Video Frame Prediction and Semantic Segmentation: SimVP - UNet

Qianqian Zhao
qz2057@nyu.edu

Xiangdong Zhang
xz3645@nyu.edu

Yiru Zhou
yz8297@nyu.edu

Abstract

In this paper, we introduce the performance of the combination of the SimVP[1] and the UNet[2] model in video prediction and semantic segmentation. SimVP is a video prediction model that learns spatio-temporal representations from videos. It is fully based on CNN and trained by the MSE loss end-to-end. The U-Net model is a widely used model for image segmentation tasks whose outputs include localization, i.e., a class label to be assigned to each pixel. In this project, we propose a framework that integrates these two models to predict future frames in a video sequence and segment the objects by generating a mask for each object in the last frame. We apply the SimVP model to predict future frames and use the U-Net model to generate a mask for each object in the predicted last frame. The resulting masks segment the objects and differentiate them from the background. For evaluation, we conducted experiments on a hidden dataset and measured the model performance using the Intersection of Union (IOU[3]) metric and achieved a final IOU of 0.272.

1 Introduction

The prediction and segmentation of video frames depicting moving objects constitute formidable challenges within the computer vision field, boasting significant implications for fields such as robotics, autonomous vehicular technology, and surveillance systems. These tasks are inherently complex due to the dynamic interactions between the objects and their environments, compounded by the prevalent noise and uncertainty within the data. Our findings indicate that the most efficacious results were obtained utilizing the SimVP model. Concurrently, we have conducted experiments with the U-Net architecture for image segmentation, which features an encoder for downsampling the input image and a decoder for upsampling the encoded attributes to create the segmentation mask. This paper employs a synthetic 3D training dataset composed of video clips, each containing 22 frames that portray 3D moving objects with distinct attributes, including shape, material, and color. This dataset is uniquely designed to ensure that no two objects share the same attributes, thereby providing a rigorous array of visual stimuli for the training and evaluation of object segmentation models.

2 Relevant Background

Previous deep learning methods primarily focused on autoregressive RNNs, CNNs, GANs, and, more recently, transformers and autoencoders for modeling video dynamics. Vision Transformer (ViT) models, such as the Video Swin Transformer, employ shiftable local attention mechanisms to enhance the speed-accuracy trade-off, although they are predominantly designed for video classification, with limited applications in frame prediction. Autoencoder-based models, notably VideoMAE, use a masked encoding objective to reconstruct randomly masked cubes, showing promising results in pre-training for video inputs through self-supervised learning and contrastive loss. Additionally, past CNN models have been effective in per-pixel motion and optical flow prediction, with newer architectures like SimVP significantly reducing model complexity and simplifying training objectives. In semantic segmentation, notable CNN-based models include Mask R-CNN and U-Net, the latter using a vectorization approach to minimize distortion and maintain the original image structure.

3 Models

3.1 SimVP Model in Frame Prediction Task

Overview SimVP consists of an encoder, a translator, and a decoder built on CNN. The encoder takes in 11 frames of dimension 160 × 240 × 3 and extracts spatial features from the input frames, the translator learns the temporal evolution of the frames, and the decoder integrates spatiotemporal information to predict future frames.

Model Architecture In our model, Ns is 4 (we are using 4 ConvNormReLU blocks) and Nt is 8 (we have 8 groups of inception modules).

The encoder stacks Ns ConvNormReLU blocks (Conv2d+LayerNorm+LeakyReLU) to extract spatial features, i.e., convoluting C channels on (H, W). The hidden feature is:

$$z_i = \sigma(\mathrm{LayerNorm}(\mathrm{Conv2d}(z_{i-1}))), \quad 1 \leq i \leq N_s,$$

where the input zi−1 and output zi shapes are (T, C, H, W) and (T, Cˆ, Hˆ, Wˆ), respectively.

The translator employs Nt Inception modules to learn temporal evolution, i.e., convoluting T × C channels on (H, W). The Inception module consists of a bottleneck Conv2d with a 1×1 kernel followed by parallel GroupConv2d operators. The hidden feature is:

$$z_j = \mathrm{Inception}(z_{j-1}), \quad N_s < j \leq N_s + N_t,$$

where the inputs zj−1 and output zj shapes are (T × C, H, W) and (Tˆ × C, H, Wˆ).

The decoder utilizes Ns unConvNormReLU blocks (ConvTranspose2d + GroupNorm + LeakyReLU) to reconstruct the ground truth frames, which convolutes C channels on (H, W). The hidden feature is:

$$z_k = \sigma(\mathrm{GroupNorm}(\mathrm{unConv2d}(z_{k-1}))), \quad N_s + N_t < k \leq 2N_s + N_t,$$

where the shapes of input zk−1 and output zk are (T, Cˆ, Hˆ, Wˆ) and (T, C, H, W), respectively. We use ConvTranspose2d to serve as the unConv2d operator.

3.2 UNet Model in Semantic Segmentation and Mask Generation Task

Overview Our objective is to generate individual masks for every object present in the image, which we assign a categorical label to each individual pixel in the image. To achieve this, 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.

Model Architecture The network architecture consists of a contracting path and an expansive path. The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions) with N filters, N $\in$ [64, 128, 256, 512], each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step, we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer, a 1x1 convolution is used to map each 64 component-feature vector to the desired number of classes that produce the output segmentation map of dimension 160×240.

4 Experiments

4.1 Dataset

We train a SimVP 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 22nd 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.

4.2 Training Methods

4.2.1 SimVP Model Training

The SimVP model was trained over 100 epochs with a learning rate of 0.0005, using Mean Squared Error (MSE) as the loss function. The performance of the model was evaluated based on the validation loss, MSE, Mean Absolute Error (MAE), and Peak signal-to-noise ratio (PSNR). This evaluation helped in optimizing the model to accurately predict the 22nd frame from the first 11 frames of each video in the dataset.

4.2.2 Semantic Segmentation Model Training

The semantic segmentation model, based on the U-Net architecture, was trained for 50 epochs with a learning rate of 0.0005. The loss function was a combination of Cross-Entropy Loss and Intersection over Union (IoU) Loss to effectively handle the multi-class segmentation task. The model's performance was primarily assessed on the validation set using the IoU metric, which measures the overlap between the predicted masks and the ground truth masks, thereby providing insight into the model's accuracy at pixel-level segmentation.

5 Results and Conclusions

We evaluated the SimVP model's performance on 1,000 images from the validation dataset. As shown in **Figure 1**, the SimVP model trained on 100 epochs with a learning rate of 0.0005 performed the best on these unseen images, achieving a Structural Similarity Index Measure (SSIM) score of 0.95, confirming that the model's predictions are very similar to the original true picture. This suggests that our frame prediction model is capable of predicting well. As shown in **Figure 2**, the U-Net model's performance on 1,000 images from the validation dataset with 30 epochs and a learning rate of 0.005 achieves a Jaccard index of 0.9522, which also illustrates our good performance on the mass prediction. The final IOU values were obtained with our combined pipeline of SimVP + U-Net predictions on the entire validation dataset of 1,000 video clips. The best-performing model achieved a **final IOU of 0.275** and was obtained from the combination of 30 epochs of combined model training with a learning rate of 0.00001.

These pictures in **Figure 1** are the last frame prediction results of the first video in the validation set of the SimVP model at 20, 40, and 60 epochs respectively. We can see that the picture is getting clearer and clearer. **Figure 2** shows the mask predicted for the last frame of the first video in the validation set of the U-Net model.

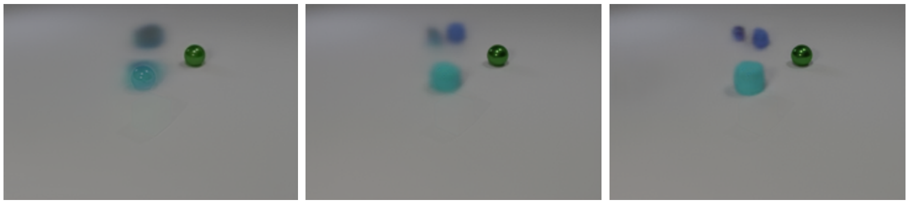

Figure 1

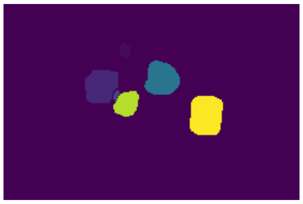

Figure 2

We believe that the future direction is to train the frame prediction model 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 0.275 on the validation dataset and 0.2723 on the hidden dataset.

References

[1] Gao, Z., Tan, C., Wu, L., & Li, S. Z. (2022). SimVP: Simpler yet Better Video Prediction. *ArXiv*. /abs/2206.05099

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv*. /abs/1505.04597

[3] Rosebrock, A. (2023, May 12). *Intersection over union (IOU) for object detection*. PyImageSearch.https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/