# SimVP-UNet: Framework For Video Frame Prediction and Semantic Segmentation using SimVP and U-Net

**Abhipsha Das*** ad6489@nyu.edu

**Simran Makariye*** sdm8499@nyu.edu

**Srushti Pawar*** sxp8182@nyu.edu

## Abstract

In this paper, we introduce an innovative approach to video prediction and semantic segmentation by combining SimVP[1] and U-Net[2]. SimVP is a video prediction model that learns spatio-temporal representations from videos using convolutional neural networks (CNNs). The U-Net model is a widely used model for image segmentation tasks that has been shown to be effective in various applications. In this work, we propose a framework that integrates these two models to predict future frames in a video sequence and segment the objects of interest in the last frame. Specifically, the SimVP model predicts the future frames, while the U-Net model generates a mask for each object in the predicted last frame. The resulting masks are then utilized to segment the objects of interest and differentiate them from the background. To evaluate the effectiveness of our approach, we conducted experiments on an unseen dataset and measured performance using the Jaccard index, achieving a value of 0.251 with minimal training and fine-tuning. We believe that our proposed approach has potential in a wide range of applications, including anomaly detection, object tracking and autonomous driving.

## 1 Introduction

Video frame prediction and segmentation of moving objects are challenging computer vision tasks with practical applications in robotics, autonomous driving, and surveillance. These tasks are complicated by the interactions between moving objects and their environment, as well as the inherent noise and uncertainty in the data.

In recent years, deep learning techniques have shown promising results for video frame prediction and image segmentation tasks. We explored RNN-based architectures such as ConvLSTM + GANs and advanced architectures like masked autoencoders. However, we found that the best performance was achieved using the SimVP model.We also experimented with U-Net for image segmentation task. The U-Net architecture consists of an encoder, which downsamples the input image, and a decoder, which upsamples the encoded features to generate the segmentation mask.

In this paper, we used a synthetic 3D training dataset consisting of video clips, each containing 22 frames. The videos depict 3D moving objects, with each object having a unique combination of three attributes - shape (cube, sphere, or cylinder), material (metal or rubber), and color (gray, red, blue, green, brown, cyan, purple, or yellow). The dataset is designed such that no two objects have identical attributes and offers a challenging set of visual stimuli for training and evaluating object segmentation models.

*These authors contributed equally to this work.

## 2 Relevant Background

Video frame prediction is a challenging task for a machine as it deals with motion and object occlusion, and the possibility of multiple outcomes of a video. Past deep learning approaches have been concentrated on autoregressive RNN based implementations as well as CNN architectures, GANs, and more recently, transformer and autoencoder based architectures to model latent video dynamics more efficiently. Video datasets such as MovingMNIST [3] have been widely used as benchmark datasets for research in this area. RNN based approaches such as Convolutional Long-short Term Memory[4] have been popular due to their ability to model forecasting, but alternatives that can capture long term dependencies when predicting multiple steps into the future are required. Generative adversarial networks have been used in conjunction with RNN architectures to differentiate between model predicted and true images. Vision transformer(ViT) based models like the Video Swin Transformer[5] have shiftable local attention schema resulting in higher speed-accuracy trade-off; however, most of these are designed for video classification and ViT based implementations particularly aimed at solving frame-prediction are still limited. Auto-encoder based models such as VideoMAE[6] which employs the masked encoding objective by masking random cubes and reconstructing the missing ones have shown great results as a pretraining step for video inputs, trained with the self-supervised learning objective and contrastive loss computation; and seem promising to downstream tasks such as frame prediction. Past CNN architectures have been used to predict per-pixel motion and optical flow prediction[7] and more recently architectures like SimVP have performed extremely well with reduced model complexity and simple training objectives.

Semantic segmentation on objects is another important computer vision task that has seen multiple successful CNN-based architectures over the years, most notably the Mask R-CNN architecture[8] and the U-Net model which is an autoencoder model in which an image is converted into a vector and then the same mapping is used to reconstruct an image. This reduces the distortion by preserving the original structure of the image.

## 3 Methods

For our implementation, we explored a number of architectures to find the best performing ones for the joint frame-prediction and mask prediction task.

### 3.1 Models

#### 3.1.1 Future Frame Prediction

**Overview** In the first part, we need to predict the future frames of a video conditioned on past frames, to accurately depict the motion and tracking of the moving objects in the video. For this, we explored ConvLSTM + GAN initially to predict future frames by taking the past 11 frames of a video and then autoregressively predicting 11 future frames, and the final frame was then passed to a GAN module where the discriminator learnt to distinguish between the ground truth final predicted frame and the model predicted images. The complexity of the model and slowness of training made us adopt a fully-CNN based architecture called SimVP. The prediction pipeline was much the same as the one we employed previously, where for every 11 past frames, we predict the next 11 frames.

**Model Architecture** SimVP consists of an encoder, a translator and a decoder module. The encoder takes in 11 frames of dimension $160 \times 240 \times 3$ and extracts spatial features from the input frames, the translator learns temporal the evolution of the frames, and the decoder integrates spatio-temporal information to predict future frames. The encoder module has 4 blocks of Conv2D, LayerNorm and LeakyReLU layers stacked together to give hidden features. The hidden feature is:

$$z_i = \sigma(LayerNorm(Conv2D(z_{i-1})))\forall 1 \leq i \leq 4$$

This is then passed to a translator module that is made up of 8 Inception blocks, each with a different kernel size ([3, 5, 7, 11]). The Inception module consists of a bottleneck Conv2d with $1\times1$ kernel followed by parallel GroupConv2d operators. Using a CNN-based translator module was a choice we made as on running experiments, CNN required the least finetuning. The hidden feature is:

$$z_j = Inception(z_{j-1})\forall 4 \leq j \leq 4 + 8$$

The decoder module is used for the unconvolution operation, consisting of 4 blocks of ConvTranspose2D, GroupNorm and LeakyReLU layers, where the hidden feature is:

$$z_k = \sigma(GroupNorm(ConvTranspose2D(z_{k-1})\forall 4+8 \leq k \leq 2*4+8$$

to give the output of 11 future predicted frames, each of dimension 160×240×3.

### 3.1.2 Semantic Segmentation

**Overview** In the second phase of the problem, our objective is to generate individual masks for every object present in the image, which we perform using semantic segmentation of the images that assigns a categorical label to each individual pixel in the image. To achieve this, we used the U-Net model. U-Net is a fully convolutional network that employs a symmetric encoder-decoder structure, which enables it to effectively capture both high-level and low-level features of an image.

**Model Architecture** We use U-Net for image segmentation that takes as input an image with size 160×240, and produces a single-channel masked image of the same size. The U-Net architecture consists of encoding blocks and decoding blocks. In our implementation, the encoding blocks consist of 4 blocks, each of two 3×3 Conv2D layers with N filters followed by a ReLU, BatchNorm and MaxPool to downsample the image, with $N \in [64, 128, 256, 512]$. The number of filters in the convolution layers increases with depth to capture higher-level features. After the last encoding block, a bottleneck layer with 1024 filters is added, followed by 4 decoding blocks, each consisting of 2 3×3 ConvTranspose2D layers, ReLU and BatchNorm with decreasing numbers of filters (same as in encoding layers in opposite direction). Finally, a convolution layer with a single filter is used to produce the output segmentation map of dimension 160×240. Skip connections are added between the encoding and decoding blocks to preserve the spatial information lost during downsampling.

## 3.2 Experiments

### 3.2.1 Dataset

We train a frame prediction model on an unlabeled dataset of 13,000 videos of objects, consisting of 22 frames of size 160×240×3, and use a hidden dataset of the first 11 frames to predict the $22^{nd}$ frame, which is then passed to a semantic segmentation model trained on 1,000 videos of 22 frames each and the masks for each frame to predict the mask for the final frame. A segmentation output produces masks that classifies 49 different combinations of object characteristics.

### 3.2.2 Training

Vanilla MSE loss was used to train the frame prediction model for 25 epochs on the unlabeled dataset. We performed hyperparamter tuning for this model on different optimizers [Adam, RMSProp] and learning rates [1e-3, 1e-2] and finally trained it at a learning rate of 1e-3, using Adam optimizer and OneCycle LR scheduler using DataParallel for speedup on 2 GPUs. For the segmentation model, cross entropy loss was utilized to calculate the loss per epoch, with a learning rate of 1e-4. The model was trained for 10 and 20 epochs and its performance was assessed using the Jaccard Index on 1,000 randomly selected images from the validation set. Our models were trained on NVIDIA Tesla V100 GPUs and combined training and finetuning took approximately 70 GPU hours.

# 4 Results

We evaluated U-Net model's performance on 1,000 images from the validation dataset. The Jaccard Index values obtained from this experiment are presented in Table 1. The U-Net model trained on 20 epochs performed the best on these unseen images, achieving a maximum Jaccard Index score of 0.9693. This suggests that our segmentation model has the ability to predict well. The Jaccard Index values were obtained with our combined pipleine of SimVP + U-Net predictions on the entire validation dataset of 1,000 video clips and are presented in Table 2. The best performing model achieved a Jaccard Index of 0.2452 and was obtained from the combination of 25 epochs of SimVP training with a learning rate of 0.001. Figure4, 4 and 4 shows how features are learnt over the epochs. The final jaccard index achieved on the hidden dataset is **0.251** based on the final results.

Table 1: Performance of the U-Net Model on Validation dataset

| U-Net (epochs) | Jaccard Index |
|---|---|
| 10 epochs | 0.9680 |
| 20 epochs | **0.9693** |

Figure 1: After 1 epoch, model learns to distinguish objects from background

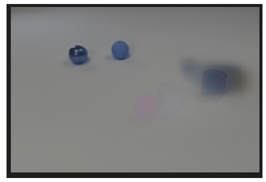

Figure 2: After 5 epochs, model predicts shapes, stationary objects across frames are predicted more easily than moving ones

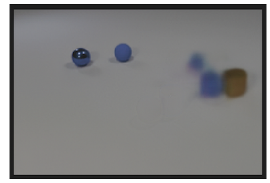

Figure 3: After 25 epochs, model learns to predict position of moving objects and identify their shape and colour

Table 2: Evaluation Results

| Models | | |
|---|---|---|
| SimVP Params (epochs and learning rate) | U-Net | Jaccard Index |
| 25 epochs and 0.01 learning rate | 20 | 0.2417 |
| **25 epochs and 0.001 learning rate** | **20** | **0.2452** |

## 5 Conclusion

Our experiments show that training the model for longer leads to better predictions, with no loss of generalization as seen in the results obtained on unseen data and the loss curves showing a converging trend. The fully-CNN architecture is lightweight and easily trained with little finetuning required, and learns well at even 25 epochs of training. We believe that the future direction is to train the frame prediction model it for a longer duration and add post-processing steps to get less noisy frames, as the current implementation lacks noise-reduction steps. For the baseline, our approach performs well with a Jaccard index of **0.245** on validation dataset and **0.251** on the hidden dataset.

## References


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

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

[3] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms, 2016.

[4] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584, 2015.

[5] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer, 2021.

[6] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022.

[7] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image, 2015.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.