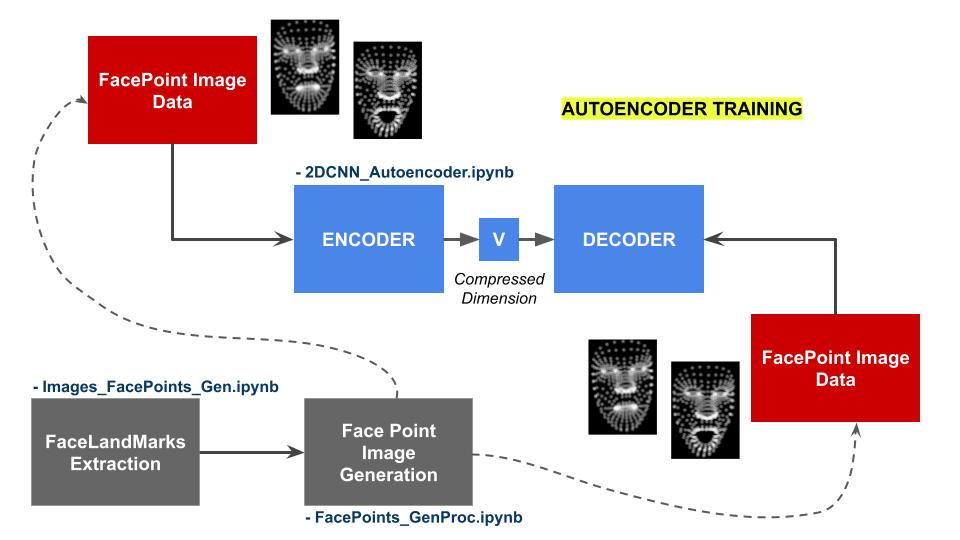
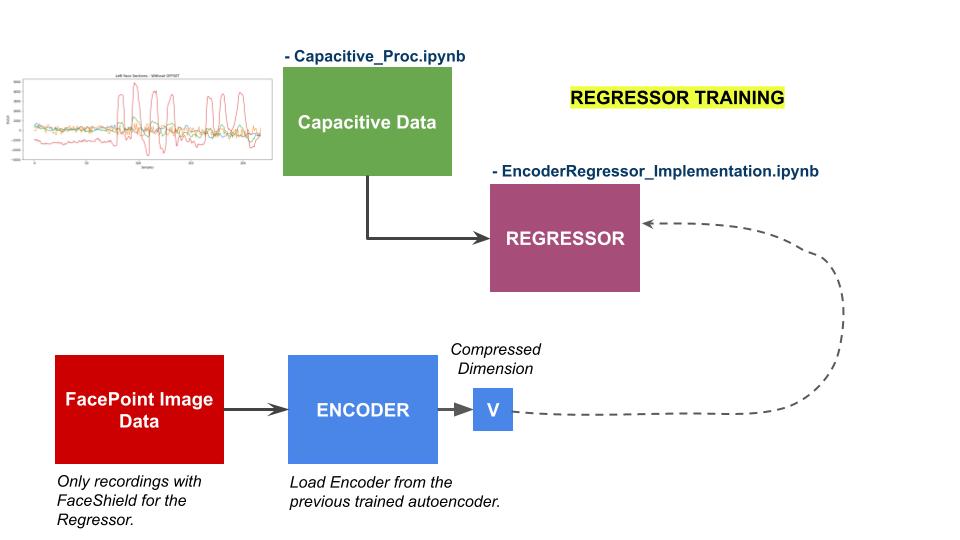
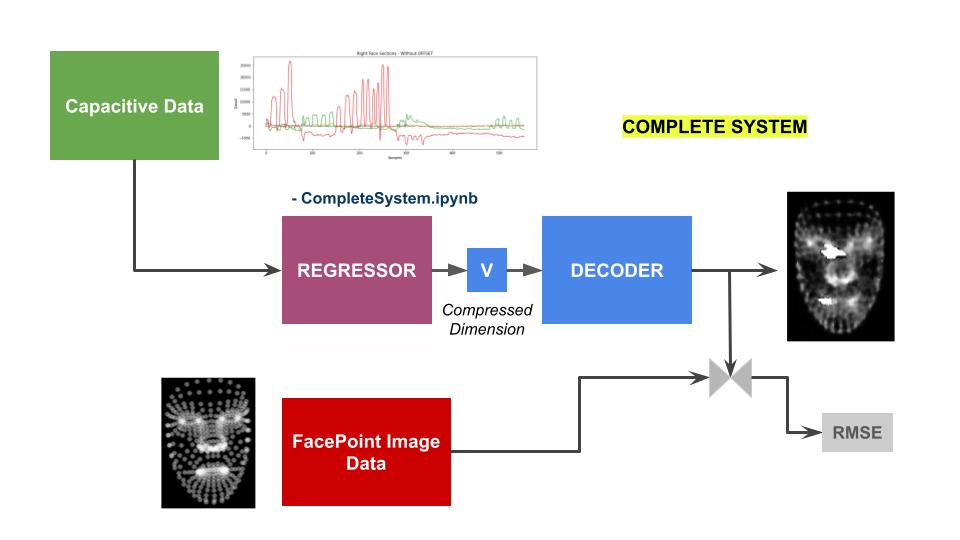
**MODEL TRAINING STEPS**

****

****

**FacePoints\_GenProc.ipynb**

**Task Description:** Generates the face landmark points from a video. Tasks include: resampling and synchronization of input data with respect to capacitive data.

**Input Data:** Video (.MOV)

**Output Data:** Numpy file (.npy)

**Input Files:**

**For Autoencoder:**

* Train:

/MyDrive/AutoEncoder/Face\_Videos\_Train/Video\_Face(X).MOV

* Test:

/MyDrive/AutoEncoder/Face\_Videos\_Test/Video\_Face(X).MOV

**For Regressor (Video with Face Shield):**

* Train:

/MyDrive/AutoEncoder/FaceShield\_Videos\_Train/Video\_FaceShield(X).MOV

* Test:

/MyDrive/AutoEncoder/FaceShield\_Videos\_Test/Video\_FaceShield(X).MOV

**Output Files:**

**For Autoencoder:**

* Train:

/MyDrive/AutoEncoder/FacePoints\_numpy\_Train/Video\_Face\_np(X).npy

* Test:

/MyDrive/AutoEncoder/FacePoints\_numpy\_Test/Video\_Face\_np(X).npy

**For Regressor (Video with Face Shield):**

* Train:

/MyDrive/AutoEncoder/FaceShieldPoints\_numpy\_Train/Video\_FaceShield\_np(X).npy

* Test:

/MyDrive/AutoEncoder/FaceShieldPoints\_numpy\_Test/Video\_FaceShield\_np(X).npy

**Images\_FacePoints\_Gen.ipynb**

**Task Description:** Generates 255x255 gray scale images from face landmark points.

**Input Data:** Numpy file (.npy)

**Output Data:** 255x255 gray scale images

**Input Files:**

**For Autoencoder:**

* Train:

/MyDrive/AutoEncoder/FacePoints\_numpy\_Train/Video\_Face\_np(X).npy

* Test:

/MyDrive/AutoEncoder/FacePoints\_numpy\_Test/Video\_Face\_np(X).npy

**For Regressor (Video with Face Shield):**

* Train:

/MyDrive/AutoEncoder/FaceShield\_Videos\_Train/Video\_FaceShield(X).npy

* Test:

/MyDrive/AutoEncoder/FaceShield\_Videos\_Test/Video\_FaceShield(X).npy

**Output Files:**

**For Autoencoder:**

* Train:

/MyDrive/AutoEncoder/Image\_FacePoints\_Train/Img\_Train(X).png

* Test:

/MyDrive/AutoEncoder/Image\_FacePoints\_Test/Img\_Test(X).png

**For Regressor (Video with Face Shield):**

* Train:

/MyDrive/AutoEncoder/FaceShieldPoints\_numpy\_Train/Img\_Record(Y)/Video\_FaceShield\_np(X).png

* Test:

/MyDrive/AutoEncoder/FaceShieldPoints\_numpy\_Test/Img\_Record(Y)/Video\_FaceShield\_np(X).png

**Capacitive\_Proc.ipynb**

**Task Description:** Generates a process npy file of the capacitive data. Tasks include: synchronization, resampling with respect to video data.and generation of time window slices.

**Input Data:** JSON.

**Output Data:** Numpy file (.npy)

**Input Files:**

* Train:

/MyDrive/Regressor/Capacitive\_Data\_Train/Capacitive\_Train\_Record(X).json

* Test:

/MyDrive/Regressor/Capacitive\_Data\_Train/Capacitive\_Train\_Process(X).npy

**Output Files:**

* Train:

/MyDrive/Regressor/Capacitive\_Data\_Test/Capacitive\_Test\_Record(X).json

* Test:

/MyDrive/Regressor/Capacitive\_Data\_Test/Capacitive\_Test\_Process(X).npy

**2DCNN\_Autoencoder.ipynb**

**Task Description:** Load and pre-process all face point images to train the Autoencoder model, compute overall RMSE and predicted image from the model.

**Input Data:** FacePoint Images.

**Output Data:** autoencoder model (.h5)

**Input Files:**

* Train:

/MyDrive/Autoencoder/Image\_FacePoints\_Train/Img\_Train(X).png

* Test:

/MyDrive/Autoencoder/Image\_FacePoints\_Test/Img\_Test(X).png

**Output Files:**

* conv\_autoencoder\_CNN:

/MyDrive/Models/conv\_autoencoder\_CNN.h5

* decoder\_CNN:

/MyDrive/Models/decoder\_CNN.h5

* encoder\_CNN

/MyDrive/Models/encoder\_CNN.h5

**EncoderRegressor\_Implementation.ipynb**

**Task Description:** Loads the processed capacitive data (.npy). Computes the encoded vector from FaceShieldPoint based generated images, and trains the Regressor, considering the capacitive data as features and the image encoded vector as target.

**Input Data:** 1) encoder\_CNN, 2) capacitive data (.npy), 3) FaceShieldPoint images.

**Output Data:** autoencoder model (.h5)

**Input Files:**

* encoder\_CNN:

/MyDrive/Models/encoder\_CNN.h5'

* Capacitive Data Train:

/MyDrive/Regressor/Capacitive\_Data\_Train/Capacitive\_Train\_Process(X).npy

* FaceShieldPoint Images Train:

/MyDrive/Regressor/Image\_FaceShieldPoints\_Train/Img\_Record(Y)/Img\_Train(X).png

**Output Files:**

* model\_regresor:

/MyDrive/FaceLandmarks\_numpy/model\_CNN\_regresor.h5

**CompleteSystem.ipynb**

**Task Description:** Loads the processed capacitive data (.npy) from the test file. Computes encoded image vectors using the capacitive test data and the model\_regressor. Computes the FacePoint image with the decoder\_CNN considering the encoded image vectors as input. Evaluate overall model performance with RMSE score using the FaceShieldPoint Images Test as ground truth.

**Input Data:** 1) decoder\_CNN, 2) model\_regresor, 3) capacitive data (.npy), 4) FaceShieldPoint images.

**Output Data:** overall prediction and score.

**Input Files:**

* decoder\_CNN:

/MyDrive/Models/decoder\_CNN.h5

* model\_regresor:

/MyDrive/FaceLandmarks\_numpy/model\_CNN\_regresor.h5

* Capacitive Data Test:

/MyDrive/Regressor/Capacitive\_Data\_Test/Capacitive\_Test\_Process(X).npy

* FaceShieldPoint Images Test:

/MyDrive/AutoEncoder/FaceShieldPoints\_numpy\_Test/Img\_Record(Y)/Video\_FaceShield\_np(X).png

**Output Files:** None