Kroop AI Recruitment Assignment

The Problem

The task is to improve the quality of the region inside the mouth.

Solution I experimented on :
1 . Tried getting a pre-trained Autoencoder or GAN model and training it on our dataset to get enhanced images, but since I didn't have CUDA enabled GPU most of the options were not feasible.

2. Sharpening and Increasing the resolution of the Image
The basic way to remove blur is sharpening the image but it didn't get the clarity and it changed the texture of that particular part of the image, hence I tried to increase the resolution of the image using Super Resolution models and then sharpening the image, which gave better results.

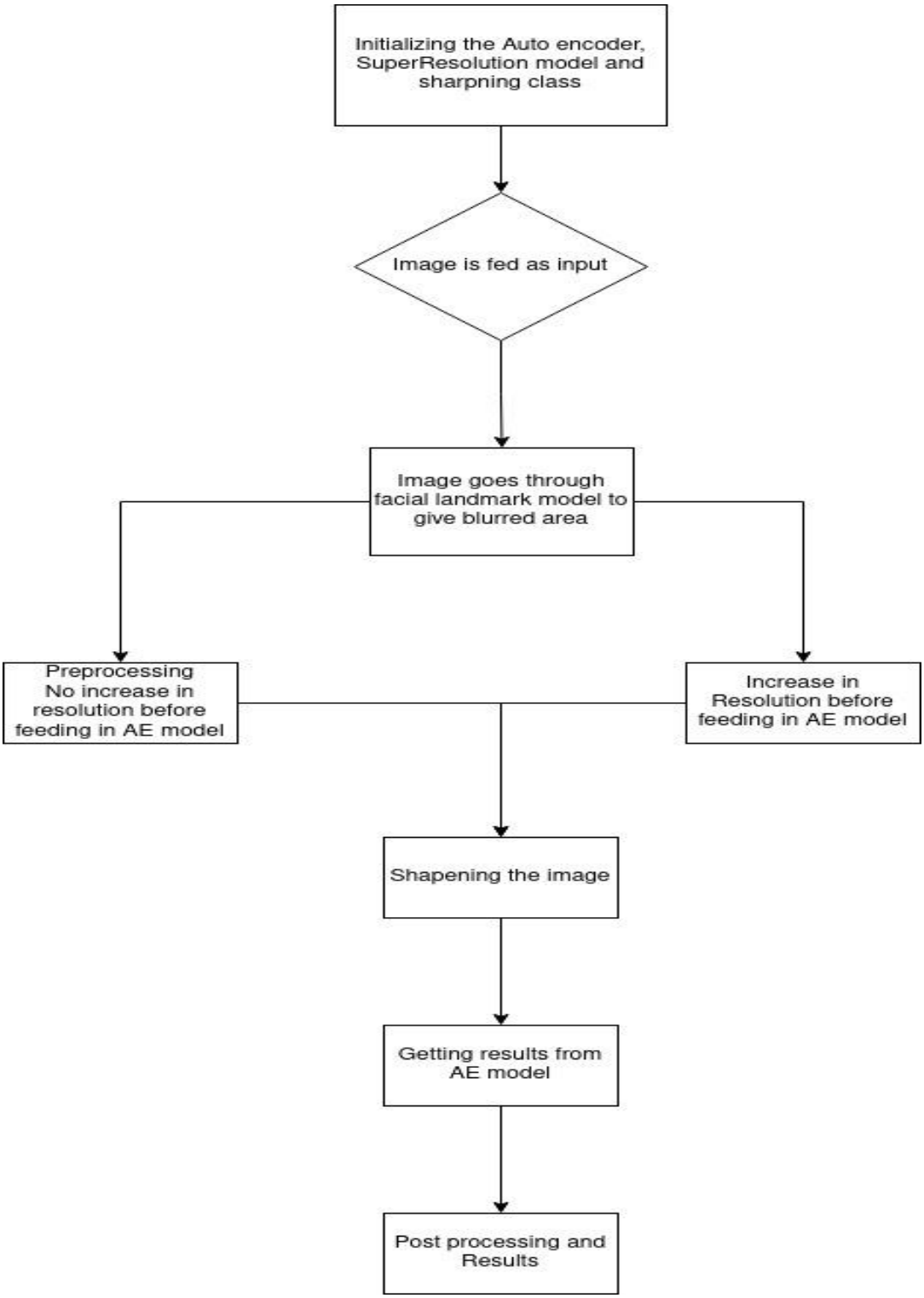
3. Training an auto-encoder model from scratch and inference on that model.

The solution uses the third way to interact with the image and get results with a bit of pre-processing with the image which comes from the 2nd way described above.

The default model for Super Resolution - EDSR
A ResNet style architecture is used without the Batch Normalization layers. They found that those layers get rid of range flexibility from the features' networks, improving the performance. This allows them to build a larger model with better performance. To counter the instability found in large models, they used residual scaling with a factor of 0.1 in each residual block by placing constant scaling layers after the last convolutional layers. Also, ReLu activation layers are not used after the residual blocks.

The default sharpening class - cv2.addweighted

The model flow -

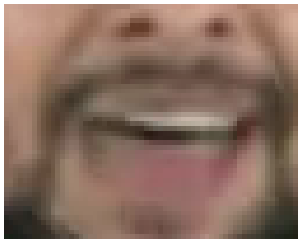


Getting the Blurred ROI:

To get the blurred ROI, facial landmarks are used where the first landmark covering the face around the mouth is found, a rectangle is defined and the image is cropped to train the model.

The process :

Finally : Cropping the Image



This image is sent into model for training and inference.

The model Summary :

Model: "encoder"

Layer (type)	Output Shape	Param #
encoder_input (InputLayer)	[(None, 128, 128, 3)]	0
conv2d_3 (Conv2D)	(None, 64, 64, 32)	896
conv2d_4 (Conv2D)	(None, 32, 32, 40)	11560
dropout_4 (Dropout)	(None, 32, 32, 40)	0
conv2d_5 (Conv2D)	(None, 16, 16, 64)	23104
conv2d_6 (Conv2D)	(None, 8, 8, 80)	46160
dropout_6 (Dropout)	(None, 8, 8, 80)	0
conv2d_7 (Conv2D)	(None, 4, 4, 128)	92288
conv2d_8 (Conv2D)	(None, 2, 2, 256)	295168
dropout_8 (Dropout)	(None, 2, 2, 256)	0
flatten_1 (Flatten)	(None, 1024)	0
latent_vector (Dense)	(None, 256)	262400

Total params: 731,576
Trainable params: 731,576
Non-trainable params: 0

Model: "decoder"

Layer (type)	Output Shape	Param #
decoder_input (InputLayer)	[(None, 256)]	0
dense_1 (Dense)	(None, 1024)	263168
reshape_1 (Reshape)	(None, 2, 2, 256)	0
conv2d_transpose_3 (Conv2DTranspose)	(None, 4, 4, 256)	590080
dropout_9 (Dropout)	(None, 4, 4, 256)	0
conv2d_transpose_4 (Conv2DTranspose)	(None, 8, 8, 128)	295040
conv2d_transpose_5 (Conv2DTranspose)	(None, 16, 16, 80)	92240
dropout_11 (Dropout)	(None, 16, 16, 80)	0
conv2d_transpose_6 (Conv2DT	(None, 32, 32, 64)	46144

conv2d_transpose_7 (Conv2DT ranspose) (None, 64, 64, 40) 23080

dropout_13 (Dropout) (None, 64, 64, 40) 0

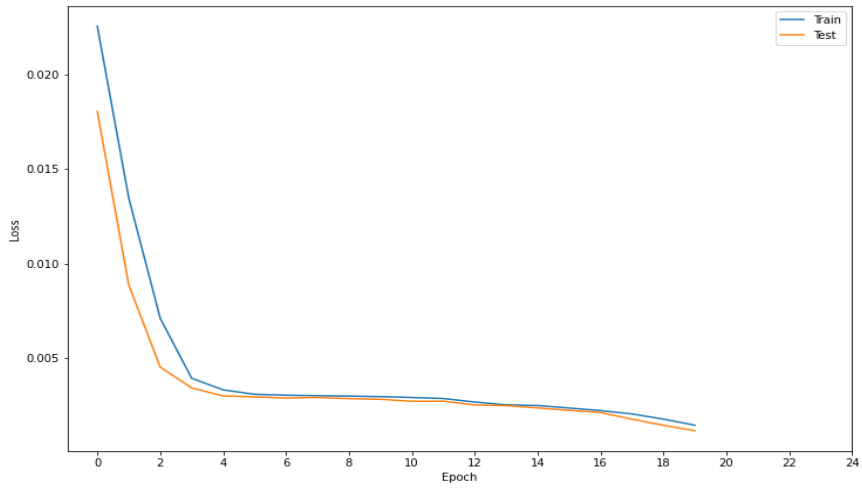
conv2d_transpose_8 (Conv2DT ranspose) (None, 128, 128, 32) 11552

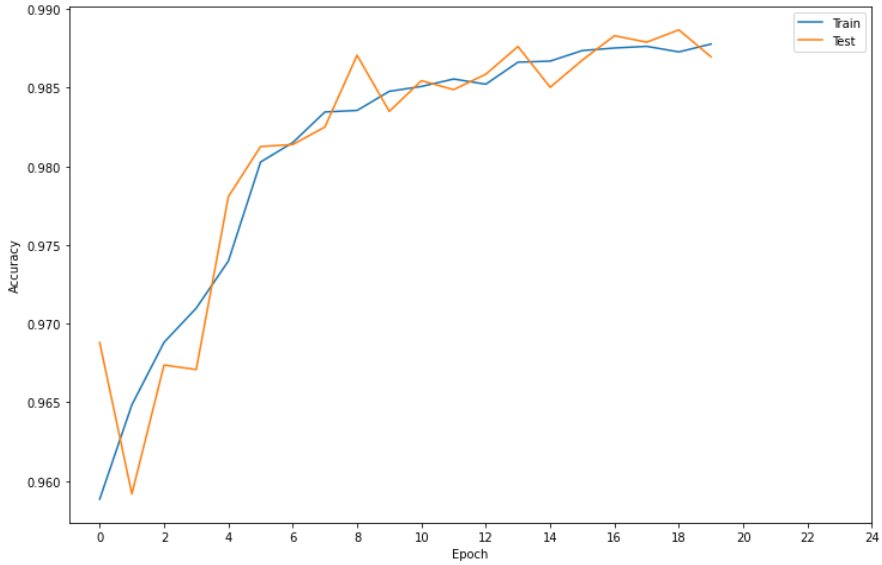
dropout_14 (Dropout) (None, 128, 128, 32) 0

decoder_output (Conv2DTrans pose) (None, 128, 128, 3) 867

===
Total params: 1,322,171
Trainable params: 1,322,171
Non-trainable params: 0
_____

Performance :





Results without sharpening The Image :