Report

Working Process

I first used the combination of LSTM and some convolutional network, which is for single image recognition. For example, Resnet50+LSTM and VGG16+LSTM. But these solutions can only get 80%~90%.

Then I asked ChatGPT for help and it provided some pretrained models, then I chose Video Swin Transformer. But it can only get nearly 92%. Then I added the normalization process to the data preparation. And the mean value and standard deviation is from template code from ChatGPT, which increases the accuracy to 94%.

Finally, I notice that the frame list provided is the first 16 frames of each video, which is hard to represent the entire video. So, with the help of GPT again, I wrote a program for extracting 32 frames evenly throughout the entire video and compressed them to reduce occupation of disk space and increase speed of data loading. Then I got 98% and stopped advancing.

File List

- `train.py`: main process of training, which controls the selection of optimizer, learning rate and number of epochs.
- `test2csv.py`: get the submission .csv file
- `makedata.py`: to get the frame list described above.
- `models.py`: all the models I tried.
- `dataset.py`: data loader for training and testing.

All the data files are in `data/` while models are in `models/`, and submission files are in `output/`. These three directories should be created manually before training and testing. The training process will create `run/` automatically for storing training result. Additionally, I modify the directory structures of `hw_16fpv/` and `hw3_video/`, because their subfolder structure is unnecessarily nested.

Then the frame list made by `makedata.py` is in `hw_32fpv/`, which has a similar structure with `hw_16fpv/`.

Reproduce process

```
python ./train.py transformer_32fpv --epochs 7
python ./test2csv.py
```

Screenshot