Recognition of substance abuse tendencies using facial features through Image Processing, CBIR and Machine Learning

Submitted for Technical Answers For Real World Problems in B.Tech Engineering Computer Science (CSE)

**By:**

**Aishwarya Jayachandran (17BCE1297)**

**Akanksha Chawla (17BCE1286)**

**Ram Chander (17BCE1038)**

**Slot: TDD1**

**Name of faculty:** NITHYA DARISINI P

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**



# Problem statement:

Predicting the alcoholic or drug abuse tendencies of people based on facial features.

# Introduction:

In this project, we will develop a mechanism to search through a dataset of images with and classify them on the basis of their facial features and analyze the possibility of addiction towards alcohol or drugs. According to various studies conducted by psychologists, substance abusers tend to have increased flushing, rough patches on their cheeks, spider veins and red discoloration in their eyes and so on. Research also shows abusers experience varied forms of skin deterioration as well as damage in their teeth, baldness, and unusual facial hair patterns. This project can sort through a large colossal dataset of pictures and classify them accurately based on a picture of their face. The project can be broke down into two parts. The first part consists of studying the pictures in given dataset and extracting the attributes necessary to do then predict the outcome. This is done with the help of Image processing and CBIR concepts. The second part consists of applying machine learning in order to predict their substance abuse tendencies as accurately as possible

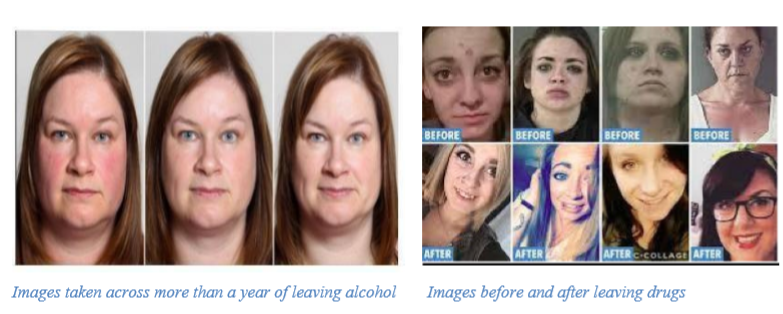
# Motivation:

Addiction is a serious disease. According to WHO, Alcohol abuse results in 3.3 million deaths each year and less than half the population(38.3%) of the world only consumes alcohol which means that every person drinks almost 17 liters of alcohol per annum. 31 million people have high level drug disorders. We want to develop a method to recognize drug or alcohol addiction and to make it as easily accessible as possible.

# Implementation Details:

## Description about the dataset:

For the purpose of this project, a dataset was required that consists of pictures of people with and without substance abuse tendencies. Since this problem has never been handled before, there is no dataset existing on the internet that fulfilled our requirements, as of now. Therefore, a dataset was created from scratch by combining an existing dataset of people with no substance abuse tendencies with pictures found on online blogs made to encourage abusers to quit with uploads of ‘before’ and ‘after’ images of alcohol and drug addicts.





## Algorithm and explanation:

A dataset consisting of about 730 pictures is created, consisting of alcoholics, drug addicts and non-substance abusers. The first step is to go through these images and extract all the six attributes one by one, for all three datasets (alcoholics, drug addicts, non-substance abusers) and create excel sheets that provide numerical values for the machine learning aspect of this project.

## Image processing and CBIR:

After face cropping the image, to avoid cluttering and miscalculation of unnecessary values, using the Viola Jones algorithm the six attributes extracted are:

1. Squinting of eye (if eyes are close, drowsy or not)

2. Number of distinctive spots on a face (high amounts present in drug addicts)

3. Amount of excessive lines/wrinkles around a person’s eyes and face

4. Presence of redness in eyes

5. Amount of excessive redness on a face expressed in percentage (present in high amounts in alcoholics and low in the other two)

6. Degree of paleness on a face expressed in percentage (used for identifying both alcohol and drug addicts)

## The Viola Jones algorithm

Detection happens inside a detection window. A minimum and maximum window size is chosen, and for each size a sliding step size is chosen. Then the detection window is moved across the image as follows: 1. Set the minimum window size, and sliding step corresponding to that size.

2. For the chosen window size, slide the window vertically and horizontally with the same step. At each step, a set of N face recognition filters is applied. If one filter gives a positive answer, the face is detected in the current widow.

3. If the size of the window is the maximum size stop the procedure. Otherwise increase the size of the window and corresponding sliding step to the next chosen size and go to the step 2.

Each face recognition filter (from the set of N filters) contains a set of cascade connected classifiers. Each classifier looks at a rectangular subset of the detection window and determines if it looks like a face. If it does, the next classifier is applied. If all classifiers give a positive answer, the filter gives a positive answer and the face is recognized. Otherwise the next filter in the set of N filters is run.

Each classifier is composed of Haar feature extractors (weak classifiers). Each Haar feature is the weighted sum of 2-D integrals of small rectangular areas attached to each other. The weights may take values ±1. Fig.2 shows examples of Haar features relative to the enclosing detection window. Gray areas have a positive weight and white areas have a negative weight. Haar feature extractors are scaled with respect to the detection window size.

Method used to obtain attributes, explained below:

## To obtain if person’s eyes are squinting:

## The picture needs to be edited in order to ensure that the dimensions of the picture are optimal to the code. The Viola Jones algorithm is used to crop the person’s eyes and Hough’s transform is used to detect if the pupil is as visible as it would be in a person who is not drowsy.

## To obtain the number of spots on the persons face (used for identifying drug addicts):

The Viola Jones algorithm is used to crop the person’s face and this new image is used. Hough’s transform is used in order to detect all the possible circles in the image and these are further filtered on the basis of colour.

## To obtain the amount of wrinkles on the persons face and lines around their eyes:

The Viola Jones algorithm is used to crop the persons face and this new image is used. This image is then converted to HSI as this gets rid of all unnecessary colour details. In order to get the H, S, V thresholds for segmentation based on the ROI (the wrinkles that show up in deep blue colour). The colour thresholding toolbox is used to adjust the values until a suitable mask is obtained. This mask is then used to obtain the average value of the H, S, and V and assuming a +- 0.06 per value, the thresholding is done.

To detect presence of redness in eyes:

The eyes are cropped out of the image using the Viola-Jones algorithm, the red pigmentation of the image is separated out. A median filter is made which is then subtracted from the original image which is then used to detect the red parts of the eye.

Machine Learning

After extracting numerical values for features in question, machine learning aspect of this project comes to play. Several machine learning algorithms were tried and tested until the highest accuracy of prediction was obtained. The dataset is split into training and testing data and standardization along with normalization of data has been carried out to prevent erratic predictions, which is often what is observed when individual features do not look represent a standard normally distributed data.

# Why and how algorithms have been executed:

Researches show alcohol and drug addiction in a person is more often than not a spectrum. While in some cases, the person’s addiction is already established; in many others it is important to predict the addiction along the path of its development. Since this project requires the algorithm to take in all the information regarding these attributes and produce the probability of addiction related symptoms in a person. Logistic Regression has been employed. In addition to these prospects, Linear Regression algorithms tend to be highly efficient and easy to train. Apart from Linear Regression, SVM algorithm was also implemented as it works well with non linear functions and has no limit in terms of dimensionality of the function that applies to the dataset which is the scenario in this case. They can also be a useful tool for insolvency analysis, in the case of non-regularity in the data, for example when the data are not regularly distributed or have an unknown distribution. SVMs also tend to show more insensitivity towards outliers than other algorithms which proved to be extremely helpful on this case due to the lack of availability of a uniform dataset.

**Solvers selected:**

When the logistic regression function was called, ‘solver’ was set to liblinear. LIBLINEAR is a linear classifier for data with millions of instances and features. It supports:

* L2-regularized classifiers L2-loss linear SVM, L1-loss linear SVM, and logistic regression (LR)
* L1-regularized classifiers (after version 1.4) L2-loss linear SVM and logistic regression (LR)
* L2-regularized support vector regression (after version 1.9) L2-loss linear SVR and L1-loss linear SVR.

**Main features of LIBLINEAR include**

* Same data format as LIBSVM, our general-purpose SVM solver, and also similar usage
* Multi-class classification: one-vs-the rest, is the classification we used for our data, as we needed multi-classes for our data
* Automatic parameter selection
* Probability estimates (logistic regression only)
* Weights for unbalanced data: Class weights were declared separately and passed as a parameter to the algorithm

# CHALLENGES FACED:

Class imbalanced data

Out of the total 731 images available in the dataset, 566 are of those who are non substance abusers, 100 are of alcoholics and drug addicts amount to 65 pictures. In order to combat class imbalance in the three classes in given dataset, weights given to each of the three classes have been adjusted and oversampling using SMOTE has been implemented.

When there is a gross imbalance between the classes of a dataset, classes with less data are often considered as outliers or noise by the machine learning algorithm and are ignored. In this case, before any solutions were applied, alcohol and drug addict classes were considered as outliers. So instead of assigning the default weights, i.e equal weights to all three classes, these weights are used to balance out the class disparities.

Over-Sampling increases the number of instances in minority classes by randomly replicating them in order to present a higher representation of the minority class in the sample. SMOTE works by creating synthetic samples from the minor class instead of creating copies as machine learning algorithms have trouble learning when class disparities exist. The argument ‘random-state’ is used as seed for the random number generator and it centered the randomization of our algorithm.

## Noise

Since dataset used for alcohol and drug addicts does not consist of professionally taken photographs and includes selfies, these are very low quality photographs. Images have issues such as poor lighting, graininess, uneven saturation and so on. These also act as a hindrance in the prediction process as values taken from these images do not translate well numerically.

## Accuracy v/s Recall value

Once class imbalance was combated, it was essential to understand the significance of both terms with respect to our problem statement. Accuracy as a metric for this model would have been a poor choice mainly because of the class imbalance present in the dataset. A high accuracy in this case meant that because of the larger size of the non substance abusers class, they happened to have a higher chance at being classified more correctly. Recall actually calculates number of actual positives model captures through labeling it as positive (true positive). Applying the same understanding, we know that recall shall be the model metric we use to select our best model when there is a high cost associated with false negative, which in this case computes percentage of correctly classified alcohol or drug. As the problem statement of this project specifies, our end goal is to predict alcohol and drug using tendencies in an individual and though it would be incorrect to label a non substance abuser as an alcohol or drug addict, it would be even more catastrophic to label an alcohol or drug addict a non substance abuser.

# Experimental Results and Discussion:

## Outputs from MATLab:

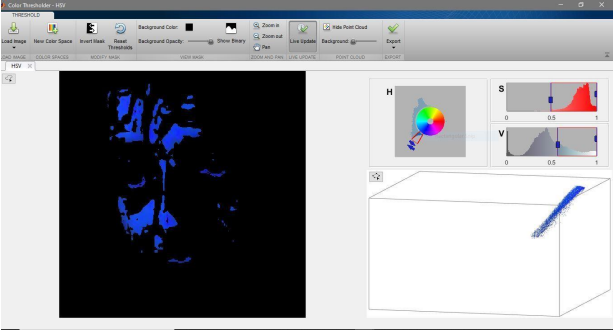
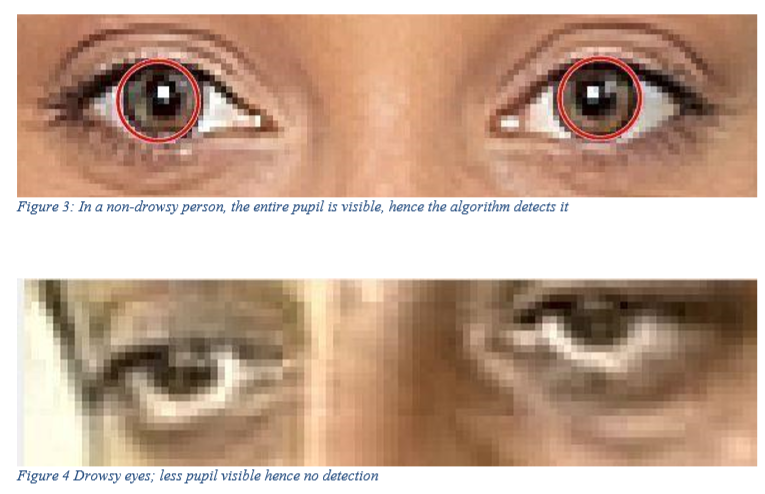


Figure 1: Wrinkles are highlighted by using MATLAB tool to get precise HSV values

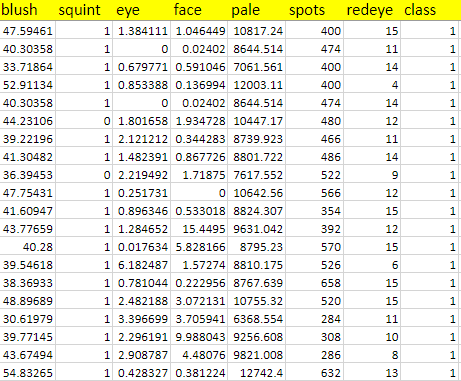
Figure 2: Redness detection in an alcoholic person’s eye



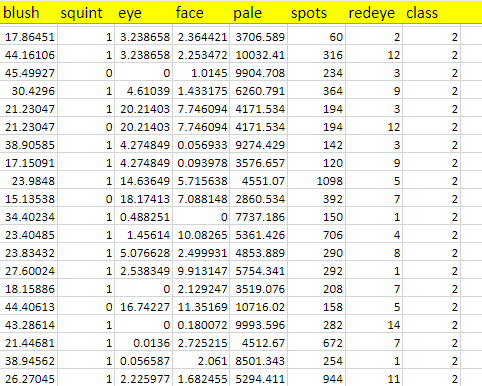


## Snippet of excel sheets:

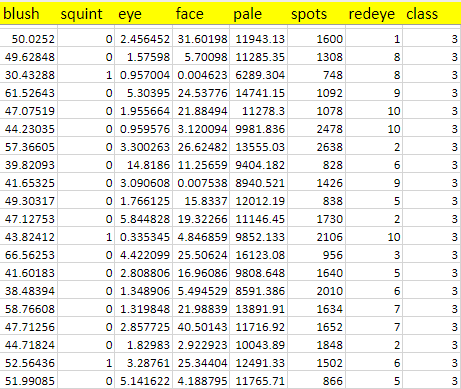
### Excel sheet for alcoholics:



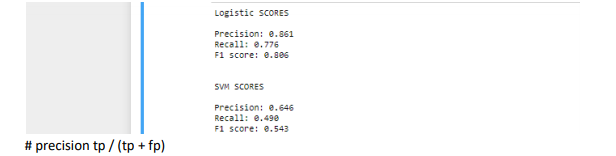
## Excel sheet for drug abusers:

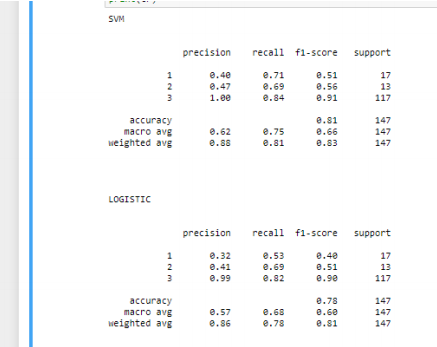


## Excel sheet for non-substance abusers:



## Machine Learning Code Result:





# Conclusion:

As the problem statement of this project specifies, our end goal is to predict alcohol and drug using tendencies in an individual and though it would be incorrect to label a non substance abuser as an alcohol or drug addict, it would be even more catastrophic to label an alcohol or drug addict a non substance abuser. As we can see, recall values using SVM are high

# References:

<https://www.insider.com/stop-drinking-what-happens-to-your-skin-2018-8>

<https://www.gq.com/story/how-bad-is-drinking-alcohol-for-your-skin>

<https://discoveryplace.info/addiction-blog/the-visible-signs-of-drug-abuse/>

# Appendix:

Codes written for the project are mentioned below

## To Obtain if the Person’s Eyes are Squinting:

%to sharpen image

I = imread('C:\Users\0wner\Desktop\VIT\image processing\project\drug1.jpg');

%I = imresize(img, [336 295]);

%Kaverage = filter2(fspecial('average',3),J)/255; B= imsharpen(I,'Radius',1,'Amount',2); EE=step(EyeDetect,B); EyeDetect = vision.CascadeObjectDetector('EyePairBig'); FaceDetect = vision.CascadeObjectDetector();

EE=step(EyeDetect,B); for k= 1:size(EE,1) eye= imcrop(B, EE(k,:));

end figure, imshow(eye);

[centers,radii] = imfindcircles(eye,[5 7],'ObjectPolarity','dark', ... 'Sensitivity',0.92,'Method','twostage','EdgeThreshold',0.31);

## To Obtain Number Of Spots On Person’s Face:

data= imread('C:\Users\0wner\Desktop\VIT\image processing\project\normal.jpg');

%temp = imresize(temp1,[6000,6000]);

FDetect = vision.CascadeObjectDetector;

BB = step(FDetect,data); for j=1:size(BB,1) I= imcrop(data, BB(j,:));

end

MouthDetect = vision.CascadeObjectDetector('Mouth','MergeThreshold',300); MM=step(MouthDetect,I);

EyeDetect = vision.CascadeObjectDetector('EyePairBig','MergeThreshold',50);

EE=step(EyeDetect,I);

hold on

for x = 1:size(MM,1)

rectangle('Position',MM(x,:),'LineWidth',4,'LineStyle','

','FaceColor','0,0,0','EdgeColor','black');

end

for y = 1:size(EE,1)

rectangle('Position',EE(y,:),'LineWidth',4,'LineStyle', '-

','FaceColor','0,0,0','EdgeColor','black');

end %

spots=0;

Rmin =1;

Rmax =15;

[centersBright, radiiBright] = imfindcircles(I,[RminRmax],'Sensitivity',0.976,'Obj ectPolarity','bright'); [centersDark, radiiDark] = imfindcircles(I,[Rmin Rmax],'Sensitivity',0.976,'ObjectPolarity','dark'); %imshow(I);

viscircles(centersBright, radiiBright,'Color','b');

viscircles(centersDark, radiiDark,'LineStyle','-- ');

%hold off;

[brows, bcolumns, numberOfColorChannels] = size(centersBright);

[drows, dcolumns, numberOfColorChannels] = size(centersBright);

spots= brows+ drows;

## To Obtain The Amount Of Wrinkles:

EyeDetect = vision.CascadeObjectDetector('EyePairBig');

FaceDetect = vision.CascadeObjectDetector();

B= imread('C:\Users\0wner\Desktop\VIT\image processing\dpic4.jpg');

data= imsharpen(B,'Radius',1,'Amount',2);

% BB=step(FaceDetect,I);

%for j= 1:size(BB,1)

%face = imcrop(I, BB(j,:));

%end BB = step(FaceDetect,data);

for j=1:size(BB,1)

I= imcrop(data, BB(j,:));

end

EE=step(EyeDetect,data);

for k= 1:size(EE,1) eye = imcrop(data, EE(k,:));

end

new\_eye\_image= rgb2hsv(eye);

%imshow(new\_eye\_image);

ewp=0;

havg=0;

savg=0;

vavg=0;

[rows, columns, numberOfColorChannels] = size(new\_eye\_image(:,:,1));

for x=1:rows for y=1:columns pixel= new\_eye\_image(x,y);

if (pixel~= 1 ) havg = havg+pixel(:,:,1);

end

end

end

[rows, columns, numberOfColorChannels] = size(new\_eye\_image(:,:,2));

swp=0 for x=1:rows for y=1:columns pixel= new\_eye\_image(x,y);

if (pixel ~= 1 ) savg = savg+pixel; swp= swp+1;

end

end

end

sfavg= savg/swp;

[rows, columns, numberOfColorChannels] = size(new\_eye\_image(:,:,3));

vwp=0 for x=1:rows for y=1:columns pixel= new\_eye\_image(x,y);

if (pixel ~= 0 ) vavg = vavg+pixel;

vwp= vwp+1;

end

end

End

vfavg= vavg/vwp;

perc\_wrinkle = ((ewp+vwp+swp/(rows\*columns))\*100;

Code for machine learning:



