Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Bonn-Aachen
International Center for
Information Technology

Cuda Visin Lab Report

Predicting Video Frames Using Transformer Models

Aidin Masroor

Instructed by

Angel Angel Villar-Corrales

October 2025

1 Problem Statement

- In this project, we will explore the use of transformer models for predicting future frames in video sequences. The goal is to develop a model that can accurately forecast upcoming frames based on a series of preceding frames.
- So we train a transformer model on a dataset of video sequences, where the input is a sequence of frames and the output is the predicted next frame.
- The model will be evaluated based on its prediction accuracy, using metrics such as Least Absolute Deviations (L1), Structural Similarity Index (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS).
- The model contains and encoder, a predictor (encoder) and a decoder.
- The encoder reads video frames and maps them to a feature space, the predictor reads five feature encoded vectors, which are corresporning to five consecutive video frames, and generates the next feature vector, which corresponds to the next video frame. In this way the predictor generates multiple feature vectors as predicted feature vectors. The decoder reads them and generates images for each of those feature vectors.
- This is how the model reads video frames and predicts the following video frames.
- The dataset consists of 250 videos for validation and 9737 videos for training, each video containing 24 frames of size 128x128 pixels.

2 Approach

- The model architecture is based on transformer model, and will be explained in detail in here.

2.1 Encoder

- There are two types of encoders implemented, one is object based encoder and the other is image based encoder.

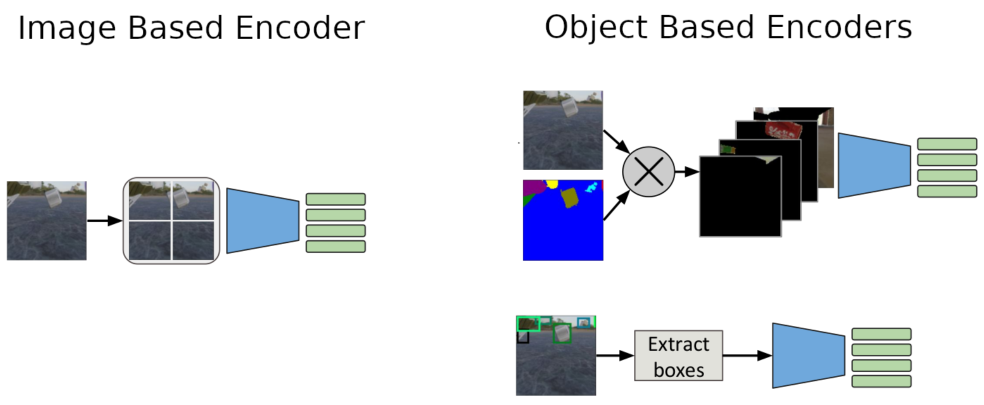

Figure 1: My figure caption

2.1.1 Object Based Encoders

- The chose encoder model here is Vision Transformer (ViT) based encoder.
- To handle inpu, either we use bounding boxes of objects in the video frames, or we use masks to encode the video frames.
- Whether we pass bounding boxes or masks along side the rgb images to the encoder, the encoder will read them and extraces each object in the frame, also encoder behaves the background as an object.
- The implemented encoder is on the main branch and accessible on the file `src/models/transformers/encoders/vit_encoder.py`. And its usage is in the file `src/models/transformers/encoders/VIT_Encoder.ipynb`.
- Data shapes are as following, bounding boxes have data shape of `[batch_size, number_of_frames_per_video, number_of_objects, 4]`, masks have data shape of `[batch_size, number_of_frames_per_video, image_height, image_width]`, and the rgb images have data shape of `[batch_size, number_of_frames_per_video, 3, image_height, image_width]`.

- The encoder outputs one feature vector for each rgb scene, so output of the encoder is a feature vector with data shape of `[batch_size, number_of_frames_per_video, feature_vector_dimension]`.

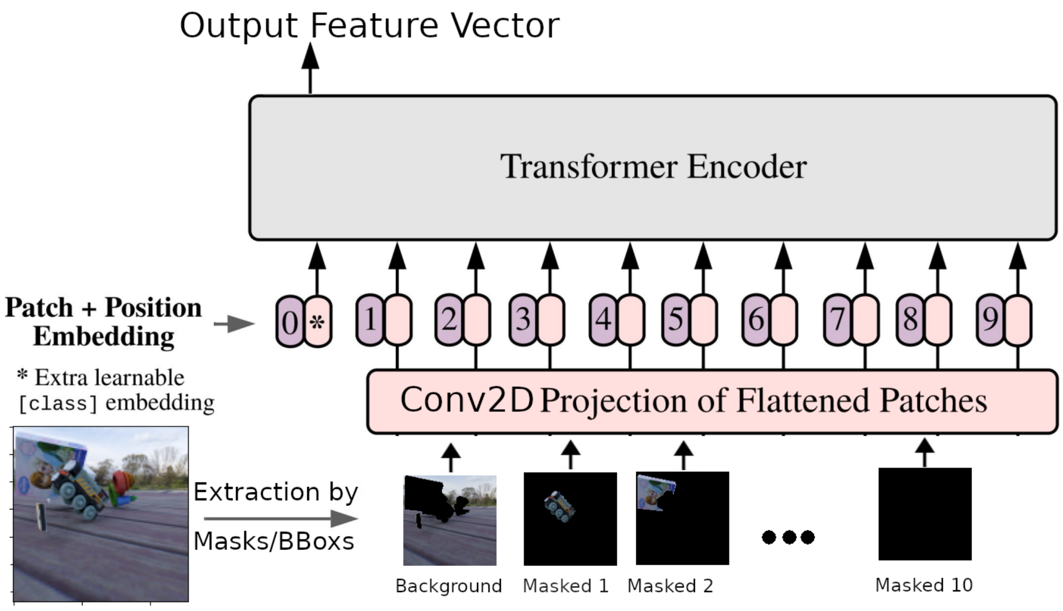
Output Feature Vector
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Conv2D Projection of Flattened Patches
Extraction by Masks/BBoxs
Background
Masked 1
Masked 2
Masked 10

Figure 2: My figure caption

2.1.2 Image Based Encoder

- To see the codes for this encoder please refer to the branch *Image_Based_Branch*, and the file `src/models/transformers/encoders/image_level/ vit_like_encoder.py`.
- The structue here is similar to the object based encoder, but here we do not pass bounding boxes or masks to the encoder, instead we pass patched images to the ViT encoder.

2.2 Evaluation

- For evaluation we use three metrics, Least Absolute Deviations (L1), Structural Similarity Index (SSIM) and Learned Perceptual Image Patch Similarity

(LPIPS). Each have their owm characteristics and advantages, and their contribution to the final loss should be weighted. The weights used for L1, SSIM and LPIPS are 1.0, 0.5 and 0.1 respectively.

- During training the loss value when we used masks instead of bounding boxes, dropped quicker. So we mostly continued our evaluation using mask extractors.
- After 60 epochs of training the model using masks, the loss value decreased but the reconstruction of decoder images were not satisfactory. mostly images were nosity like images.

3 Challenges

4 Results