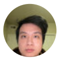
Video Frame Prediction with ViViT and U-Net

Preprint · May 2024

DOI: 10.13140/RG.2.2.36713.10087



Zhou Zhou
New York University

Video Frame Prediction with ViViT and U-Net

Zhou Zhou **Yunqing Zhu**
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012-1185
{zz3382, yz9661}@nyu.edu

Lisha Tong
Center for Data Science
New York University
60 5th Ave, New York, NY 10011
lt2421@nyu.edu

Abstract

The deep learning final competition is video frame prediction and semantic classification. Specifically, this competition needs to use the first 11 images to find the 22nd image object class. Every team cannot use any pre-trained model and outside dataset. Our team divided this question into two separate questions, an image generator question and a semantic classification question. We use JEPA encoder and ViViT model to implement the image generator. Then we choose the U-net model to implement the object classification and achieve 0.92 IOU score in the validation dataset. Finally, our team got 0.24 IOU score in the hidden dataset.

1 Introduction

The competition task is to use the first 11 frames to predict the 22nd frame segmentation mask. The provided competition dataset includes 13,000 unlabeled videos with 22 frames, 1,000 labeled training videos with 22 frames, 1,000 labeled validation videos with 22 frames and 5,000 unlabeled videos with 11 frames for hidden test. The objects in the videos have three shapes (cube, sphere, and cylinder), two materials (metal and rubber), and eight colors (gray, red, blue, green, brown, cyan, purple, and yellow). This competition includes two different questions. First, how to generate the 22nd from the first 11 frames. Second, how to implement semantic classification. The first question is a classic video frame prediction problem. Gao et al [1] implement SimVP to achieve video prediction and it is completely built upon CNN and trained by MSE loss in an end-to-end fashion. Oliu et al[2] choose folded recurrent neural networks and Shi et al [3] implement ConvLSTM model to complete the video prediction task. We tried the recurrent models, like ConvLSTM, Floded RNN and PredRNNv2 [4], whose performance is not as good as recurrent free models. In this case, we choose the JEPA (Joint-Embedding Predictive Architecture) model [5] as encode to process input frames and use ViViT model [6] to generate the 22nd image frames. Besides, we choose the U-net [7] model to implement semantic segmentation and find the generated 22nd frame-related mask.

2 Methods

2.1 Part1: JEPA Encoder

We use the JEPA (Joint-Embedding Predictive Architecture) model [5] as our encoder. The goal of the encoder is to extract the model's features into vectors through the ViT model by self-supervised learning. The model covers some patches and then uses the encoder to extract the features from both covered patches and uncovered patches separately. Then, the model uses a predictor to predict the covered patch's features from the uncovered patch's features. The loss is the MSE loss between two features. By comparing the features instead of images, the encoder can be more generalized and better understand the features of the image.

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

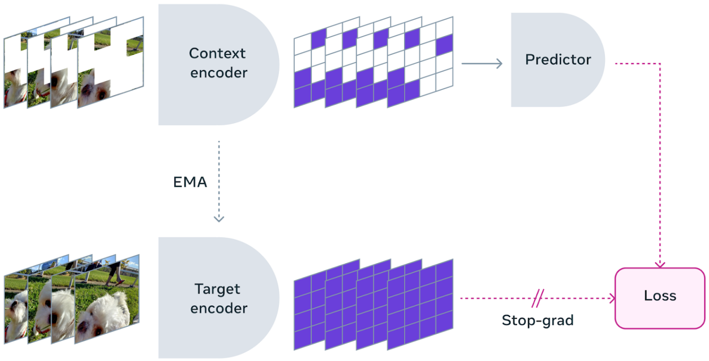


Figure 1: The architecture of the JEPA model. [5]

We use the ViT-tiny with 16*16 patch size as the encoder model. While the image size is 160*240, there are 150 patches, and we covered 112 patches, which is around 75% of all patches. The positional embedding is fixed: using 2D sinusoidal positional embedding in the encoder and 1D sinusoidal positional embedding in the predictor. The encoder has an embedding size of 192, a hidden size of 768, and 12 layers with 3 heads. The predictor has the same size as the encoder. The Hyperparameters are the same as those in the original paper: the batch size is 2048, the model is trained in 300 epochs, and the optimizer is AdamW. The learning rate is initially 10^{-4} and linearly increases to 10^{-3} in 15 epochs and then decreases to 10^{-6} with cosine function afterward. The weight decay is initially 0.04 and linearly increases to 0.4. The target network is updated using EMA with a momentum of initially 0.996 and linearly increases to 1.

2.2 Part2: Image Generator

We use the ViViT model to generate the target image [6]. We use tubelet embedding and embed all frames together so that the patch embedding kernel size is (11, 16, 16). The model is a ViT-base model with an embedding size of 768, a hidden size of 3072, and 12 layers with 12 heads. We use a transposed convolutional layer at the end of the model as the decoder. The generator's input is 11 images with timestamps 1-11, and the output is 11 images with timestamps 12-22. After the generating process, both target images and generated images will go through the JEPA encoder, and we use the MSE loss between the features from the generated images and the target images as the loss of the model.

2.3 Part3: Masker

We choose the U-net model to implement semantic classification and Figure 2. There are 49 different class objects (including background) in the video. The mask file shape is (1, 160, 240) for each image. The U-net model accepts the input image shape as (3, 160, 240) and the output shape as (49, 160, 240). Then, we can use the cross-entropy loss to implement the semantic classification. We can get a 0.95 Jaccard score on the training set and a 0.92 Jaccard score on the test set. Figure 3 indicates the sample classification result.

2.4 Part4: Masker Decoder From Transfer Learning

In this step, we utilize transfer learning to train the mask decoder in our model. This step aims to train a decoder to generate the mask from the image feature extracted by the pre-trained encoder. Because of the limited number of the training set, we need to use the unlabeled pictures to train the masker decoder. The decoder has 6 attention layers with the same architecture as the encoder. The patch size is 192, and the number of heads is 3. We use the U-net masker to generate the pseudo mask and use those masks as the target for the decoder. The encoder is frozen, and only the 6-layer decoder is trained. The final layer of the decoder is also a transposed convolutional layer that can generate masks with 49 channels. The loss is the cross entropy loss.

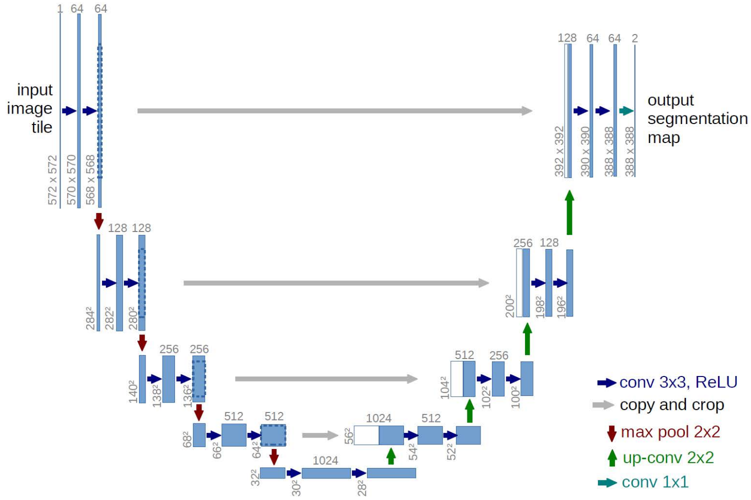


Figure 2: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.[8]

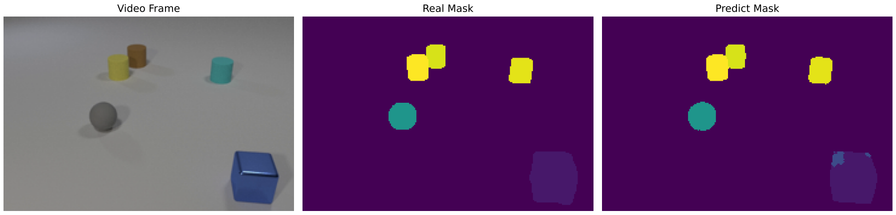


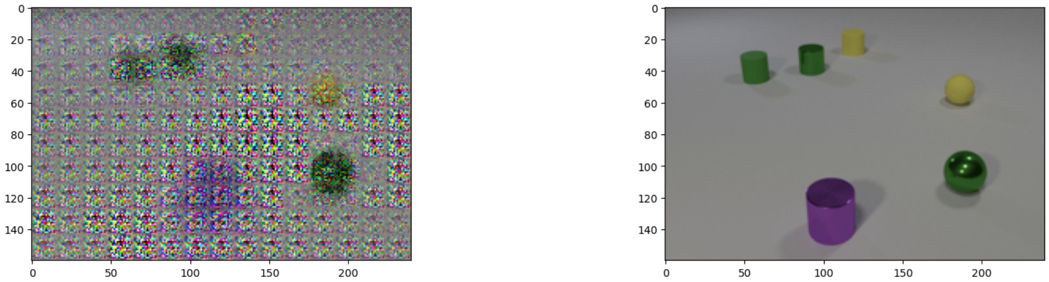
Figure 3: Semantic classification using U-net

2.5 Part5: Final Fine Tuning

Overall, the inference pipeline uses the first 11 images as input to the generator, and the output is put into the next 11 images. Then, only the 22nd frame was used as input to the JEPA encoder to extract features, and the masker decoder was used to generate the final result. In the final step, we assemble every part of the model and fine-tune only the last maker decoder using the training dataset while all other parts are frozen. The improvement of this step is minimal and lets the decoder better fit the last frame instead of all images.

3 Result

Figure 4 shows the generated 22nd frame and the target. From the image, we can observe that the features of the two images are almost the same, although they look different to humans, especially in blank patches. We can see every 16*16 patch from the generated image. From our encoders' point of view, the two images are very similar, so the difference between the two images will not affect the performance of the masker decoder. In the transfer learning step of the masker decoder, the decoder gets a result of 0.92 in the Jaccard score of a single image, while without transfer learning, the decoder can get a score of 0.8 only training with the training set. Because the result of the decoder is almost the same as the result of the U-net masker, we can conclude that the encoder correctly extracts the features from the images. The loss is the cross entropy loss. As the final result, our model can get a Jaccard score 0.24 in the validation dataset.



(a) The image generated from the generator

(b) The target image

Figure 4: The generated 22nd frame and the target

4 Conclusion

In conclusion, many researchers use recurrent neural networks to implement video prediction. We still use the transformer model to execute this task. We can successfully implement semantic segmentation for every image through the U-net model. A 0.95 IOU score can be obtained in the training and 0.92 in the validation set. On the other hand, the ViViT model can track the stationary object very well but cannot handle tracking the moving object to a certain degree. The image-generated model cannot effectively retain long-term memory. We also used this model to generate the 12th image based on the first 11 frames, which worked very well. However, with the increasing number of generated images, the quality of the image could be clearer. As shown in Figure 4, the green and yellow spheres at the right of the image are correct because they are stationary in the process while other objects are moving. We tested different ways to utilize the features, but the generator's performance was not ideal. We also tested directly predicting the target feature instead of the target image, but the performance is not as good as generating the image.

References

[1] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3170–3180, June 2022.

[2] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction, 2018.

[3] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting, 2015.

[4] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning, 2022.

[5] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture, 2023.

[6] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[8] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).