**First order motion model - Paper Wise Research**

**Title -** First Order Motion Model for Image Animation

**Methodology:**

* Propose a method to generate a video sequence wherein an object in a source image will be in motion according to the reference motion of a driving video.
* In this the network model concludes the target motion and combines the extracted motion in the source image with help of driving video.
* In one way to understand , we are transforming the motion/facial expression from one source of video to the target of image as a result.
* In continuation to our method the driving videos and source images should be cropped before it can be used wherein we use FFMPEG to crop the video and image.
* The transformation behind this motion is purely based on keypoints detection where the model will accept the key points of both source and video and to generate future frames it will first detect keypoints of a moving object and predict future motion as a sequence of keypoints.

**Technique:**

* This model take Input of source image as driving frames and then extract keypoints using unsupervised keypoint detector which is provide local affine transformations with respect to the

reference after they use dense motion network for generation in optical flow.(https://github.com/YunjiKim/Unsupervised-Keypoint-Learning-for-Guiding-Class-conditional-Video-Prediction)

* The target image and the outputs of the dense motion network are used by the generator to render the target image.
* Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs)have been used to transfer facial expressions between videos to image transformation.
* It is important to install model dependencies for custom training specifically Pytorch version should be 1.0.0
* There are 2 different ways of performing animation: Absolute keypoint and Relative keypoint
* Animation with absolute keypoints is performed using the absolute positions of the driving video and appearance of the source image where this will lead to poor result as shape is transferred with irrelevant content.
* Animation using relative key points where we first estimate the relative movement of each keypoint, then we add this movement to the absolute position of keypoints in the source image which leads to better performance.
* In such cases the video frame and the source images have the object with the same pose and style.

**Pros:**

* This one requires no knowledge of the image and better generalizes .
* It works on the human face and the full body also works on animations.

**Cons:**

* It leaves artifacts in output when there is sudden movement. It performs less accurately on fast video.

**Link:**

* [**https://papers.nips.cc/paper/8935-first-order-motion-model-for-image-animation.pdf**](https://papers.nips.cc/paper/8935-first-order-motion-model-for-image-animation.pdf)

**Result:**

* [**https://drive.google.com/drive/folders/1UWijKN6cfstvlCutGnFLT3Wmu7ULuSFA?usp=sharing**](https://drive.google.com/drive/folders/1UWijKN6cfstvlCutGnFLT3Wmu7ULuSFA?usp=sharing)

**------------------------------------------------------------------------2------------------------------------------------------------------**

**Title -** Few-Shot Adversarial Learning of Realistic Neural Talking Head Models

**Method**

* The model uses the image-to-image translation architecture wherein it can downsampling and upsampling the layers with blocks. The parameters serve as the affine coefficients of instance normalization layers which follow the adaptive instance normalization technique.
* The model contains three networks to work upon , first is an embedder in which we take video as an input , divided into frames(vector) that is invariant to the pose and mimics in a particular frame and that all embedded in this network.
* Second is the generator which is trained to maximize the similarity between its outputs and the ground truth frames the parameters of the generator are split into two sets: the person-generic parameters and the person-specific parameters.
* Third is the discriminator which takes a video frame , and an image with the index sequence.The discriminator contains a ConvNet that maps the input frame and the landmark image into an N-dimensional vector.It indicates and matches whether the input frame is resembling with frame of video sequence or not

**Technique**

* The images that are fed from voxceleb2 are resized from 224x224 to 256x256 by using zero-padding. This is done so that spatial dimensions don't get affected when passing through downsampling layers.
* The embedder uses 6 downsampling residual blocks in the middle a self-attention layer is added and in the end the output takes the last residual block which is resized to a vector of size 512 through max pooling.
* The generator and discriminator uses the same architecture as the embedder

**Pros**

* It able to learn highly realistic and personalized talking head models of new people and even portrait paintings

**Cons**

* the key limitations of our method are the mimics representation (in particular, the current set of landmarks does not represent the gaze in any way) and the lack of landmark adaptation.

**Link**

* [**https://arxiv.org/pdf/1905.08233.pdf**](https://arxiv.org/pdf/1905.08233.pdf)

**Result**

* [**https://drive.google.com/drive/folders/1B7Fr4fZ9xBuHAnqVkNH-aeoaGttT83R8?usp=sharing**](https://drive.google.com/drive/folders/1B7Fr4fZ9xBuHAnqVkNH-aeoaGttT83R8?usp=sharing)

**----------------------------------------------------------------------3---------------------------------------------------------------------**

**Tittle:** Monkey-Net Model

**Method**

* The paper approaches image animation in which a given input image with a target driving video results in a moving motion in image as an output.
* The model is having a deep architecture that decouples appearance and motion information.It consists of three main modules: (i) a Keypoint Detector (ii)Dense Motion prediction network for generating dense heatmaps from sparse keypoints. (iii)Motion Transfer Network, which uses the information extracted from the input image to synthesize the output frames.

**Technique**

* The Monkey-Net uses the source image and frame-by-frame from the driving video via keypoint annotations which is based on self-supervised learning
* The Monkey-Net architecture is having two frames of size extracted from the same video and image jointly learn a keypoint detector with a generator network wherein the network should be able to reconstruct the keypoint locations which describe motion as well as the object geometry with the optical flow between the keypoints.
* The keypoint detector is to predict keypoint locations that capture the object structure where the motion is having high probability.

**Pros**

* Free image animation on cropped face.Appropriate results on full body animation on taichi dataset(On HD)

**Cons**

* Poor generation quality in the case of large object pose change

**Link**

* [**https://github.com/AliaksandrSiarohin/monkey-net**](https://github.com/AliaksandrSiarohin/monkey-net)

**-------------------------------------------------------------4------------------------------------------------------------------------------**

**Tittle :** Motion Supervised co-part Segmentation

**Method**

* The model is based on a self supervised learning method wherein the model does not require a big data to train on which saves a lot of time for model training.
* The network learns to predict the segmentation (part specific) from the source image and driving video and transform it into resultant.
* To demonstrate the precise information about object parts can be extracted by leveraging motion information from vast amounts of unlabeled videos thus self-supervised co-part segmentation method is involved into this which introduces semantic representations without leaking.
* In order to test under the dataset , the model is trained on two datasets : Tai-Chi-HD and VoxCeleb.

**Technique**

* The model working is based on two modules: Segmentation Module and Reconstruction Module.
* The Segmentation Module predicts the segmentation maps for the source image and target video, along with the affine motion.On the other hand, the Reconstruction Module, is in charge of reconstructing the target video from the source image and the segmentation module outputs.
* Both the models works to computes a background visibility mask and an optical flow from the segmentation maps and the affine motion parameters where it reconstructs the target frame by warping the features of the source frame.
* Model is using an auto-encoder architecture for landmark locations.

**Pros**

* Specific part can be swapped,Working on pretrained data very smoothly

**Cons**

* Poor results on full body image and video.Tested on custom image and video

**Link**

* [**https://github.com/AliaksandrSiarohin/motion-cosegmentation**](https://github.com/AliaksandrSiarohin/motion-cosegmentation)

**--------------------------------------------------------------------5-----------------------------------------------------------------------**

**Tittle :** Everybody Dance Now

**Method**

* This model is based on video to video translation using pose estimation.The paper resembles the concept of image-video animation where the model has source video and it transforms the motion into target as same as the source motion (Motion replica).
* The method behind the model is based on keypoint-based pose preserves motion and use of pose detectors like OpenPose, It is a representation for frame-to-frame transfer so to transfer motion from source to target in the same pose as the source we have used OpenPose.
* The object of the model is to generate a new video of the target replicating the same motions as the source and the target have.for this model is divided into pose detection, global pose normalization, and mapping from target.
* In pose detection , the model approached OpenPose to detect and use the key points coordinate moreover it cerates stick figure for the given source video.
* Second phase where it differentiate the pose figure with the target body shape and coordinate location within the frame and recontinue to the mapping phase
* Third phase mapping phase resembles the pose stick figure with the target video with the help adversarial approach.

**Technique:**

* The model uses the FACE-GAN and VGG-19 network for implementation.
* It is using generators and discriminator of pix2pixHD models with the help of Face-GAN , In this it is using frame by frame smoothing and temporal smoothing with the approach of Face-GAN**.**

**Pros:**

* Model is so smooth to work upon.
* It is based on full body animation with respect to source and driving video.
* Results are satisfactory and also model covers two different driving video and transform the motion on the source image without any distortion as we can see in paper results.

**Cons:**

* Model is not available on any website or platform.

**Link**

* [**https://arxiv.org/pdf/1808.07371.pdf**](https://arxiv.org/pdf/1808.07371.pdf)

**------------------------------------------------------------6-------------------------------------------------------------------------------**

**Tittle:** High-Resolution Neural Face Swapping for Visual Effects

**Method**

* Method proposes a full face-swapping pipeline including a contrast and light-preserving compositing step and a landmark stabilization procedure that allows for generating temporally stable video sequences.
* It is a progressively trained multi-way comb network and a light- and contrast-preserving blending method where we can use progressive training which enables generation of high-resolution images which allows us to achieve higher fidelity in generated expressions in face swapping.
* When compositing the generated expression onto the target face, we show how to adapt the blending strategy to preserve contrast and low-frequency lighting. Finally, we incorporate a refinement strategy into the
* With the help of face landmark stabilization algorithm to achieve temporal stability, which is crucial for working with high-resolution videos also will keep an eye on the quality of the swap results.

**Technique**

* Domain transfer technique, encoder-decoder model named comb model, Tensorflow Implementation of SSIM, Poisson blending for contrast preserving.
* Image to image translation with GAN.Usage of face-alignment model to crop the image properly with the base of face recognition model.
* Usage of Morphable Model (on geometry basis ) to classic form, 3D morphable models live in a vector space.

**Pros**

* It works on higher resolution images-1024p.
* It provides reicher representation of behaviour.
* It creates photorealistic images with artifacts in generated images.
* Due to the proposed face view interpolation, the results are slightly smoothed.
* Comparison of results with different and suitable methods like (two way model, nearest neighbour to the eight-way model result (landmark space), nearest neighbour to the eight-way model result (RGB space).

**Cons**

* Face swapping is not our goal to achieve but we are targeting the motion from the video to images which did not work for us in this model.
* Lack temporal stability in the generated faces

**Link**

* [**https://s3.amazonaws.com/disney-research-data/wp-content/uploads/2020/06/18013325/High-Resolution-Neural-Face-Swapping-for-Visual-Effects.pdf**](https://s3.amazonaws.com/disney-research-data/wp-content/uploads/2020/06/18013325/High-Resolution-Neural-Face-Swapping-for-Visual-Effects.pdf)

**-------------------------------------------------------------------7------------------------------------------------------------------------**

**Tittle :** Faceswap

**Method**

* FaceSwap is a tool that utilizes deep learning to recognize and swap faces in pictures and videos.
* Faceswap image-based face swapping algorithm, which can be used to replace the face in the reference image with the same facial shape and features as the input face.
* With a use of face alignment which is made up of detected facial landmarks, so that the aligned input face and the reference face are consistent in size and posture.

**Technique**

* Tensorflow neural networks , Keras , OpenCV , Skit-learn , Pillow and Matplotlib
* It also uses a complete package which can work on cross platforms like Windows,Linux and MacOS.
* Depending on the OS environment , setup files for all the platforms are available.
* Depending on CUDA setup we need to install the dependencies accordingly

**Pros**

* It creates realistic looking images .

**Cons**

* It requires more input data as compared to other methods.

**Link**

* [**https://github.com/deepfakes/faceswap**](https://github.com/deepfakes/faceswap)

**------------------------------------------------------------------8-------------------------------------------------------------------------**

**Tittle:** pix2pixHD

**Method**

* Pytorch implementation of our method for high-resolution (e.g. 2048x1024) photorealistic image-to-image translation.
* It can be used for turning semantic label maps into photo-realistic images or synthesizing portraits from face label maps.
* We propose a method to generate diverse results given the same input, allowing users to edit the object appearance interactively.
* It supports interactive semantic manipulation although we extend our method in two directions - we use instance-level object segmentation information, which can separate different object instances within the same category which enables flexible object manipulations.
* Second we propose a method to generate diverse results given the same input label map which allows the user to edit the appearance of the same object interactively.

**Technique**

* Model used Conditional GAN , multi scale discriminatior and instance map.
* Access towards pytorch and dominate to implement the model.
* We use the Cityscapes dataset. To train a model on the full dataset, please download it from the official website.(https://www.cityscapes-dataset.com/)
* Some examples like Cityscapes (test images) are included in the datasets folder.

**Pros**

* it works on higher resolution images and also works on other objects .
* Results are generated with adversarial training only or like pre-trained networks (e.g. VGGNet)

**Cons**

* It does not produce accurate result as compared to first order model.
* Results are often limited to low resolution and still far from realistic

**Link**

* [**https://github.com/NVIDIA/pix2pixHD**](https://github.com/NVIDIA/pix2pixHD)

**-----------------------------------------------------------------------9--------------------------------------------------------------------**

**Tittle:** Facerig

**Method**

* The project includes the input from retrieves images from a camera, detects faces and determines a head pose of the dominant face using facial landmark detection.
* Application rigs detected head pose to head model. The render has rather general-purpose nature but a facade for face rigging is defined.
* First as library through input/output streams. Second as a stand-alone application.

**Technique**

* OpenCV for face detection.
* Dlib for facial landmark detection. (Will be downloaded & compiled automatically by CMake.)
* OpenGl for render implementation.
* This is a lightweight render based on Phong reflection model which defines how a Material’s color varies depending on factors such as surface orientation, viewer direction, and lighting

**Pros**

* It creates more accurate results and realistic results .
* Fun and easy to use

**Cons**

* it can not perform 2d image transformation as it only transforms facial moments

**Link**

* [**https://github.com/jan-skarupa/facerig**](https://github.com/jan-skarupa/facerig)

**--------------------------------------------------------------------10---------------------------------------------------------------------**

**Tittle :** TransMoMo: Invariance-Driven Unsupervised Video Motion Retargeting

**Method**

* We present a lightweight video motion retargeting approach that is capable of transferring motion in spite of structural and view-angle disparities between the source and the target.
* The proposed method can be trained in an unsupervised manner by exploiting invariance properties of three orthogonal factors of variation including motion, structure, and view-angle.
* Speciﬁcally with loss functions carefully derived based on invariance, we train an auto-encoder to disentangle the latent representations of such factors given the source and target video clips..

**Technique**

* Pytorch , FFMPEG-Imageio, Matplotlib.
* We exploit the invariance property of three factors: motion, structure, and view-angle.
* Factors of variation are enforced to be independent of each other held constant when other factors vary.
* The invariance properties allow us to derive a set of purely unsupervised loss functions to train an auto-encoder for disentangling a sequence of skeletons into orthogonal latent representations.
* The learned latent representation is meaningful when interpolated where body structure is interpolated on the horizontal axis while motion is interpolated on the vertical axis.

**Pros**

* Smooth transform from source to target.
* it combines motion parameters of source and target in the form of a skeleton.
* The combinely parameters are view angle,motion,structure of skeleton which we can see in retargeted skeleton to video conversion with the help of another model.
* We can explicitly manipulate the view of the decoded skeleton in the 3D space rotating it before projecting down to 2D.

**Cons**

* Limitation in large structure variations or extreme motion in videos

**Link**

* [**https://github.com/yzhq97/transmomo.pytorch**](https://github.com/yzhq97/transmomo.pytorch)

**--------------------------------------------------------------------------------------------------------------------------------------------**