

FILM for z-upscaling

Fisseha Ferede

October 29, 2024

Introduction

- ▶ **Goal:** A pipeline built to enhance the axial resolution of medical image volumes of different modalities and achieve isotropic resolution.
- ▶ **Technique:** A motion estimation based spatial interpolation technique motivated by state-of-the-art video resolution enhancement (temporal interpolation) methods.

Training FILM on a new dataset

Dataset structure

- ▶ To train film on new dataset, the dataset should be structured as a sequence of three images I_1, I_2, I_3 .
- ▶ Images I_1 and I_3 will be used to interpolate I_2 .
- ▶ The sequence should have the following structure.

```
<path_to_training_data>/sequences/00/00/img1.png  
                                /img2.png  
                                /img3.png  
                                .  
                                .  
                                /04/img1.png  
                                /img2.png  
                                /img3.png  
<path_to_training_data>/trainlist.txt
```

Dataset structure Cont...

- ▶ Where the "sequences" directory contains sequences of three images and "trainlist.txt" contains list of sequence paths such as 00/00, 00/01....
- ▶ Images I_1 and I_3 will be used to interpolate I_2 .
- ▶ Sample datasets in such structures can be found under the following path.

```
..common/Ferede/FILM/frame_interpolation_main/  
    training_data
```

Tensorflow TfreCORD

- ▶ FILM expects the training frame triplets in tensorflow TfreCORD storage format.
- ▶ To store your training data in TfreCORD format, run the following script with appropriate arguments.

```
..common/Ferede/FILM/frame_interpolation_main/  
film_video_dash_tfrecord.sh
```

Bash Script to store data in Tfrecored

- ▶ The bash script has three input arguments: *input_dir* (location of your dataset sequence), *input_triplet_list_filepath* (text file containing list of triplet paths) and *num_shards* (number of tfrecord shards).

```
#!/bin/bash
.
python3 ../Ferede/FILM/frame_interpolation_main/
        datasets/create_vimeo90K_tfrecord.py
--input_dir=../Ferede/FILM/frame_interpolation_main/
        training_data/raw_vimeo_triplet \
--input_triplet_list_filepath=../training_data/
        raw_vimeo_triplet/trainlist.txt \
--num_shards=200
```

Training FILM

- ▶ Once you have your dataset in tfrecord format, you can train FILM on the new dataset using the bash script under the following path with appropriate arguments.

```
.../common/Ferede/FILM/frame_interpolation_main/  
    film_video_bash_train.sh
```


Training FILM cont...

- ▶ The bash script has three input arguments: *gin_config* (configuration file containing info about training parameters), *base_folder* (main path where your training model is) and *label* (name of your training).

```
#!/bin/bash
```

```
python3 ./Ferede/FILM/frame_interpolation_main/  
        training/train.py \  
--gin_config ./Ferede/FILM/frame_interpolation_main/  
        training/config/film_net-style-vimeo.gin \  
--base_folder ./Ferede/FILM/frame_interpolation_main \  
--label my_first_train_vimeo
```

Saving trained model

Saving Trained Models

- ▶ By default trained model will be saved at the end of the training. However, if you would like to save model during training at its current stage, you can call the following script.

```
.../common/Ferede/FILM/frame_interpolation_main/  
    film_video_bash_savedmodel.sh
```

Bash Script to Save Model

```
#!/bin/bash  
.  
python3 ../Ferede/FILM/frame_interpolation_main/  
training/build_saved_model_cli.py  
    --base_folder ../Ferede/FILM/  
    frame_interpolation_main \  
    --label test_run_ft_droso_
```

Fine-tuning FILM on a new dataset

Fine-tuning FILM

- ▶ Fine-tuning on FILM is similar to training on a new dataset, except that we add pre-trained weights to initialize the learning.
- ▶ This is done by adding the Checkpoint, Index and Data files of the pretrained model under *train* sub-folder of your fine-tuning label.
- ▶ The checkpoint metadata contains the checkpoint path of 10 most recent trained weights (saved every 3,000 iteration).
- ▶ Data files store the actual data for the variables in the model, such as the weights and biases.
- ▶ Index ensures that TensorFlow knows which part of the .data files correspond to which variables in the model (such as layer names).

Bash script for Fine-tuning FILM

- ▶ Same as training on a new dataset, except that *my_first_fine-tune_FlyEM/train*, contains the Index, data and checkpoint files.
- ▶ An example of how this looks like is provided in

```
/Ferede/FILM/frame_interpolation_main/  
my_first_fine-tune_FlyEM
```

```
#!/bin/bash
```

```
python3 ./Ferede/FILM/frame_interpolation_main/  
        training/train.py \  
--gin_config ./Ferede/FILM/frame_interpolation_main/  
            training/config/film_net-style-vimeo.gin \  
--base_folder ./Ferede/FILM/frame_interpolation_main \  
--label my_first_fine-tune_FlyEM
```

Inference on FILM

Inference using FILM

- ▶ FILM is already integrated with IPP, so a user can directly use FILM using IPP.
- ▶ To run from cluster, a user can run the following bash script provided that it has a singularity container or Conda environment containing all the dependencies.

Bash Script for Inference using FILM

- ▶ Bash script for inference can be found here:

```
../Ferede/FILM/frame_interpolation_main/  
film_video_bash.sh
```

```
#!/bin/bash  
python3 ../Ferede/FILM/frame_interpolation_main/  
evAl/interpolator_cli.py  
--pattern "../input_path_of_2Dimgs_or_3D_vols/*" \  
    frame_interpolation_main \  
--outfile ../output_path \  
--times_to_interpolate 2 \  
--output_volume "False" \  
--remove_sliced_volumes "False"
```

References



Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless.

Film: Frame interpolation for large motion.

In *European Conference on Computer Vision*, pages 250–266.
Springer, 2022.