 11 segmentation masks and predicting the next 11 segmentations. We used (v2) random horizontal and vertical flips, crops, and rotations for data augmentation.

3.1 Segmentation Task

To motivate our choice for this task we examine the assumptions of the problem which we can abbreviate to *locality* and *sparsity*. Here, locality is present because we are performing per-pixel classification and not global image classification. Sparsity is here in the sense that each solid is uniquely determined by its shape and color, and hence, we expect many 0 terms in a class probability tensor. These two assumptions naturally lead us to consider convolutions as a model for segmentation. We opted to use a "fully-convolutional network" (from Ronneberger et al. [2015]), which we adopt the colloquial term UNet. This model was a good fit because we needed to train it ourselves and the literature (see Section 2) has shown they perform well for learning segmentations. Figure 1 from Ronneberger et al. [2015] shows the structure of the UNet.

The downsampling and upsampling structure allows it to extract the features from an image and then convert those into the segmentation, and patching allows it to model the global image well. We use the repo from Alexandre [2023] for a PyTorch implementation of the UNet and call it with $n_{channels} = 3, n_{classes} = 49$ and with bilinear interpolation disabled.

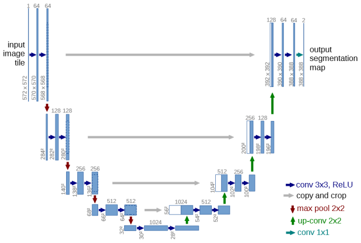


Figure 1: Structure of the UNet

3.2 Prediction Task

For our second task, prediction, we adopted the SimVP architecture introduced by Gao et al. in 2022 Gao et al. [2022]. SimVP was designed for video frame prediction and achieved state-of-the-art performance in benchmarks like Moving-MNIST. It used a CNN-CNN-CNN architecture. In specific, the architecture consists of three main modules: the encoder, the translator, and the decoder. The encoder is used to extract spatial features, and the translator, consisting of inception modules, can learn temporal evolution. The decoder will reconstruct the ground truth frames. This architecture is simple and should be scalable for small datasets like the one presented in the paper.

Model Architecture The SimVP architecture takes in several parameters that define the model scale. N_S stands for the number of ConvNormReLU blocks in the encoder (also the number of unConvNormReLU blocks in the decoder); N_T represents the number of inception modules in the translator. hid_S and hid_T are the number of hidden channels in the inception modules. SimVP's output has the same dimension as the input, which adapted well for our prediction task since it takes 11 frames and predicts the future 11 frames. Our choice of input/output dimensions depended on our training approach and the loss we adopted, elaborated in the model training section below.

Dataset We used the masks provided in the training set to train the model and evaluated the model performance using the validation set.

Model Training & Loss Definition Our experimentation involved two primary approaches centered on varying the loss function during training.

• **MSE Loss** We employed the MSE loss, standard for frame prediction, with both input and output dimensions of 11, 1, 160, 240. We treated each frame's segmentation as a 1-channel image input, and we trained the model to predict the 1-channel segmentation mask. (We also experimented with frame prediction but the outcome was not ideal). We experimented with different model configurations, including the number of inception modules and encoder/decoder, and learning rates. Since the values in the output are often floats rather than integers, we directly took the floor of the float values and used that as the final prediction. We saved the model with the lowest MSE loss on validation set as our best model.

- **Cross Entropy Loss** We also implemented cross-entropy loss, transforming labels into one-hot vectors, increasing channels from 1 to 49 (as compare to the MSE Loss). Therefore, the input dimension becomes 11, 49, 160, 240. Additionally, we applied weighted loss, emphasizing the last frame in loss calculation by calculating its cross entropy loss separately from other frames, scaling it by a positive integer (2-4), and then normalizing the sum of this loss and the loss from other frames. Back propagation is performed based on the normalized loss. We first saved the model with the lowest loss on validation set as our best model; as the validation loss reached its bottleneck, we saved the model with the highest IoU on last frame's segmentation prediction on the validation set.

4 Results

4.1 Segmentation

We used pixel-wise cross entropy loss to train the UNet, and used the Jaccard index (IoU accuracy) as a metric on the validation set to select the best performing model. Using a $70 - 30$ split of the 1000 videos (22000 total images) and optimizing using Adam we were able to achieve 89.90% IoU score on the validation set after $\sim$ 30 epochs. We used this model to generate the segmentation masks for the prediction model.

4.2 Prediction

4.2.1 Models Trained with MSE Loss

The training results based on different model configurations are included in Table 1. Figure 2 below includes the training trajectories of the 5 different training configurations we picked. Some trajectories represents multiple rounds of training for the same model. Notice that the training always reached a bottleneck, with the validation loss hardly going below 20.

N_S	N_T	hid_S	hid_T	groups	Train Loss (MSE)	Val Loss (MSE)
8	16	64	512	8	9.809	28.112
8	16	64	256	8	8.695	28.482
4	8	64	256	4	13.657	20.862
4	8	16	64	4	20.149	22.544
2	4	4	32	1	16.804	21.188

Table 1: Experimental training results of models trained using MSE Loss.

We included the training details of the best model among the configurations we tried. The model's configuration is 4_8_64_256_4 (N_S, N_T, hid_S, hid_t, groups; listed in the third row in Table 1). The IoU between the last-frame predictions and the ground truth segmentaion on the whole validation set was 0.0287. Figure 4 shows the model prediction versus the ground truth segmentation from the validation set. We can tell that our model could somewhat predict the still objects, but all the moving objects were blurry or missing from the prediction. Additionally, we can tell that the color of an object is not uniform, suggesting that our transformation from the model output to final prediction needs further refinements.

4.2.2 Models Trained with Cross-Entropy Loss

We directly use the same model configuration from the best model we selected from our previous training using the MSE Loss and changed the group size from 4 to 7. The model's configuration is 4_8_64_256_7 (N_S, N_T, hid_S, hid_t, groups). The trajectories from multiple rounds of training in Figure 3 suggested that this training also reached a bottleneck. This type of training achieved around 0.269 in validation. Looking at Figure 4, this model is better at predicting stationary objects, but

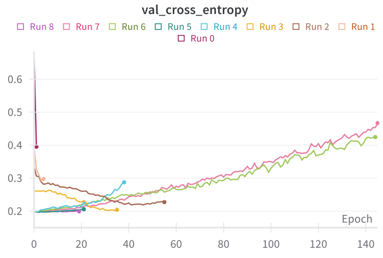
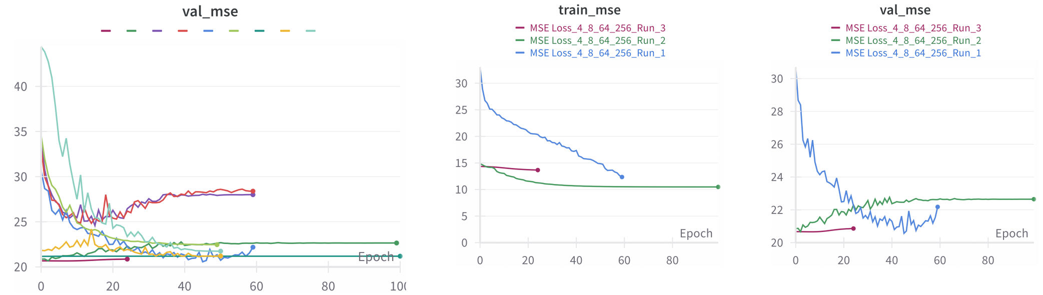


Figure 3: Validation Loss During Model Training Using the Cross-Entropy Loss.



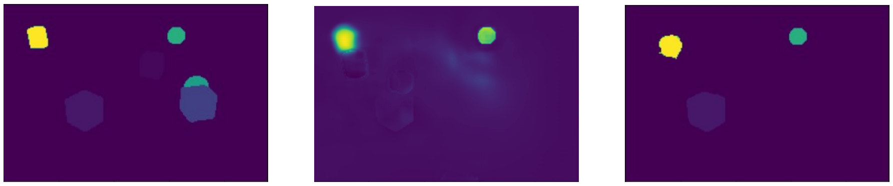
(a) Overview of Validation Loss (MSE Loss) During Different Training Trajectories.

(b) Training Loss During 3 Rounds of Training of the Same Model.

(c) Validation Loss During 3 Rounds of Training of the Same Model.

Figure 2: Model Training Trajectories Using MSE Loss.

moving objects are still missing. Due to higher Jaccard similarity, we picked a model from this training method for our final submission.



(a) Target Mask.

(b) Prediction of model using MSE

(c) Prediction of model using CrossEntropy

Figure 4: Target mask and prediction mask for validation video with index 1050.

5 Takeaways & Conclusion

Our training results highlight a critical issue: a performance bottleneck characterized by reduced training loss but stagnant or rising validation loss. This suggests we didn't fully leverage the available data, as incorporating pseudo-labels from an unlabelled dataset could expand our training set. Future strategies may include using pre-trained models in a sequential net and fine-tuning, or adjusting the loss function to focus on the last frame. Expanding the training set may also help the model to better predict motion trajectories, that our current model struggled with. Additionally, smaller-scale SimVP may be more effective for our dataset size, as suggested by high-performing teams during their presentation.

In all, the combination of Unet and SimVP under our current training paradigm did not yield a very high performance on the segmentation prediction task, indicating that there is still room for improvements. Major issue could be related to not taking advantage of the trained models and the unlabelled dataset to expand the training set and using a large model scale given that we had limited data. Nevertheless, we have experimented with different approaches for model training and found a more appropriate loss form that could lead to 0.27 IoU in the prediction task. Improvements in performance could be achieved by expanding the dataset using our segementation model.

References

Milesi Alexandre. Pytorch-unet. https://github.com/milesial/Pytorch-UNet/tree/master, 2023.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. Simvp: Simpler yet better video prediction, 2022.

Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.