**CYPHER CAM**

* Windows/Linux/Mac OS any version, hence it can run on any platform.
* Python3, it need python to be installed in your system to run this successfully.
* Packages in python -
  + openCV
  + skimage
  + numpy
  + tkinter

**1 – Detecting faces in the frames**

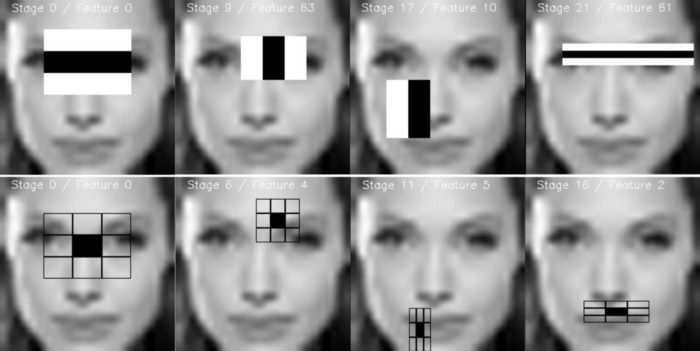
This is done via **Haarcascade** classifiers which are again in-built in openCV module of python.

Cascade classifier, or namely cascade of boosted classifiers working with haar-like features, is a special case of ensemble learning, called boosting. It typically relies on **[Adaboost](https://www.google.com/url?q=https%3A%2F%2Fmaelfabien.github.io%2Fmachinelearning%2Fadaboost&sa=D&sntz=1&usg=AFQjCNFeQbwPdev7PoN6X7Mhb9A-3Bh0iQ)** classifiers (and other models such as Real Adaboost, Gentle Adaboost or Logitboost).

## There are some common features that we find on most common human faces :

* a dark eye region compared to upper-cheeks
* a bright nose bridge region compared to the eyes
* some specific location of eyes, mouth, nose…

Haar features are similar to these convolution kernels which are used to detect the presence of that feature in the given image.



**2 – Using LBPH for face recognition**

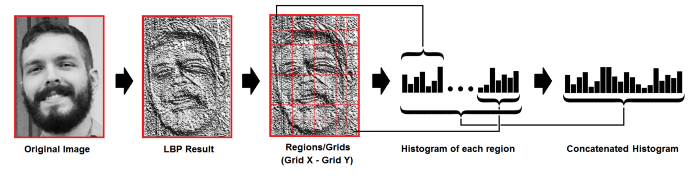
So now we have detected for faces in the frame and this is the time to identify it and check if it is in the dataset which we’ve used to train our lbph model.

(Local Binary Pattern Histogram)

The LBPH uses 4 parameters:

* Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
* Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
* Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
* Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.





**3 – Detect for Noises in the frame**

This feature is used to find the noises in the frames well this is something you would find in most of the cctv’s but in this module we’ll see how it works.

Talking in simple way all the frames are continously analyzed and checked for noises. Noise in checked in the consecutive frames. Simply we do the absolute difference between two frames and in this way the difference of two images are analyzed and Contours (boundaries of the motion are detected) and if there are no boundries then no motion and if there is any there is motion.

**4 – Visitors in room detection**

This is the feature which can detect if someone has entered in the room or gone out.

So it works using following steps:

1 – It first detect for noises in the frame.

2 – Then if any motion happen it find from which side does that happen either left or right.

3 – Last if checks if motion from left ended to right then its will detect it as entered and capture the frame.

Or vise-versa.

So there is not complex mathematics going on around in this specific feature.

**5 - Anti-Theft Alarm.**

In this feature every frame is analysed and if any object is missing from its original position then the immediately in the next frame it captures the photo of the person who is in the frame and makes an alarming sound so that user gets to know something is wrong.

After making an alarm it shows what exactly missing from the original position by compering two frames (before object misplace and after object misplace) and making a notation around that object.

By implementing this feature we can find who is the culprit and which object is stolen.

**Why Waterfall?**

For this model we have used **waterfall model,** since it was not huge project at all.

Reasons behind choosing waterfall model -

1. Good for projects.
2. Easy to follow.
3. Well Back tracking for projects.
4. Well time managed.

**References**

For making this project we have used so many websites and papers and you-tube tutorials all are below specified.

* [waterfall model geeksforgeeks](https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/)
* [Structural Similarity from medium](https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e)
* [face detection](https://medium.com/geeky-bawa/face-identification-using-haar-cascade-classifier-af3468a44814)
* [LBPH algorithm](https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b)
* [openCV](https://www.youtube.com/channel/UCx1_WfGX9D9rmsJNBM5qsMA)
* [tech with tim](https://www.youtube.com/channel/UC4JX40jDee_tINbkjycV4Sg)
* W3school.com

Also made use of so many other youtube channels and google and stack overflow to solve the errors.

Also we used Official python documentations to know basics about python