**Introduction:**

To Train a model using Deep Learning for our face recognition project we had to use Multi-task Cascaded Convolutional Networks.

MTCNN is a framework developed as a solution for both face detection and face alignment. The process consists of three stages of convolutional networks that are able to recognize faces and landmark location such as eyes, nose, and mouth

The network uses a cascade structure with three networks

1. first the image is rescaled to a range of different sizes (called an image pyramid)

then the first model (Proposal Network or P-Net) proposes candidate facial regions by making bounding boxes.

1. the second model (Refine Network or R-Net) filters the bounding boxes ( non-maximum suppression (NMS) is used to filter the candidate bounding boxes proposed by the first-stage P-Net prior to providing them to the second stage R-Net model. )
2. the third model (Output Network or O-Net) proposes facial landmarks.

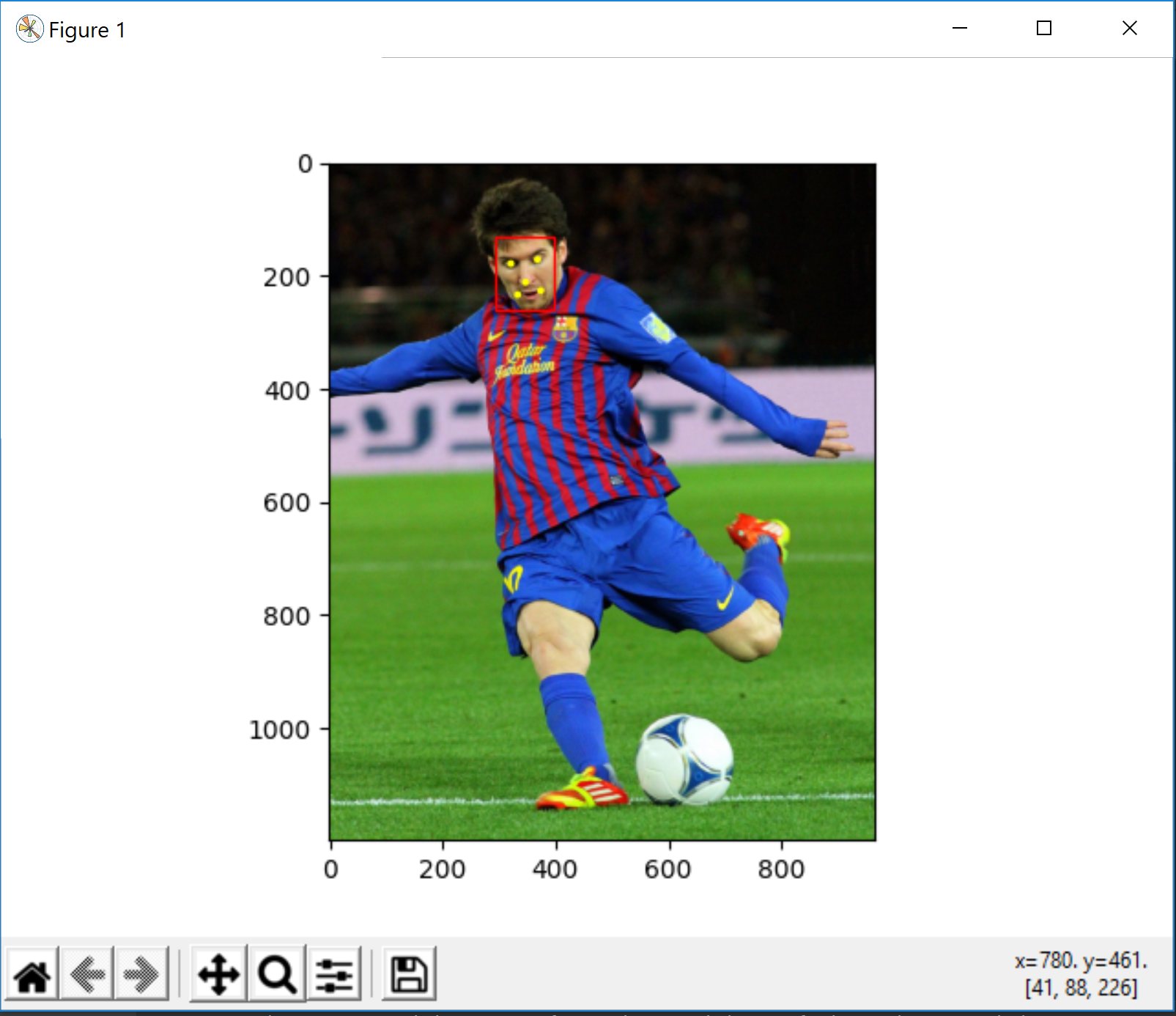
In order to train a model we used Tensor flow and keras library. It creates an image classifier using a **keras.Sequential()** model, and loads data using **preprocessing.image\_dataset\_from\_directory**(), for compiling the model **model.compile()** and for testing/training it **model .fit()**.

**The Objectives was to** :

1. Examine and understand data
2. Build an input pipeline
3. Build the model
4. Train the model
5. Test the model
6. Improve the model and repeat the process.

**Issues:**The issues I faced during training the model through deep learning were quite a few.

* I couldn’t properly finish the task, couldn’t get the resulting output showing face classification/recognition. Because:
* I did not had the version that could actually install tensor flow and run my program.
* Had an error, Which I couldn’t resolve.
* Since program wasn’t running successfully couldn’t find the accuracy aswell.
* Used a predefined model to recognize the face using mtcnn.

OUTPUT using predefined model using MTCNN libraries:  


CONCLUSION

With PCA we were able to properly test and train the data set in order to recognize the face by extracting features through eigen vectors and forming average face. In addition to test and train the models, the logistic regression library also helped us determine the accuracy of the output.  
Whereas, The facial recognition through mtcnn we were able to use predefined models to test the outputs. But we weren’t able to train the model before hand using tensorflow and keras library due to some issues of the version of the python script that I was using.