Video Activity Recognition

Prepared by:[Ali Mahran](mailto:alimahran67@gmail.com)

# Approach:

For the approach I employed a pretrained network, namely EfficientNet B0 as it is the smallest and least resource-intensive.

The model is a normal 2D CNN that takes in a batch of images for training or fine-tuning and converges based on the calculation of the cross-entropy loss.

The preprocessing that was done on the data was just resizing and normalization. Data augmentation was used in training the model to add diversity to the training set and the training was done on a 4GB RTX 3050 Nvidia GPU.

During inference, the videos are taken and divided frame by frame and given to the model , the rolling average is taken for some numbers of predictions and the class mapping Is done

# How to run the code:

First create an environment with python version 3.9 and in the directory of the project install the necessary packages using the command “ pip install -r requirements.txt”

To run on video , open the **inference\_video.py** and specify the input video to the **VIDEO\_PATH** variable and the output directory to the **OUT\_DIR** variable as shown in the image below

A computer screen shot of a black background

Description automatically generated

## Results:

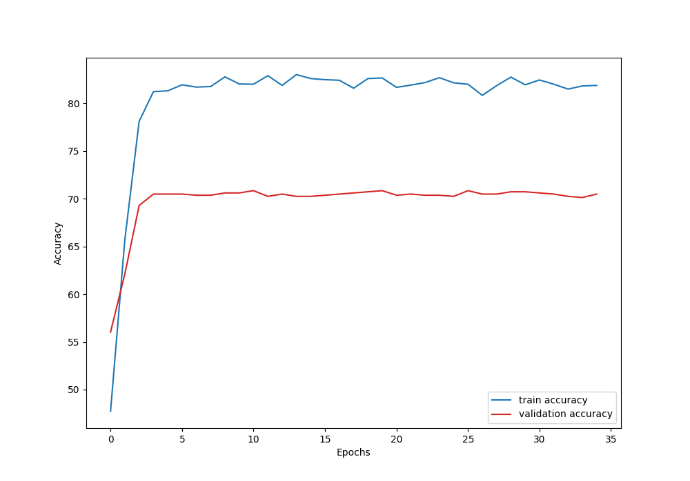
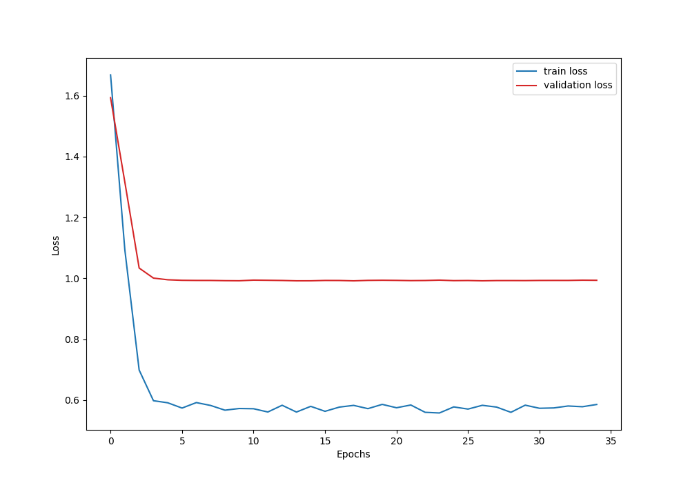
A person running on a beach

Description automatically generated

# Dataset:

The dataset used Is a dataset in Kaggle that can be found through this link https://www.kaggle.com/code/kirollosashraf/human-action-recognition-har

# Metrics & Logs



A screenshot of a computer screen

Description automatically generated

# Limitations

* The model plateaued and didn’t converge to a better loss or accuracy even though a learning rate scheduler was used.
* The data is not appropriate for training a model to work on videos.
* The data was not sufficient, and the test set didn’t have a ground truth.
* 3D models are much of a hassle to implement and require a lot of computational resources.

# Future Improvements:

* The methodology could be improved by incorporating LSTM, RNN, or GRU units with a pretrained feature extractor to better capture the temporal features along with the spatial features.
* A 3D CNN model could also be used on a dataset of videos that takes in (batch,sequence\_of\_frames,channels,frame\_width,frame\_height) instead of a normal 2D CNN that takes in a batch of arbitrary images.