

# DeepFake Detection Program v1

**Hypothesis:** People have unique combinations of facial features that will typically move together. For example, Person A's eyes lift up when they speak, and Person B wrinkles their nose when they blink. These combinations of two can be used to create probability maps for each person, and can be used to determine if two profiles share similar pairing probabilities.

## Loading Data into Program

1. Have "raw" CSV files saved into local directory (by "raw" I mean straight from OpenFace)
2. In program, declare a List of input files called "rawFiles" and a List of output files called "processedFiles"
3. Load the raw CSV files into "rawFiles"

## Processing Raw CSV Files

1. Have a function capable of stripping the CSV file into the important data we need
  - delete unconfident/unsuccessful rows
  - there are ~709 action units in the raw CSV, so we need to determine the most important ones. Ideally <100
2. Send raw CSV files into the function and store processed files in "processedFiles"

## Object-Oriented Facial Landmark Representation

1. Create Object class called "Landmark"
2. Landmark has member variables:
  - public float threshold: for holding the threshold of on/off (explained later)
  - public bool[] active: for keeping track of when this landmark is active
    - each index represents a frame, so it stores whether the landmark is on/off for each frame
  - these bool values are determined by comparing the landmark's float values to the threshold
3. Landmark has member functions:
  - Landmark(): default constructor that assigns threshold to -1 and sets active array size to 0.
  - Landmark(float threshold): takes in a threshold value and assigns it, sets active array to size 0.
  - Landmark(float threshold, int size): takes in and assigns threshold and sets active array size.
  - public void CalcActive(float[] vals): compares all floats in vals array and

compares it to the threshold, stores resulting bool in active array

### **Computations and Profile Creations**

1. Declare Landmark array and manually initialize all Landmark objects with threshold and size
2. create a square 2D matrix of floats with dimensions being the amount of facial landmarks (if 100 facial landmarks used, matrix is 100x100)
3. Iterate over the matrix, but only on the upper triangular to avoid duplication (all values below and including the diagonal are unused)
4. Compare the "on/off" times of both corresponding Landmarks, and store the probability of how likely they will both be on and both be off at the same time (XNOR comparison).
5. Do this process for multiple videos of a person, and store each matrix as well as the average of them all.
6. Make comparison against this matrix with bad deepfakes and determine a similarity value