Deep Glance Classification
Computer Vision: Spring Semester 2018

Shreyas Fadnavis, Khusaal Giri, Arun Nekkalapuddi, Aditya Singh Kushwah

30th April 2018

Introduction

Emotional engagement is crucial in making effective online content, but the current technologies do not give an accurate way to measure this. We therefore propose a novel way of accurately understanding how people really feel. People will view the content on their own and our system will track their emotions in real time.

Our system makes use of the *Facial Action Coding System* to measure people's emotions as they watch video content and the relevance of the quality of that content. Based on this fine-grained analysis of the features and emotions captured, we make use of a *recommender system* to generate feedback for the people making the content.

Description/ Methodology

From [1], basic emotion do not vary so much in their expressions, but they do vary in the context of their intensities. For example:

When a person smiles in a video, it is not so useful to just detect the smile, we also need to understand the duration and the intensity with which they smiled.

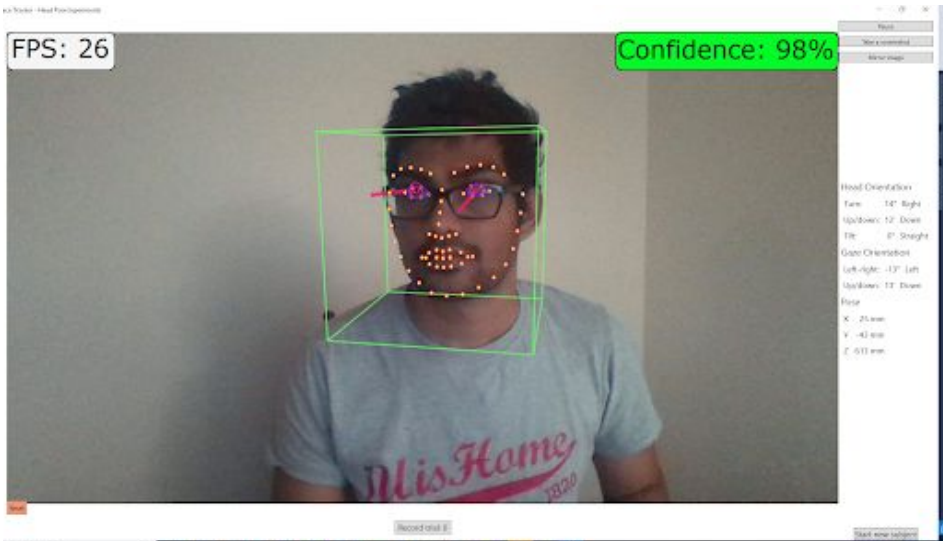
This requires us to give a context to understanding the emotion with respect to the temporal mapping of the expression sequences of the videos captured. Every single frame in our dataset is annotated with landmark points which are then bucketed into high-level descriptor buckets of emotions.

Facial expressions evolve dynamically over time so the frames are always viewed and labelled within a sequence making way for temporal analysis of this time series data.

This entire process is broken down into the following:

Face Detection: For each frame our framework estimates the position of the face with a confidence value for that frame.

Facial Keypoint Alignment: Important landmark points are tracked around the eyes, nose, mouth, etc.



To do so, our framework makes use of the Conditional Local Neural Fields (CLNF) for facial landmark detection and tracking. The two main components of CLNF are:

Point Distribution Model (PDM) which captures landmark shape variations.

(Training Datasets: LFPW and Helen training sets with correct and randomly offset landmark locations.)

- We use a simple three layer convolutional neural network (CNN) that given a face aligned using a piecewise affine warp is trained to predict the expected

landmark detection error. If the validation step fails when tracking a face in a video, we know that our model needs to be reset. This resulted in a model with 34 non-rigid and 6 rigid shape parameters.

Our model is able to **extract head pose (translation and orientation)** information in addition to facial landmark detection. We are able to do this, as CLNF internally uses a 3D representation of facial landmarks and projects them to the image using orthographic camera projection.

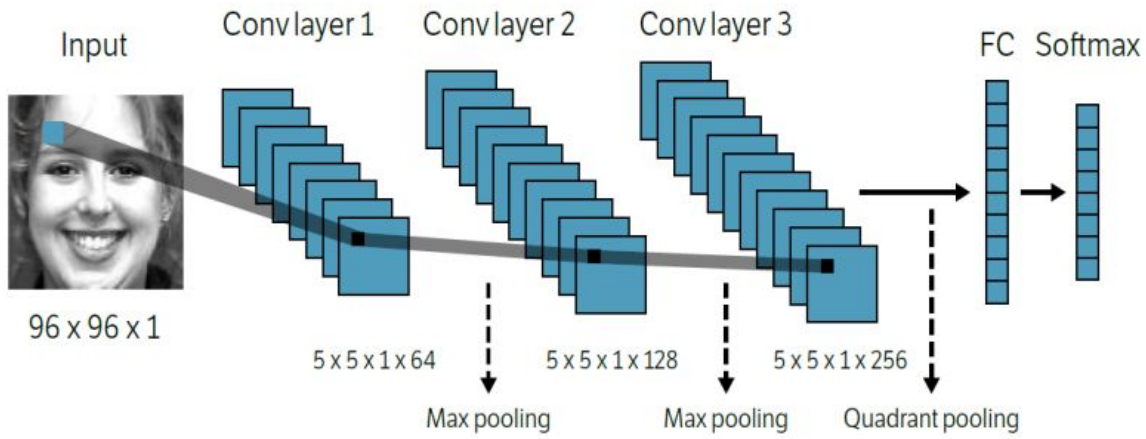
Once the **location of the eye and the pupil are detected using our CLNF model** we use that information to compute the eye gaze vector individually for each eye. We fire a ray from the camera origin through the center of the pupil in the image plane and compute its intersection with the eyeball sphere. This gives us the pupil location in 3D camera coordinates. The vector from the 3D eyeball center to the pupil location is our estimated gaze vector. This is a fast and accurate method for person independent eye-gaze estimation in webcam images.

Patch experts which capture local appearance variations of each landmark.

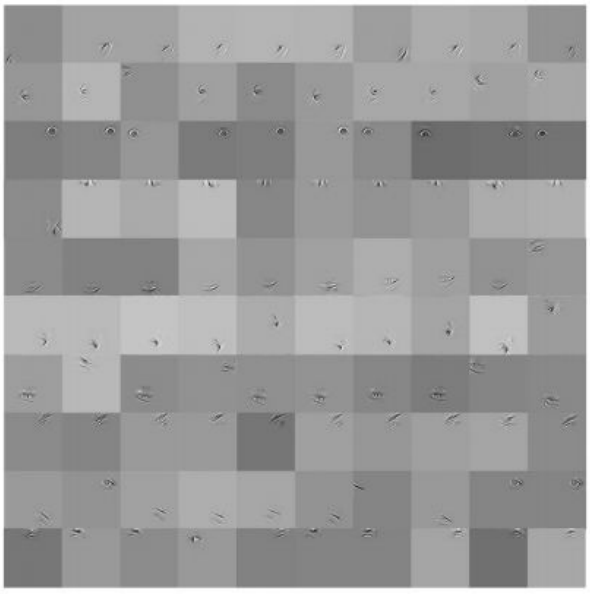
For training the CLNF patch experts we used: Multi-PIE, LFPW and Helen training sets. We trained a separate set of patch experts for seven views and four scales (leading to 28 sets in total). Having multi-scale patch experts allows us to be accurate both on lower and higher resolution face images. We found optimal results are achieved when the face is at least 100px across. Training on different views allows us to track faces with out of plane motion and to model self-occlusion caused by head rotation. To initialize our CLNF model we use the face detector found in the dlib library. The system learned a simple linear mapping from the bounding box provided by dlib detector to the one surrounding the 68 facial landmarks. When tracking landmarks in videos we initialize the CLNF model based on landmark detections in previous frame. If our CNN validation module reports that tracking failed we reinitialize the model using the dlib face detector.

Tracking a face over a long period of time may lead to drift or the person may leave the scene. In order to deal with this, we employ a face validation step.

We use a simple three layer convolutional neural network (CNN) that given a face aligned using a piecewise affine warp is trained to predict the expected landmark detection error.



Correlation Matrix of Latent Representations of our Features



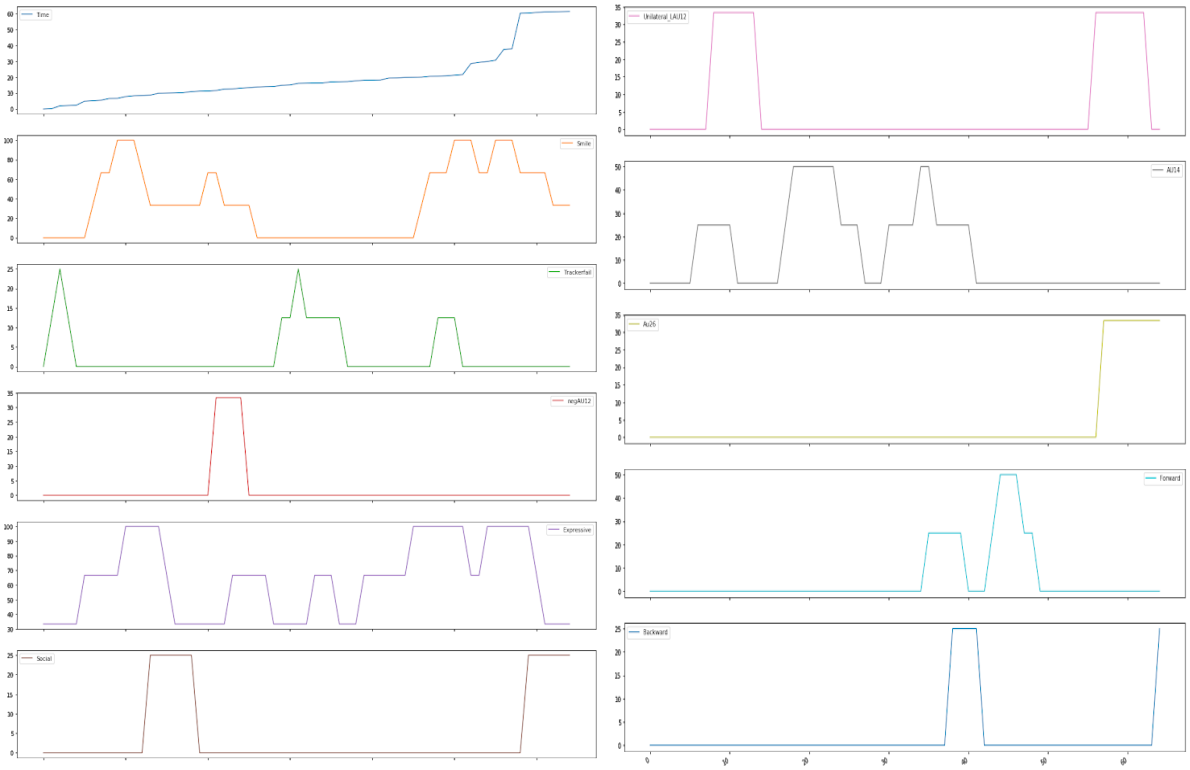
Facial Action Coding System

The 27 facial Action Units are broken down into upper and lower facial movements. The upper facial Action Units include brow raising (Action Units 1 and 2 for inner and outer edges of the brow respectively), brow lowering (4), eyelid raising (5) and cheek raising (6). Lower facial actions are more complex and include vertical, horizontal, oblique and orbital Action Units. An example of a vertical Action Unit would be number 15 which pulls the corners of the lips straight down, while an oblique Action Unit would be Action Unit 12 which pulls the lip corners up and out.

Most facial expressions include a combination of upper and lower Action Units, and some combinations are more common than others. Head and eye movement codes are broader and less commonly used but can provide important emotional information about what's happening on the face. They include head tilting, turning and general eye direction. In addition to the occurrence of each Action Unit, coders can also rate the intensity. Action Unit intensity is rated on a scale of evidence from A (trace levels of visibility) to E (maximal). Trace intensity means that the Action Unit is barely visible on the face, and intensity progresses along the scale to slight, marked, pronounced, severe, extreme and finally maximum. The requirements for reaching each level of intensity vary across Action Units.

Automated coding of facial expressions and their corresponding intensities uses a combination of computer vision and pattern recognition techniques. Manual coding reliability is generally higher for posed than genuine facial expressions, but even then, many studies have reported a mean agreement with human coders of over 96%.

Temporal analysis of each Feature



For each video now, we have a set of features which are being activated for the entire video. In order to generate a feedback, we want to summarize the activations of these features in order to feed them into our recommendation system.

We therefore need a more reliable and sophisticated way to do so because we do not want to count the drops in the confidence of capturing an emotion as a break in the continuity of that emotion in the video.

We have made use of the Natural Breaks algorithm with Jenks Optimization to capture and summarize such emotional activations in the functions.

Jenks Natural Breaks Optimization

This is a data clustering method designed to determine the best arrangement of values into different classes. This is done by seeking to minimize each class's

average deviation from the class mean, while maximizing each class's deviation from the means of the other groups. In other words, the method seeks to reduce the variance within classes and maximize the variance between classes. The Jenks optimization method is directly related to Otsu's Method and Fisher's Quadratic Discriminant Analysis(QDA).

The method requires an iterative process. That is, calculations must be repeated using different breaks in the dataset to determine which set of breaks has the smallest in-class variance. The process is started by dividing the ordered data into groups. Initial group divisions can be arbitrary. There are four steps that must be repeated:

1. Calculate the sum of squared deviations between classes (SDBC).
2. Calculate the sum of squared deviations from the array mean (SDAM).
3. Subtract the SDBC from the SDAM (SDAM-SDBC). This equals the sum of the squared deviations from the class means (SDCM).
4. After inspecting each of the SDBC, a decision is made to move one unit from the class with the largest SDBC toward the class with the lowest SDBC.

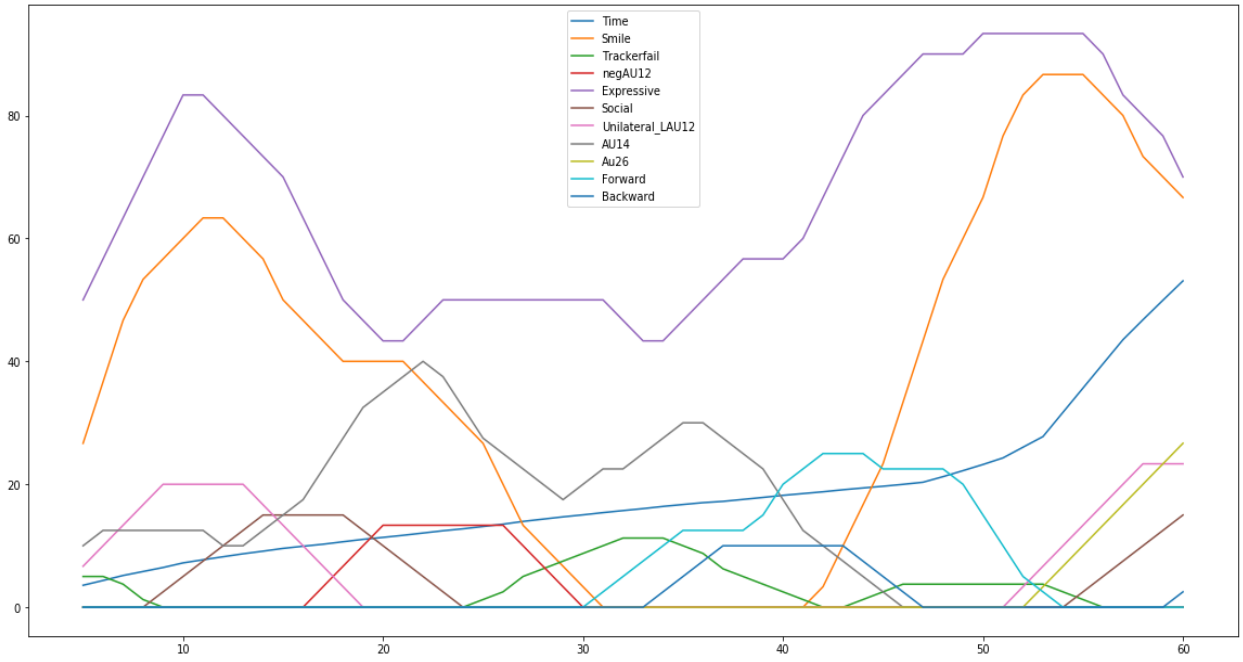
New class deviations are then calculated, and the process is repeated until the sum of the within class deviations reaches a minimal value.

Alternatively, all break combinations may be examined, SDCM calculated for each combination, and the combination with the lowest SDCM selected. Since all break combinations are examined, this guarantees that the one with the lowest SDCM is found.

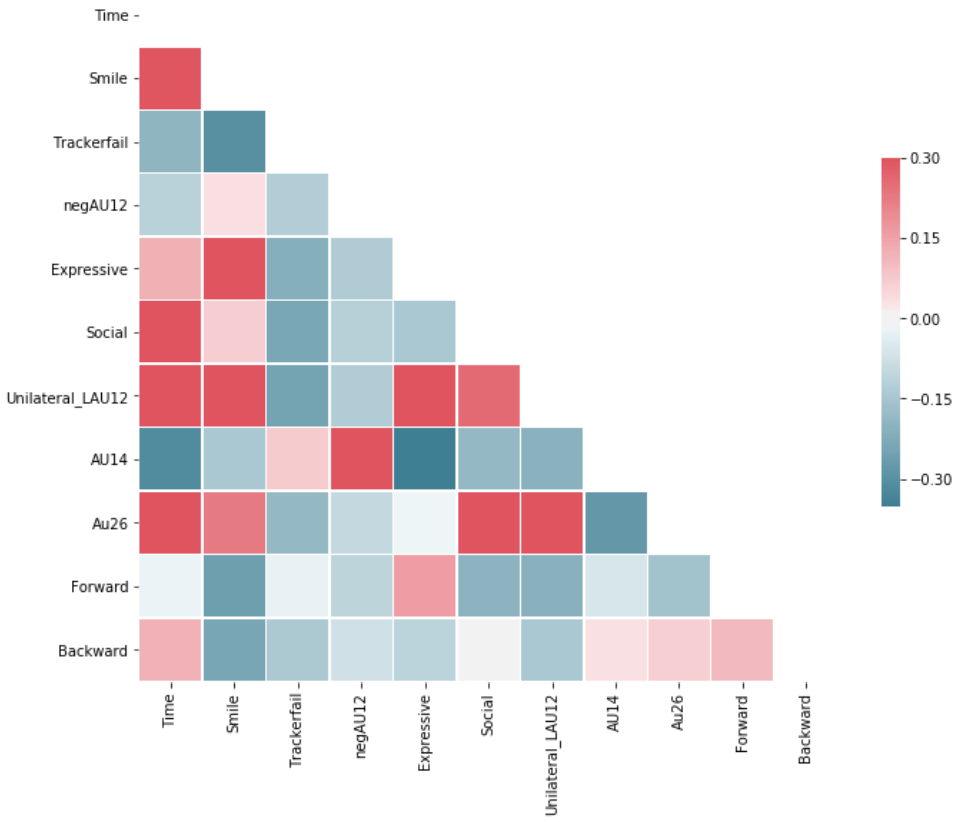
Finally, the Goodness of Variance Fit (GVF) is calculated. GVF is defined as (SDAM - SDCM) / SDAM. The GVF ranges from 0 (worst fit) to 1 (perfect fit).

Sanity Checks: Validating our Analysis

In order to make sure that we are not feeding incorrect summaries that misrepresent the temporal sequence of the emotions, we evaluate our results by doing a Moving Averages based Time Series analysis to understand the corresponding feature changes resulting in emotion sequences of the video.



We also plotted a correlation tests and analysis to make sense of these activated emotions from both our training data and testing data. An example plot for the above video is as follows:

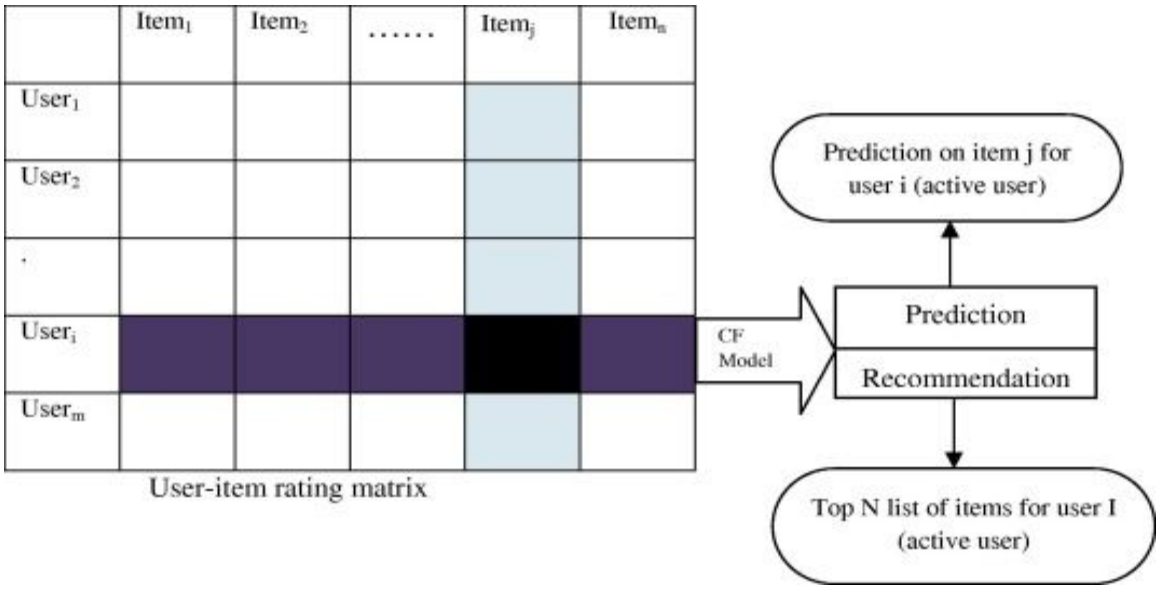


Collaborative Filtering algorithm

The main goal of this algorithm is to predict one of the seven emotions of humans(Happy,Sadness,Surprise,Fear,Anger,Disgust & Contempt) while watching the video content. Our main purpose was to collaborate all the videos (Training Data) into the categories which correspond to the seven basic human emotions. For e.g: If the action units (AUs) of Video 1 are almost similar to that of Video 2;then Video 1 & Video 2 should fall into the same category of emotion.This was the basic intuition behind using the Collaborative Filtering algorithm.

In this algorithm,we input the action units of all the videos & then,try to fit a model in which the weights represent the true loading of the action units with the contents of the video.We use a Gradient descent approach for minimizing the error rate of our

model. Once the weights are known,we can find the categorisation of the new video by computing the coefficient vector whose size is equal to the number of human emotions considered in our model; where each value represents the likelihood of the video corresponding to a particular emotion. This coefficient vector is computed by taking a dot product of the action units of the Video with the weight vector.The learning rate used is 0.004 & the regularization constant is set to 10 with 150 number of epochs for our model.



Random Forest Classifier

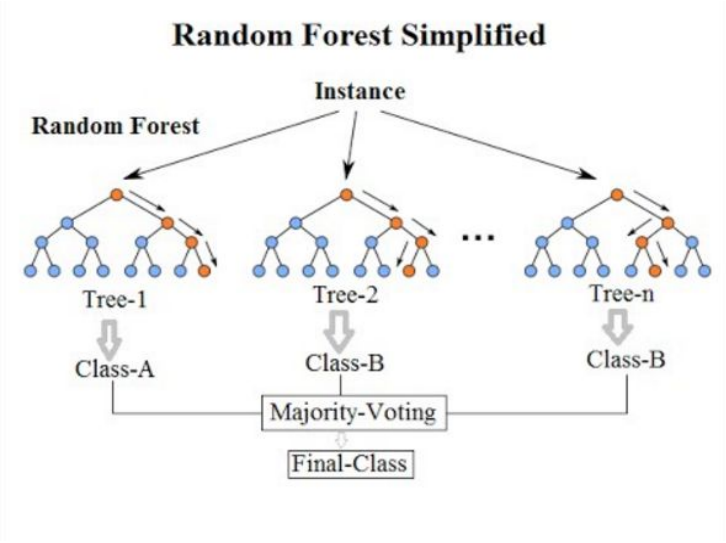
The main goal of this algorithm is to generate feedback of the user based on their emotions while watching the contents of the video. This feedback is generated by answering the following 3 questions(with all the possible response):

1. Did the person like the video ?
    - 2 =”Heck ya! I loved it.”
    - 1 =’’Meh! It was ok.”
    - 0 =”Na….not my thing.”

- -1= No response

2. Has the person seen the video before ?
    - 2 =”Yes,many times”
    - 1 =”Once or twice”
    - 0 =”Nope,first time”
    - -1= No response
3. Would the person watch the video again ?
    - 2 =”You bet!”
    - 1 =”Maybe,if it came on TV”
    - 0 =”Ugh.Are you kidding?”
    - -1 = No response

We have fitted a multi-label multi predictor Random forest classifier with 3 predictor variables,each specifying the output of the above mentioned 3 questions. Since,it was a multi-response multiple category classification problem, we decided to choose random forest classifier for better model fitting.The output of the Random forest classifier will be the answers to the above mentioned questions,which will be a basic summary of the feedback of the user for a particular video content. The number of trees in the forest is set to 4,with maximum features set to ‘auto’(i.e, square root of the total number of features in the model).



References

[1] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I., & Ave, F. (2010). The Extended Cohn-Kanade Dataset (CK +): A complete dataset for action unit and emotion-specified expression, (July), 94–101.

[2] OpenFace: an open source facial behavior analysis toolkit Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency, in IEEE Winter Conference on Applications of Computer Vision , 2016

[3] Paine, T. Le, & Huang, T. S. Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition, 19–27.

[4] Mcduff, D., Kaliouby, R. El, Senechal, T., Amr, M., Cohn, J. F., & Picard, R. Affectiva-MIT Facial Expression Dataset (AM-FED): Naturalistic and Spontaneous Facial Expressions Collected In-the-Wild.

[5] D. Baumann, M. Mahmoud, P. Robinson, E. Dias and L. Skrypchuk, " Multimodal classification of driver glance, " 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), San Antonio, TX, 2017, pp. 389-394.

[6] V. Tümen, Ö. F. Söylemez and B. Ergen, " Facial emotion recognition on a dataset using convolutional neural network ," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, 2017, pp. 1-5.

[7] J. H. L. Hansen, C. Busso, Y. Zheng and A. Sathyanarayana, " Driver Modeling for Detection and Assessment of Driver Distraction: Examples from the UTDrive Test Bed ," in IEEE Signal Processing Magazine , vol. 34, no. 4, pp. 130-142, July 2017.

[8] “ Dual-Glance Model for Deciphering Social Relationships ” Junnan Li, Yongkang Wong, Qi Zhao, Mohan S. Kankanhalli.

[9] “ Real Time Head Pose Estimation from Consumer Depth Cameras ” Fanelli, Gabriele and Weise, Thibaut and Gall, Juergen and Gool, Luc Van Proceedings of the 33rd International Conference on Pattern Recognition

[10] Facial landmark detection and tracking, Constrained Local Neural Fields for robust facial landmark detection in the wild Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. in IEEE Int. Conference on Computer Vision Workshops, 300 Faces in-the-Wild Challenge, 2013.

[11] Eye gaze tracking, Rendering of Eyes for Eye-Shape Registration and Gaze Estimation Erroll Wood, Tadas Baltrušaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling in IEEE International. Conference on Computer Vision (ICCV), 2015

[12] Facial Action Unit detection, Cross-dataset learning and person-specific normalisation for automatic Action Unit detection Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson in Facial Expression Recognition and Analysis Challenge, IEEE International Conference on Automatic Face and Gesture Recognition, 2015