**Image to animation based model.**

**Model link** : [**https://github.com/AliaksandrSiarohin/first-order-model**](https://github.com/AliaksandrSiarohin/first-order-model)

**Model description:** The goal of the model is to implement the image into animation or in first motion with reference to another or different image.

**Example:** We use a sample video of motion as a source and with reference to the video , we use images in target and as an implementation it will convert the target images into motion with the reference of source.

**Dataset:** We can use different types of dataset in different categories

* For human face movement - VoxCeleb Dataset
* For body movement - Fashion Dataset
* For animal movement - MGIF Dataset

**Links to download:**

1. Bair - [downloaded](https://yadi.sk/d/66vNK0EwJcVkPg).
2. Mgif- [downloaded](https://yadi.sk/d/5VdqLARizmnj3Q).
3. Fashion - [from](https://vision.cs.ubc.ca/datasets/fashion/).
4. Taichi - <https://github.com/AliaksandrSiarohin/video-preprocessing>.
5. Nemo - <https://github.com/AliaksandrSiarohin/video-preprocessing>.
6. VoxCeleb - [ttps://github.com/AliaksandrSiarohin/video-preprocessing](https://github.com/AliaksandrSiarohin/video-preprocessing).

**Libraries installation:**

* cffi==1.11.5
* cloudpickle==0.5.3
* cycler==0.10.0
* dask==0.18.2
* decorator==4.3.0
* imageio==2.3.0
* kiwisolver==1.0.1
* matplotlib==2.2.2
* networkx==2.1
* numpy==1.15.0
* pandas==0.23.4
* Pillow==5.2.0
* pycparser==2.18
* pygit==0.1
* pyparsing==2.2.0
* python-dateutil==2.7.3
* pytz==2018.5
* PyWavelets==0.5.2
* PyYAML==5.1
* scikit-image==0.14.0
* scikit-learn==0.19.2
* scipy==1.1.0
* six==1.11.0
* toolz==0.9.0
* torch==1.0.0
* torchvision==0.2.1
* tqdm==4.24.0

Clone and install all the packages listed for model computation , then we need to upload a source and video for the computation with the help of imageio.imread.

* Resize to an standard format used in the model is 256\*256 in image and video format in mp4 format else we can use python crop.py to use this crop images and videos.
* To use as a script follow and clone the model:
  + git clone https://github.com/1adrianb/face-alignment
  + cd face-alignment
  + pip install -r requirements.txt
  + python setup.py install

**Pre Trained model/checkpoints:**

To avoid model training. We can use pre-trained model checkpoints for easy evaluations and load into the model.

* Link: [google-drive](https://drive.google.com/open?id=1PyQJmkdCsAkOYwUyaj_l-l0as-iLDgeH) or [yandex-disk](https://yadi.sk/d/lEw8uRm140L_eQ).

**image-animation transformation -**  
The model can uses a relative keypoint to animate the objects but here the model uses absolute coordinates instead,

**Training:**

* Either we can train our model by creating a dataset or we can use pretrained weights of the model.
* It is mandatory to use pytorch==1.0.0 for training

**Training on custom dataset:**

* To set training on custom data , we have created a separate directory for train and test cases in data/dataset\_name and also make changes in hyper parameters to config.
* Create a config.yaml file and place this file in the directory of data/dataset\_name.

**Evaluate checkpoints**

* To evaluate on the dataset the checkpoints will be stored into the dive folder of checkpoints.
* The checkpoints for images stored in PNG format and for video it will be stored in MP4 format.

**Image animation**

Just specify the folder into the "source,driving" pairs in the corresponding.

* ! python run.py --config config/dataset\_name.yaml --mode animate --checkpoint path/to/checkpoint

**Predictions**:  
To predict the results with source and target , follow the below command:

* from demo import make\_animation
* from skimage import img\_as\_ubyte
* predictions = make\_animation(source\_image, driving\_video, generator, kp\_detector, relative=True)
* #save resulting video
* imageio.mimsave('../generated.mp4', [img\_as\_ubyte(frame) for frame in predictions])
* #video can be downloaded from /content folder
* HTML(display(source\_image, driving\_video, predictions).to\_html5\_video())