**Model Training for 81 Landmarks:**

**Data Collection:**

* All images must be in .jpg format
* Here we have used Haarcascade\_frontalface to crop the images and save them.
* Model Training is done on images having only cropped faces from original images.
* Images must contain only one face in one image.

If an image contains more than one face then landmarks will be detected more than once and the list will be appended in the same image name.

For ex: If image has 2 faces, then landmarks will be (81\*2=162) \* 2.

* When a face is not detected in the Image it will append a null row into the csv so we need to remove the null rows from csv programmatically or manually.
* Number of images in the folder and in csv must be the same.

**Model Training:**

* Model must have activation “Linear” in the last layer as we are performing the Regression Model
* Function Generate\_images( ) will take images randomly from the folder and append them.
* Function Generate\_keypoints( ) will randomly append the normalised landmarks from csv to the list.
* Then last neuron of dense layer will be twice of landmark points as they are in (x,y) form
* Ex: 26 landmarks model = 26\*2 = 52
  + model.add(Dense(52))
* 81 landmarks model = 81\*2 = 162
  + model.add(Dense(162))
* Normal CNN model (32,32,64,64,96,96,128,128,512,512) and
* BottleNeck (32,32,64,64,96,96,128,128,96,96,64,64) gives almost the same output when trained over 500 epochs.

**Model Testing:**

* We need to take the original image , crop the face from the image using a haar cascade then normalise and do pre-processing then predict the model output on the image.