**OVERVIEW**

The etiology of mental illness is believed to be a complex interaction between genetics, physiology, neurobiology, and environmental factors that lead to psychological, physiological, and/or behavioural changes. When these deviations differ significantly from societal norms and interfere with one's ability to function in daily life, the person may be diagnosed with a mental disorder (American Psychiatric Association, 2000)

**PROBLEM STATEMENT**

A major impediment to accomplishing this goal, in my view, is the lack of careful application of some of the tenets of applied behaviour analysis to areas that are typically described as “psychological” or “emotional” rather than “behavioural”. This needs to be addressed directly. As behaviour analysts, we must be willing to use terms outside of our discipline, but insist on operational definitions for these terms when we use them.

**ABSTRACT:**

The field of applied behaviour analysis (ABA) has increasingly come to be associated with the treatment of autism in young children. This phenomenon is largely the result of empirical research demonstrating effective treatment outcomes in this population. The same cannot be said with regard to the treatment of conditions often referred to as emotional or psychological problems. The current article describes the philosophical and descriptive differences that likely account for the lack of application of ABA in these areas and proposes potential solutions to help ABA practitioners more effectively address these issues. Specifically, the issue of how to objectively describe these “conditions” needs to be addressed so that careful study of treatment effects can occur in a manner similar to the way that brought ABA to prominence in autism treatment.

**INTRODUCTION**

The etiology of mental illness is believed to be a complex interaction between genetics, physiology, neurobiology, and environmental factors that lead to psychological, physiological, and/or behavioral changes. When these deviations differ s'ignificantly from societal norms and interfere with one's ability to function in daily life, the person may be diagnosed with a mental disorder (American Psychiatric Association, 2000). Often a licensed physician, psychiatrist, or psychologist assesses an individual, diagnoses a mental disorder, and then designates a treatment plan for that individual. Although an interdisciplinary approach, where in representatives from various disciplines such as medicine, psychiatry, clinical psychology, neuroscience, education, social work, and behaviour analysis convene to devise a treatment plan would be preferable, the logistics and resources required limit this practice to select clinical facilities. We posit that behaviour analysis, which includes refined techniques for teaching and motivating adaptive behavior, should be an integral part of a multidisciplinary approach to mental health services. Combining technologies derived from behavior analysis and other disciplines could broaden our understanding of mental disorders, expand the range of available interventions, and improve therapeutic outcomes and client satisfaction. This article briefly examines early applied behavior analysis (ABA) research with mental disorders, the development of functional behavior assessment and functional analysis of behavior problems, potential contributions of ABA to multiolsciplinary mental health services, and recent ABA studies with mental disorders in children and adults. While covering these topics, the present article highlights some of ABA's technological developments within mental health services and special challenges it has faced. The treatment of individuals with mental illness changed dramatically during the 20th century as custodial arrangements progressed to a mix of educative and therapeutic programs within mental hospitals, outpatient clinics, and community-based facilities (Braddock & Parish, 2002). The use of psychosurgery and electroconvulsive shock therapy decreased as pharmacology became the treatment of choice for many mental health impairments (Braddock & Parish, 2002; Wong, 2006). A parallel change has been occurring within the field of behavior analysis as its investigations have extended from basic research with nonhuman animals in . laboratories to improving socially significant behavior of humans in applied settings (Baer, Wolf, & Risley, 1968, 1987). Much of the early research within the field of ABA was conducted within state mental hospitals using operant procedures such as token economies, reinforcement procedures, shaping, and extinction for persons with severe mental disorders such as schizophrenia (Ayllon & Azrin, 1965; Ayllon & Haughton, 1964; Ayllon & Michael, 1959). Over the last 50 years, behavior analysis has been successfully applied in mental institutions and community-based facilities to increase social, self-care, vocational, leisure, and recreational skills while concurrently reducing behavioral problems such as delusional speech, bizarre behavior, and aggression (Wong, 1996; Wong, Wilder, Schock, & Clay, 2004). Despite beneficial outcomes, behavior analytic techniques are often underused or supplanted by interventions with limited scientific support (Scotti, Morris, McNeil, & Hawkins, 1996; Wong, 2006). The enhanced prognostic and therapeutic outcomes associated with ABA indicate that this approach could contribute much to the treatment of clients with mental health disorders. A central premise of ABA is that focusing on observable behaviors provides an objective and empirically based framework for the assessment and treatment of mental disorders (Scotti et al., 1996; Wong, 1996). By concentrating on behavioral manifestations of mental disorders clinicians can obtain specific and independently verifiable measures of clients' problems. This method can also facilitate the discovery of flmctional relations between overt behavior and environment stimuli, leading to interventions that reengineer aspects of clients' social and physical surroundings. Assessment of mental health problems is complicated by reliance on selfreports of mental states, often evaluating covert behaviors and unobservable events. Interpreting the roles of mental events and behavioral sequela are a challenging endeavor with multiple confounding variables that must be controlled or, ruled out during the course of treatment (MacCotquodale & Meehl, 1948). For example, it.can often be difficult to disentangle the sedating and enervating effects of medications from the symptoms of a mental disorder, such as the negative symptoms of schizophrenia (Wilder & Wong, 2007). Current ABA research with mental health disorders uses mixed assessment methods, employing direct observation of overt behaviors as well as interview, questionnaire, and other self-report measures of covert behavior and internal processes to overcome the limitations of singular assessment procedures.

**EXISTING SYSTEM**

As behaviour analysts, we must be willing to use terms outside of our discipline, but insist on operational definitions for these terms when we use them. For example, a behaviour analyst can treat a child who presents with a “mood disorder” by specifying the behavioural evidence of the disorder. Is it that the child is often happy, but becomes enraged when told “no” or when there is a change in their schedule? Is it that they describe high levels of variability in their mood and would like to describe more stable levels?, or it is that the child behaves in certain ways more often than we would like (hitting, yelling, inactivity,) and describe this as evidence of a mood disorder? We can, if we choose to, make specific and measureable the evidence for the disorder/diagnosis, and then apply treatment. Subsequent evaluation of levels of the symptoms can enable us to determine empirically if treatment has reduced, increased or had no effect on these symptomms

**PROPOSED SYSTEM**

In this project student want to detect people’s mental behaviour by analysing there’s emotions and to implement this project we have used Facial Expression dataset and SVM machine learning algorithm. SVM is trained with facial expression dataset and then SVM trained model can be used to detect human mental behaviour. To implement this project we have designed following modules

1. Webcam connection: using this module application will be connected to live webcam
2. Load & Preprocess Dataset: using this module application read all dataset images from numpy array and then normalize and extract features from images.
3. Train SVM Algorithm: Extracted features will be used to train SVM algorithm
4. Capture Person: using this module we will capture person image and then detect face from that image
5. Detect Emotion: This module will take detected face as input and then by using SVM algorithm will predict person mental behaviour as SAD, HAPPY, NEUTRAL, ANGRY etc.

**LITERARTURE REVIEW**

# IN “SENSING BEHAVIORAL SYMPTOMS OF MENTAL HEALTH AND DELIVERING PERSONALIZED INTERVENTIONS USING MOBILE TECHNOLOGIES” Unlike most other health conditions, the treatment of mental illness relies on subjective measurement. In addition, the criteria for diagnosing mental illnesses are based on broad categories of symptoms that do not account for individual deviations from these criteria. The increasing availability of personal digital devices, such as smartphones that are equipped with sensors, offers a new opportunity to continuously and passively measure human behavior in situ. This promises to lead to more precise assessment of human behavior and ultimately individual mental health. More refined modeling of individual mental health and a consideration of individual context, assessed through continuous monitoring, opens the way for more precise and personalized digital interventions that may help increase the number of positive clinical outcomes in mental healthcare. In this paper, we provide a conceptual review of such techniques for measuring, modeling, and treating mental illness and maintaining mental health Over the last 10 years, technology has become more proximal to human activity. As more and more people adopt today’s technology, healthcare involving technology in some respect is becoming increasingly acceptable. Ownership of smartphones is especially prevalent among underserved minority groups: 47% of black non-Hispanics and 49% of Hispanics own smart phones, compared to 42% of non-Hispanic whites (Pew Research Center, 2013). From the perspective of mental healthcare, mobile technology appears to be a feasible medium for delivering care; for example, a recent community-based survey of over 1,500 people with serious mental illnesses found that over 80% of patients with bipolar disorder (BD) owned and used mobile phones regularly for calling, texting, and the internet (Ben-Zeev, Davis, Kaiser, Krzsos, & Drake, 2013). The evolution and rapid dissemination of mobile and sensor technology has created unprecedented opportunities for personalized data collection in an extremely granular, unobtrusive, and even affordable way. As a result, there has been a recent and significant increase in research to identify novel methods to measure, understand, and treat mental illness (and improve mental well-being). However, in terms of day-to-day clinical practice technological approaches for measuring and treating mental illness are far behind developments in general health. This shortcoming is particularly important given that the accurate understanding of daily behavior is central to mental health and wellness assessment. The rise of Personal Informatics, consumer friendly tools that allow individuals to conveniently track a wide range of elements including mood, activity, and sleep, alongside the related Quantified Self movement, which focuses on using these tools to empower individuals, is enabling people to have more control over an ever-increasing amount of personal data. As early as 2006, the Institute of Medicine recognized that these technologies had the potential to transform mental health services by providing more continuous and precise information on patient-specific behavior, symptoms, and medication side effects. To transform this potential into practice requires that we develop methods to transform raw data into actionable knowledge and then use this in turn to support personalized interventions. In this paper, we provide a broad overview of recent innovations in digital technologies to measure, understand, and intervene in mental healthcare and finish by presenting a three-stage framework that can be used when integrating such technologies into clinical practice. In practice, mental illnesses are defined by clinical criteria outlined in diagnostic manuals like the Diagnostic and Statistical Manual (DSM).Many of these criteria center on assessing behavior as a direct 2017 Wiley Periodicals, Inc. symptom of a mental illness. For example, decreased physical activity, abnormal sleep patterns, and social interactions are well-documented symptoms of depression (American Psychiatric Association, 2013). Significant changes in the amount of physical and mental activity can be a behavioral indicator of manic and depressive states in BD (Goodwin, & Jamison, 2007). Currently these conditions are measured using subjective questionnaires that have well-identified limitations (Stone, Shiffman, Schwartz, Broderick, & Hufford, 2003). In addition, the diagnostic criteria provide general bins in which to categorize symptoms, yet there may be other symptomatic and prodromal behaviors shared among some individuals or that are individual specific that have not yet been detected because there has been no way to detect them.Some researchers have called for a move away from traditional methods of diagnosis such as the DSM-5—criteria used by mental health professionals to assign a diagnosis to patient—toward treatments that “convey the reality of the patient’s experience” (Licinio, 2005). Where current clinical criteria for mental illness make it easier for clinicians to identify and treat clusters of common symptoms, in reality each mental illness category—be it anxiety disorder or major depression—are umbrella terms for heterogeneous diseases. There is a growing recognition, evidenced by the creation of the Research Domain Criteria (RDoC), of the potential for neuroscience and genetics to transform the classification of disorders into clusters of pathophysiological symptoms (Insel et al., 2010). Although RDoC is promising for clinical research, it is not clear how long it will be before genetic and neuroscientific classifications of mental illnesses will be ready for clinical use (Kupfer, & Regier, 2011).Objective behavior sensing via personal digital devices offers the potential to bridge this gap between the classical diagnostic categories and the fundamental neuroscientific classifications. There has been a marked rise in the use of personal technologies to automatically detect behavior, as evidenced by the growing number of commercially available devices for with inbuilt sensors. An increasing number of studies demonstrate the promise of behavior-monitoring like this to mental health (Lane et al., 2010).There are two main methods for assessing behavior related to mental health:

# Measurement of real-world behavior through sensors (most commonly using sensors embedded in smartphones).

# Measurement of behavior mediated through technology (e.g., assessing mental health via social media behavior).

## **IN “**[**MENTAL HEALTH, MORBIDITY AND IMPACT**](https://www.sciencedirect.com/science/article/pii/B9780128036785002885)**”** The World Health Organization estimates that up to 450 million people are affected by mental, neurological and behavioral disorders worldwide. These disorders include unipolar depression, bipolar affective disorder, schizophrenia, alcohol and drug use disorders, posttraumatic disorder, panic disorder, Alzheimer’s disease and other dementias, and primary insomnia. The prevalence of mental disorders is generally higher than that of any other class of chronic conditions and this is further reflected in the fact that four out of the six leading causes of years lived with disability are neurological or mental disorders. Even though estimates vary, depending on definition and ascertainment methods, in general approximately one in every four individuals will develop one or more mental disorders in their lifetime. Cross-national estimates of 12-month prevalence of between 4.3% and 26.4% have been reported in studies conducted among large community samples. In the community, the most common disorders are anxiety, mood, and substance use disorders. The burden attributable to mental disorders results not only from their high prevalence but also from the relatively early age of their onset as well as their tendency to be chronic or recurrent. For example, the median age of onset for anxiety disorders is early teenage and many affected individuals will go on to develop other types of mental disorder in adulthood. Mental disorders are highly prevalent in the community. Large-scale community surveys, which have only become possible with the development in the last few decades of reliable lay-administered interviews, have shown that between 25% and 50% of adults will develop one mental disorder or the other in their lifetime. In a 12-month period, between one-in-ten and one-in-five adults will have significant levels of symptoms sufficient for a categorical diagnosis. Even though such estimates have generated controversy and concern about their reliability, their replication in several different settings has provided credibility. These estimates have varied depending on the mode of ascertainment, the diagnostic categories covered, and the age group studied. There is also variability between countries. For example, the largest mental health survey ever conducted, the World Mental Health Surveys, reported rates of 12-month disorder that vary between 4.7% in Nigeria and 26.3% in the United States (Table 1). Whether this reflects the performance of the assessment tools, the reporting styles of people from different cultural backgrounds, or a true difference in propensity to develop mental disorders is still unknown. The likelihood is that some or all of these factors are involved. Irrespective of where they are conducted, it is a common observation that prevalence rates in the general adult population typically underestimate projected lifetime risk, so that more people are indeed likely to develop mental disorders in their lifetime than cross-sectional estimates suggest. The most common group of mental disorders in the community is anxiety disorders. Lifetime estimates of anxiety disorders of up to 25% have been commonly reported. Among these, specific phobia is the most prevalent often followed by social phobia, posttraumatic disorder, and generalized anxiety disorder. Mood disorders, in particular major depressive disorder, are also highly prevalent, with some studies suggesting that up to one in five adults may experience at least one episode of depression in their lifetime. Substance use disorders may affect up to 10% of adults in their lifetime, with alcohol abuse the most prevalent condition reported. While anxiety and mood disorders tend to be more prevalent among females, males are commonly the more likely to report substance use disorders. Rates of substance use disorders, as a group, tend to vary considerably between cultural settings and age groups. However, there is now a common observation for a trend for a prominent cohort effect in which higher rates of these disorders are often to be found in the teens and young adults. Nonaffective psychotic disorders and dementia are less common in the community. Schizophrenia has a lifetime risk of about 1%. There is evidence that males are more affected than females, with a male to female ratio of about 1.4, and that migrants, especially second-generation migrants, tend to have higher incidence than native-born individuals. Bipolar disorder also has a lifetime morbid risk of about 1%. However, recent studies suggest that subthreshold bipolar syndrome, which is also a disabling disorder, has a much higher prevalence in the community. In addition, several studies have now documented widespread experience of psychotic symptoms in the community, although the import for such experiences in regard to disability is not yet fully understood. About 1% of persons aged 65 years will have dementia. However, with the prevalence of the disorder doubling every 5 years, over 40% of elderly persons 90 years and above will have the disorder.

**IN “W. L. OU, M. H. SHIH, C. W. CHANG, X. H. YU, C. P. FAN, "INTELLIGENT VIDEO-BASED DROWSY DRIVER DETECTION SYSTEM UNDER VARIOUS ILLUMINATIONS AND EMBEDDED SOFTWARE IMPLEMENTATION", 2015 INTERNATIONAL CONF. ON CONSUMER ELECTRONICS - TAIWAN, 2015”** An intelligent video-based drowsy driver detection system, which is unaffected by various illuminations, is developed in this study. Even if a driver wears glasses, the proposed system detects the drowsy conditions effectively. By a near-infrared-ray (NIR) camera, the proposed system is divided into two cascaded computational procedures: the driver eyes detection and the drowsy driver detection. The average open/closed eyes detection rates without/with glasses are 94% and 78%, respectively, and the accuracy of the drowsy status detection is up to 91%. By implementing on the FPGA-based embedded platform, the processing speed with the 640x480 format video is up to 16 frames per second (fps) after software optimizations. For the past few years, accidents caused by fatigue driving have occurred frequently. Therefore, many researchers and experts have paid great efforts in this issue [1-3]. Some useful techniques for detecting driver drowsiness can be generally separated into three main categories [1]: the first category is based on driver’s current state, which is relating to the eye and eyelid movements, closed eyes period, and physiological state changes. The second category is based on the vehicle’s behavior, e.g. the driving speed. The third category is based on combination of the driver’s current state and driver performance [1]. Regarding the fatigue detection issue, the driver’s drowsy status can be evaluated through the closed eyes and head gesture conditions [2-3] by using facial image processing. In this work, by the NIR-based facial image processing, firstly the accurate eye position is recognized, and then the driver drowsy condition without/with glasses can be detected effectively. Finally, the fatigue alert is generated to warn the driver properly

**IN “W. B. HORNG, C. Y. CHEN, Y. CHANG, C. H. FAN, “DRIVER FATIGUE DETECTION BASED ON EYE TRACKING AND DYNAMIC TEMPLATE MATCHING”, IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL, TAIPEI, TAIWAN, MARCH 21-23, 2004.”** A vision-based real-time driver fatigue detection system is proposed for driving safely. The driver’s face is located, from color images captured in a car, by using the characteristic of skin colors. Then, edge detection is used to locate the regions of eyes. In addition to being used as the &namic templates for eye tracking in the next frame, the obtained eyes’ images are also used for fatigue detection in order to generate some warning alarms for driving safety. The system is tested on a Pentium Ill 550 CPU with 128 MB RAM. The experiment results seem quite encouraging andpromising. The system can reach 20 fiames per second for eye tracking, and the average correct rate for eye location and tracking can achieve 99.1% on four test videos. The correct rate for fatigue detection is loo%, but the average precision rate is 88.9% on the test videos It is a hard test of endurance for drivers to take long distance driving. It is very difficult for them to pay attention to driving on the entire trip unless they have very strong willpower, patience, and persistence. Thus, the driver fatigue problem has become an important factor of causing traffic accidents. A driver fatigue detection system can detect whether the driver is tired, such as dozing or inattention, so as to generate some warning alarms to alert the driver. Therefore, the system can reduce not only traiffic accidents hut also the social cost caused by these accidents. Because human eyes express the most direct reaction when dozing, eye blinking is usually used as the basis for driver fatigue detection by researchers. In recent years image processing on human faces has been used in many applications, such as face recognition, face analysis [4], eye detection [6], gaze tracking [9], etc. Among all of these researches, the fust step is usually to locate faces. Recently, the human face detection techniques have matured gradually. The detection techniques can he Chuu-Hai Fan Department of Transportation Management Tamkang University Taipei, Taiwan, R.O.C. chunhai@mail.tku.edu.tw divided into two major categories. The first is based on face features [4][6]. The second is based on face colors [1][7- 111. Feature-based face detection methods utilize some well-known knowledge on human faces, such as the shape of the face, the relative locations of the eyes, the nose, and the mouth in a face, and so forth, to judge whether the observed features on an image satisfy such criteria. On the other hand, color-based face detection methods build on specific color models to locate faces based on skm colors. Smce skin colors have quite stable distrihution in some color models, face detection based on skin colors becomes more popular. Io this paper, the HSI color model is used to locate faces since face colors have fmed distribution range on the hue component of the HSI model, decreasing the influence of brightness changes. When observing a facial image, the most perceptible facial features are usually the places with obvious edges, particularly the ocular outlines. Since eyes have the nature of sophisticated edges, many researches use this attribute for eye detection [1][2][4][7]. In this paper the Sohel edge operator and projection technique are used for eye location. The obtained eyes’ images are used as the dynamic templates for eye tracking in the next kame. Besides, they are also used to determine whether the eyes are open or closed, which is the basis of driver fatigue detection in this research. In the past ten years, many countries all over the world have begun to pay attention to the driving safety problem and to investigate the drivers’ mental states relating to driving safety. Also, waming systems for drivers are also presented [1][2][7]. In 1997 and 1999, Eriksson and Papanikolopoulos [2] and Singh and Papankolopoulos [7], respectively, proposed two papers on driver fatigue detection based on image processing techniques for driving safety. In [2], the authors used the symmetric property of faces to detect facial area on an image. Then, they used pixel difference to fmd the edges on the facial region to locate the vertical position of the eyes. Since the edge detection they used cannot clearly mark edges, it is not easy to locate accurate ocular locations. The authors used thresholding to improve the location of the ocular places. Afier finding the approximate eyes positions, a concentric circle template was designed to locate the exact eyes locations, and the template was used to track eyes in the following images. Face symmetry is an obvious feature for an upright face. However, it usually fails to locate the correct face position when the face tilts, rotates, or is shadowed. Instead of using symmetric central line method, the authors in [7] used the Gaussian distribution of the skm colors to distinguish skin and non-skin pixels [lO][ll] for face detection. The Sobel vertical edge operator was used for eye detection. Besides, they built a database of eye images as the templates for eye detection and tracking. Although the Gaussian distribution of skin colors based on the RGB color model was used to predict skin quite well, the method cannot get rid of the factor of brightness changes. In addition, the eye images in the database as templates may be quite different kom drivers' eyes, which will reduce the accuracy for eye location. In order to cope with the problem of intensity changes, the HSI color model is adopted to represent face skm in this paper since the model detaches the intensity from the hue of a color. Furthermore, the obtained eyes' images are used as the dynamic templates for eye tracking in the paper, instead of using the.static eye images in the database as used in the above method, to enhance the reality and reliability of the templates so as to increase accuracy of eye tracking.

**IN “S. SINGH, N. P. PAPANIKOLOPOULOS, “MONITORING DRIVER FATIGUE USING FACIAL ANALYSIS TECHNIQUES”, IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEM, PP 314-318.”** In this paper, we describe a non-intrusive visionbased system for the detection of driver fatigue. The system uses a color video camera that points directly towards the driver’s face and monitors the driver’s eyes in order to detect micro-sleeps (short periods of sleep). The system deals with skin-color information in order to search for the face in the input space. After segmenting the pixels with skin like color, we peflorm blob processing in order to determine the exact position of the face. We reduce the search space by analyzing the horizontal gradient map of the face, taking into account the knowledge that eye regions in the face present a great change in the horizontal intensity gradient. In order to find and track the locatibn of the pupil, we use gray scale model matching, We also use the same pattern recognition technique to determine whether the eye is open or closed. If the eyes remain closed for an abnormal period of time (5-6 sec), the system draws the conclusion that the person is falling asleep and issues a warning signa A large number of automobile accidents are caused due to driver fatigue. The U.S. National Highway Traffic Safety Administration has indicated that driver fatigue is responsible for as many as 240,000 motor vehicle accidents a year in the U.S.A. alone. Sleep deprivation and sleep disorder are becoming common problems for car drivers in a society in which people seem not to have enough time to perform all the activities they need to carry out on a daily basis. By monitoring the driver’s symptoms, we can determine driver fatigue early enough as to take preventive course to avoid an accident due to lack of awareness. There are many indicators of oncoming fatigue, some of which are possible to detect with the use of a camera. Two well-known symptoms that we consider feasible to be detected are micro-sleeps (short periods, 2-3 seconds, in which the driver loses consciousness) and the forward or backward bouncing movement of the driver’s head. At the moment, we focus on the detection of microsleeps. The input to the system is a continuous sequence of images fed from a video camera. From this sequence, the system can analyze the eyes in each image, as well as compare the eyes between frames. A previous version of the system [l] used computer vision techniques to extract and track eye locations throughout the entire video sequence. However, the techniques described in this paper have significant forward steps towards improved accuracy and robustness in the process of detecting driver fatigue. Localizing the eyes at the first frame is the most computationally expensive phase of the tracking system. In this phase, the system has no previous information about the eye locations in the image. The system has to find the area of the image that will be used in subsequent frames in order to track the eyes. During the tracking phase, the search space is reduced as the system has an approximate knowledge of the eye’s position from the previous frame. This tracking can be done at a relatively low computational cost. In order to detect failure in tracking, general constraints such as the distance between the eyes and the horizontal alignment of the two eyes are used. To make sure that the correct feature is being tracked, the eyes should be relocated periodically, even if no failure has been detected. By determining whether the eyes are open or closed during eye tracking, we can determine if there are any micro-sleep symptoms that can help us determine driver fatigue.

**IN “B. ALSHAQAQI, A. S. BAQUHAIZEL, M. E. A. OUIS, M. BOUUMEHED, A. OUAMRI, M. KECHE, “DRIVER DROWSINESS DETECTION SYSTEM”, IEEE INTERNATIONAL WORKSHOP ON SYSTEMS, SIGNAL PROCESSING AND THEIR APPLICATIONS, 2013”** Drowsiness and Fatigue of drivers are amongst the significant causes of road accidents. Every year, they increase the amounts of deaths and fatalities injuries globally. In this paper, a module for Advanced Driver Assistance System (ADAS) is presented to reduce the number of accidents due to drivers fatigue and hence increase the transportation safety; this system deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence. We propose an algorithm to locate, track, and analyze both the drivers face and eyes to measure PERCLOS, a scientifically supported measure of drowsiness associated with slow eye closure Currently, transport systems are an essential part of human activities. We all can be victim of drowsiness while driving, simply after too short night sleep, altered physical condition or during long journeys. The sensation of sleep reduces the driver's level of vigilance producing dangerous situations and increases the probability of an occurrence of accidents. Driver drowsiness and fatigue are among the important causes of road accidents. Every year, they increase the number of deaths and fatalities injuries globally. In this context, it is important to use new technologies to design and build systems that are able to monitor drivers and to measure their level of attention during the entire process of driving. In this paper, a module for ADAS (Advanced driver assistance System) is presented in order to reduce the number of accidents caused by driver fatigue and thus improve road safety. This system treats the automatic detection of driver drowsiness based on visual information and artificial intelligence. We propose an algorithm to locate, track and analyze both the driver face and eyes to measure PERCLOS (percentage of eye closure). Some efforts have been reported in the literature on the development of the not-intrusive monitoring drowsiness systems based on the vision. Malla et al. [1] develop a light-insensitive system. They used the Haar algorithm to detect objects [2] and face classifier implemented by [3] in OpenCV [4] libraries. Eye regions are derived from the facial region with anthropometric factors. Then, they detect the eyelid to measure the level of eye closure. Vitabile et al. [5] implement a system to detect symptoms of driver drowsiness based on an infrared camera. By exploiting the phenomenon of bright pupils, an algorithm for detecting and tracking the driver's eyes has been developed. When drowsiness is detected, the system warns the driver with an alarm message. Bhowmick et Kumar [6] use the Otsu thresholding [7] to extract face region. The localization of the eye is done by locating facial landmarks such as eyebrow and possible face center. Morphological operation and Kmeans is used for accurate eye segmentation. Then a set of shape features are calculated and trained using non-linear SVM to get the status of the eye. Hong et al. [8] define a system for detecting the eye states in real time to identify the driver drowsiness state. The face region is detected based on the optimized Jones and Viola method [2]. The eye area is obtained by an horizontal projection. Finally, a new complexity function with a dynamic threshold to identify the eye state. Tian et Qin [9] build a system that checks the driver eye states. Their system uses the Cb and Cr components of the YCbCr color space. This system locates the face with a vertical projection function, and the eyes with a horizontal projection function. Once the eyes are located the system calculates the eyes states using a function of complexity. Under the light of what has been mentioned above, the identification of the driver drowsy state given by the PERCLOS is generally passed by the following stages: 1) Face detection, 2) Eyes Location, 3) Face and eyes tracking, 4) Identification of the eyes states, 5) Calculation of PERCLOS and identification of driver state

**IN “M. KARCHANI, A. MAZLOUMI, G. N. SARAJI, A. NAHVI, K. S. HAGHIGHI, B. M. ABADI, A. R. FOROSHANI, A. NIKNEZHAD, “THE STEPS OF PROPOSED DROWSINESS DETECTION SYSTEM DESIGN BASED ON IMAGE PROCESSING IN SIMULATOR DRIVING”, INTERNATIONAL RESEARCH JOURNAL OF APPLIED AND BASIC SCIENCES, VOL. 9(6), PP 878-887, 2015.”** Drowsiness detection has many implications including reducing roads traffic accidents importance. Using image processing techniques is amongst the new and reliable methods in sleepy face. The present pilot study was done to investigate sleepiness and providing images of drivers' face, employing virtual-reality driving simulator. In order to detecting level of sleepiness according to the signal, information related to 25 drivers was recorded with imaging rate of 10 fps. Moreover, on average 3000 frames was analysed for each driver. The frames were investigated by transforming in grey scale space and based on the Cascade and Viola & Jones techniques and the images characteristics were extracted using Binary and Histogram methods. The MPL neural network was applied for analysing data.70% of information related to each driver were inserted to the network of which 15% for test and 15% for validation. In the last stage the accuracy of 93% of the outputs were evaluated. The intelligent detection and usage of various criteria in long-term time frame are of the advantages of the present study, comparing to other researches. This is helpful in early detection of sleepiness and prevents the irrecoverable losses by alarming. Keywords: drowsiness, driving, image processing Driver drowsiness detection contributes to the decrease in the number of deaths occurring in traffic accident. Driver drowsiness is an overcast nightmare to passengers. This nightmare has been realized in Iran and we regularly observe mishaps leading to death and serious injuries. Traffic accident is not a problem only in Iran but also is a global problem. According to the recent researches by SLEEP RESEARCH CENTER (Lough Borough) in England , drowsiness and fatigue are responsible for majority of car accidents in England [Narelle L et al (1988 June)] .Generally, methods developed to detect drowsy drivers are classified in two categories ; surveillance methods and methods based on the movement of car. In surveillance method required equipments comprise sensors and camera to record user's visual characteristics. Then, these characteristics are transferred to a computer to determine driver vigilance after processing the driver’s characteristics. It is worth to mention that these methods are more popular and are different in comparison with the second category. Nevertheless, difficulties to utilize devices and costs confine the utility of these methods. According to another study, drowsiness is recognized as follows sensors: 1.physiological indices based 2.Driver's performance based 3.Vehicle movements based methods Methods developed on the basis of user's physiological and natural effects are efficient. These methods function in two ways: 1.Measure changes in physiological signals such as EEG, ECG, EOG. 2.Evaluation of physical changes like obliquity, inclination, deviation to sides, closed or open eyes. First methods are intrusive because some electrodes should be put on the driver’s body. In addition to this, during a long time drive, driver's perspiration diminishes the sensitivity of sensors. Monitoring driver's performance is a practical technique in which a camera is placed on front of the driver to detect drowsiness. Many methods have been developed to determine influences of drowsiness on driver's performance consisting of: Driver's performance and automobile behavior by monitoring the steering wheel movement, acceleration or brake pattern, velocity, lateral acceleration and lane deviation. These techniques are external methods according to the conditions and vehicle [Hargutt V, Study H (2000)]. Consequently, the target of this study is to represent a new experimental technique on the basis of histogram models eyes which determines eyes situation and blinks and tracks these facial expressions in a sequence of images. In latter systems, intelligent algorithms like neural network are employed to study and monitor information gathered by cameras, wheels rotations and accelerations of automobile. By adding the information about dynamics of automobile error is reduced. To do this, companies such as FORD and VOLVO have developed some systems to detect driver's drowsiness [Narelle L et al (1988 June)]. When the driver is drowsy, facial features especially eyes and mouth situations and sitting manner change. By taking images and using image processing approaches, visual signs of drowsiness are inferred [Bergasa L et al (2008)]. Drivers show particular behaviors while they are fatigued and drowsy and these behaviors are obviously considering visual changes in facial expression. Longer blink duration, slow movement of eyelids, eyes closure and even closed eyes, repeated nodding, yawning, gazing eyes; anesthesia, bout and hang over are most common visual traits amongst drivers. Machine vision is not annoying to recognize sleepy driver's visual signals [Khan M, I, Mansoor A, B (2008)]. In this technique, images are taken by a camera placed in front of the driver, then, desired signs are extracted by car sights and image processing techniques. Majority of investigations conducted in this area focus on eyes and inferring eyes signals. Eyes study is a strong technique to detect drowsiness in which movements of eyes and changes are investigated. Most important changes are the number of blinks, duration of closed eyes and impudence path [Dong W, Wu X (2005)]. Eye recognition and tracking in image trace is widely used especially in driver's fatigue detection. By planning and conducting a system to detect driver's fatigue, the probability of car accident decreases. Since drowsiness detection is crucial to collect information about physiological signals and, the level of alertness is important to find the path and drive the car, in this study, after investigations, we introduce a model to improve alarm system. This system is efficient to detect driver's fatigue as well as is quick enough without any restraint [Ryan W, J et al (2008)]. Regularly, the most effective technique to measure the level of drowsiness and lack of vigilance is to measure brain waves and heart beating rate. This method is troublesome since different sensors should be put on the driver's body. On the other hand , blink diagnosis is an effective and precise technique because it is not an annoying technique

**INPUT AND OUTPUT DESIGN**

**INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**SYSTEM STUDY**

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**Three key considerations involved in the feasibility analysis are,**

* **ECONOMICAL FEASIBILITY**
* **TECHNICAL FEASIBILITY**
* **SOCIAL FEASIBILITY**

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

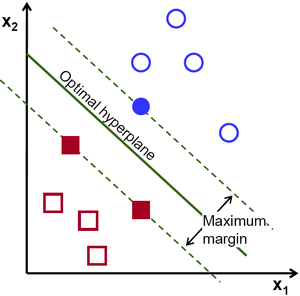
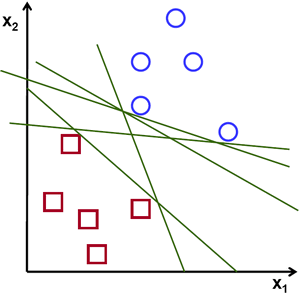
**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**ALGORITHMS**

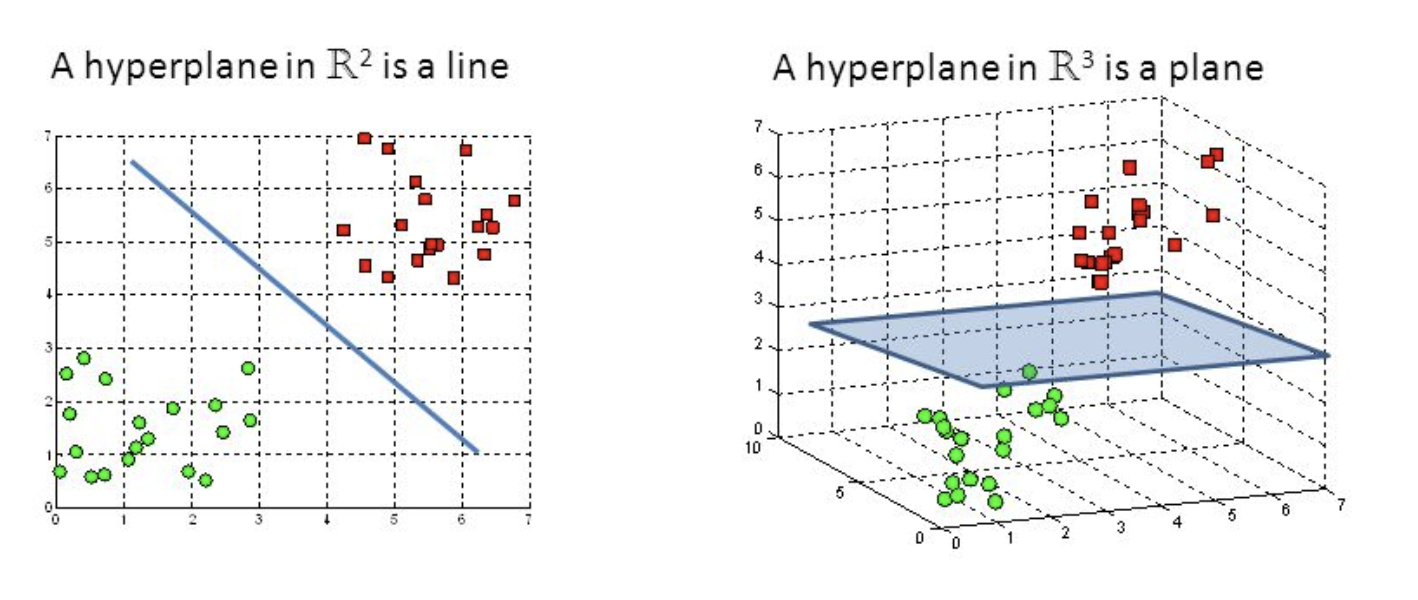
**SUPPORT VECTOR MACHINE?**

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

Possible hyperplanes

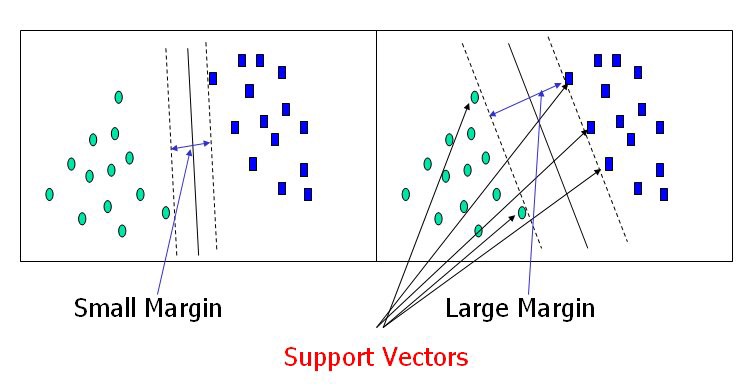
To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes and Support Vectors



Hyperplanes in 2D and 3D feature space

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.



Support Vectors

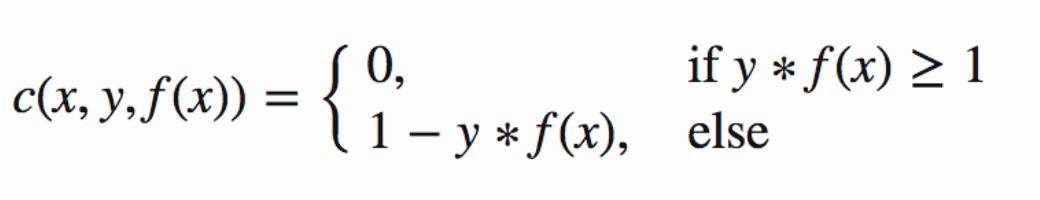
Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

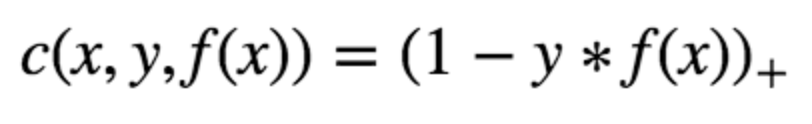
Large Margin Intuition

In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function. If the squashed value is greater than a threshold value(0.5) we assign it a label 1, else we assign it a label 0. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold values are changed to 1 and -1 in SVM, we obtain this reinforcement range of values([-1,1]) which acts as margin.

Cost Function and Gradient Updates

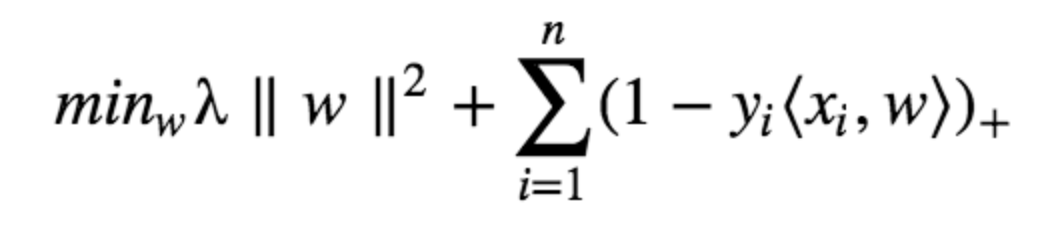
In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.





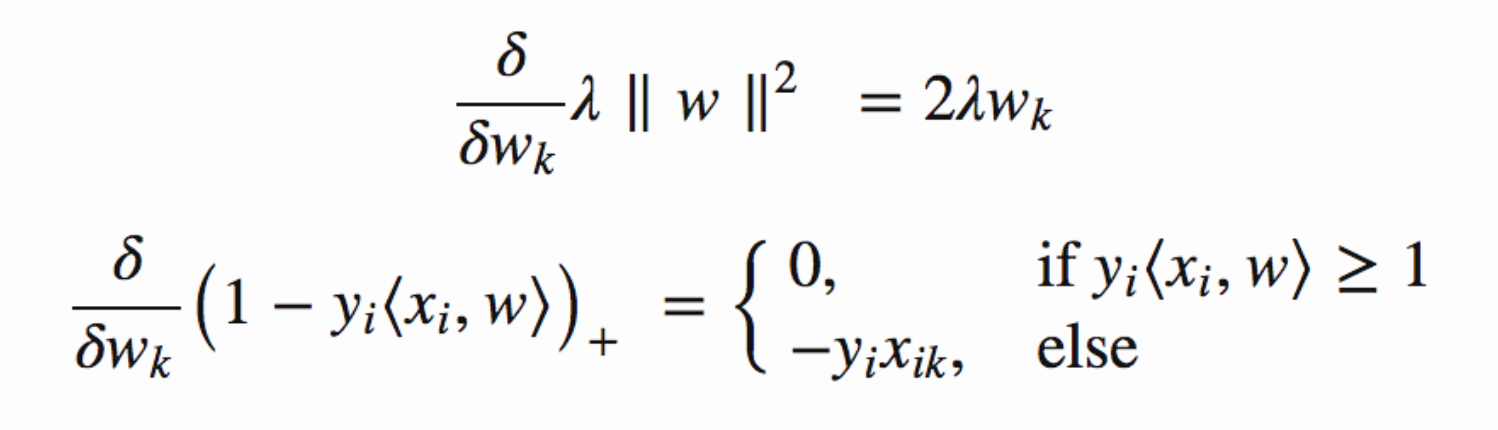
Hinge loss function (function on left can be represented as a function on the right)

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.



Loss function for SVM

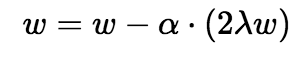
Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.



Gradients

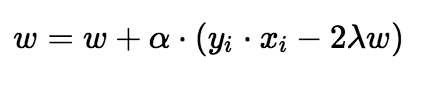
When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.





Gradient Update — No misclassification

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.



Gradient Update — Misclassification

**PYTHON**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language](https://en.wikipedia.org/wiki/Interpreted_language), Python has a design philosophy that emphasizes code [readability](https://en.wikipedia.org/wiki/Readability) (notably using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than might be used in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B)or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). [CPython](https://en.wikipedia.org/wiki/CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source) software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation). Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

**Interactive Mode Programming**

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

$ python

Python 2.4.3 (#1, Nov 11 2010, 13:34:43)

[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>

Type the following text at the Python prompt and press the Enter −

>>> print "Hello, Python!"

If you are running new version of Python, then you would need to use print statement with parenthesis as in print ("Hello, Python!");. However in Python version 2.4.3, this produces the following result −

Hello, Python!

**Script Mode Programming**

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file −

Live Demo

print "Hello, Python!"

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

$ python test.py

This produces the following result −

Hello, Python!

Let us try another way to execute a Python script. Here is the modified test.py file −

Live Demo

#!/usr/bin/python

print "Hello, Python!"

We assume that you have Python interpreter available in /usr/bin directory. Now, try to run this program as follows −

$ chmod +x test.py # This is to make file executable

$./test.py

This produces the following result −

Hello, Python!

**Python Identifiers**

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in Python.

Here are naming conventions for Python identifiers −

Class names start with an uppercase letter. All other identifiers start with a lowercase letter.

Starting an identifier with a single leading underscore indicates that the identifier is private.

Starting an identifier with two leading underscores indicates a strongly private identifier.

If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

**Reserved Words**

The following list shows the Python keywords. These are reserved words and you cannot use them as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

and exec not

assert finally or

break for pass

class from print

continue global raise

def if return

del import try

elif in while

else is with

except lambda yield

**Lines and Indentation**

Python provides no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation, which is rigidly enforced.

The number of spaces in the indentation is variable, but all statements within the block must be indented the same amount. For example −

if True:

print "True"

else:

print "False"

However, the following block generates an error −

if True:

print "Answer"

print "True"

else:

print "Answer"

print "False"

Thus, in Python all the continuous lines indented with same number of spaces would form a block. The following example has various statement blocks −

Note − Do not try to understand the logic at this point of time. Just make sure you understood various blocks even if they are without braces.

#!/usr/bin/python

import sys

try:

# open file stream

file = open(file\_name, "w")

except IOError:

print "There was an error writing to", file\_name

sys.exit()

print "Enter '", file\_finish,

print "' When finished"

while file\_text != file\_finish:

file\_text = raw\_input("Enter text: ")

if file\_text == file\_finish:

# close the file

file.close

break

file.write(file\_text)

file.write("\n")

file.close()

file\_name = raw\_input("Enter filename: ")

if len(file\_name) == 0:

print "Next time please enter something"

sys.exit()

try:

file = open(file\_name, "r")

except IOError:

print "There was an error reading file"

sys.exit()

file\_text = file.read()

file.close()

print file\_text

Multi-Line Statements

Statements in Python typically end with a new line. Python does, however, allow the use of the line continuation character (\) to denote that the line should continue. For example −

total = item\_one + \

item\_two + \

item\_three

Statements contained within the [], {}, or () brackets do not need to use the line continuation character. For example −

days = ['Monday', 'Tuesday', 'Wednesday',

'Thursday', 'Friday']

Quotation in Python

Python accepts single ('), double (") and triple (''' or """) quotes to denote string literals, as long as the same type of quote starts and ends the string.

The triple quotes are used to span the string across multiple lines. For example, all the following are legal −

word = 'word'

sentence = "This is a sentence."

paragraph = """This is a paragraph. It is

made up of multiple lines and sentences."""

Comments in Python

A hash sign (#) that is not inside a string literal begins a comment. All characters after the # and up to the end of the physical line are part of the comment and the Python interpreter ignores them.

Live Demo

#!/usr/bin/python

# First comment

print "Hello, Python!" # second comment

This produces the following result −

Hello, Python!

You can type a comment on the same line after a statement or expression −

name = "Madisetti" # This is again comment

You can comment multiple lines as follows −

# This is a comment.

# This is a comment, too.

# This is a comment, too.

# I said that already.

Following triple-quoted string is also ignored by Python interpreter and can be used as a multiline comments:

'''

This is a multiline

comment.

'''

Using Blank Lines

A line containing only whitespace, possibly with a comment, is known as a blank line and Python totally ignores it.

In an interactive interpreter session, you must enter an empty physical line to terminate a multiline statement.

Waiting for the User

The following line of the program displays the prompt, the statement saying “Press the enter key to exit”, and waits for the user to take action −

#!/usr/bin/python

raw\_input("\n\nPress the enter key to exit.")

Here, "\n\n" is used to create two new lines before displaying the actual line. Once the user presses the key, the program ends. This is a nice trick to keep a console window open until the user is done with an application.

Multiple Statements on a Single Line

The semicolon ( ; ) allows multiple statements on the single line given that neither statement starts a new code block. Here is a sample snip using the semicolon.

import sys; x = 'foo'; sys.stdout.write(x + '\n')

Multiple Statement Groups as Suites

A group of individual statements, which make a single code block are called suites in Python. Compound or complex statements, such as if, while, def, and class require a header line and a suite.

Header lines begin the statement (with the keyword) and terminate with a colon ( : ) and are followed by one or more lines which make up the suite. For example −

if expression :

suite

elif expression :

suite

else :

suite

**Command Line Arguments**

Many programs can be run to provide you with some basic information about how they should be run. Python enables you to do this with -h −

$ python -h

usage: python [option] ... [-c cmd | -m mod | file | -] [arg] ...

Options and arguments (and corresponding environment variables):

-c cmd : program passed in as string (terminates option list)

-d : debug output from parser (also PYTHONDEBUG=x)

-E : ignore environment variables (such as PYTHONPATH)

-h : print this help message and exit

You can also program your script in such a way that it should accept various options. Command Line Arguments is an advanced topic and should be studied a bit later once you have gone through rest of the Python concepts.

**Python Lists**

The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Creating a list is as simple as putting different comma-separated values between square brackets. For example −

list1 = ['physics', 'chemistry', 1997, 2000];

list2 = [1, 2, 3, 4, 5 ];

list3 = ["a", "b", "c", "d"]

Similar to string indices, list indices start at 0, and lists can be sliced, concatenated and so on.

A tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values. Optionally you can put these comma-separated values between parentheses also. For example −

tup1 = ('physics', 'chemistry', 1997, 2000);

tup2 = (1, 2, 3, 4, 5 );

tup3 = "a", "b", "c", "d";

The empty tuple is written as two parentheses containing nothing −

tup1 = ();

To write a tuple containing a single value you have to include a comma, even though there is only one value −

tup1 = (50,);

Like string indices, tuple indices start at 0, and they can be sliced, concatenated, and so on.

Accessing Values in Tuples

To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example −

Live Demo

#!/usr/bin/python

tup1 = ('physics', 'chemistry', 1997, 2000);

tup2 = (1, 2, 3, 4, 5, 6, 7 );

print "tup1[0]: ", tup1[0];

print "tup2[1:5]: ", tup2[1:5];

When the above code is executed, it produces the following result −

tup1[0]: physics

tup2[1:5]: [2, 3, 4, 5]

Updating Tuples

Accessing Values in Dictionary

To access dictionary elements, you can use the familiar square brackets along with the key to obtain its value. Following is a simple example −

Live Demo

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

print "dict['Name']: ", dict['Name']

print "dict['Age']: ", dict['Age']

When the above code is executed, it produces the following result −

dict['Name']: Zara

dict['Age']: 7

If we attempt to access a data item with a key, which is not part of the dictionary, we get an error as follows −

Live Demo

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

print "dict['Alice']: ", dict['Alice']

When the above code is executed, it produces the following result −

dict['Alice']:

Traceback (most recent call last):

File "test.py", line 4, in <module>

print "dict['Alice']: ", dict['Alice'];

KeyError: 'Alice'

Updating Dictionary

You can update a dictionary by adding a new entry or a key-value pair, modifying an existing entry, or deleting an existing entry as shown below in the simple example −

Live Demo

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

dict['Age'] = 8; # update existing entry

dict['School'] = "DPS School"; # Add new entry

print "dict['Age']: ", dict['Age']

print "dict['School']: ", dict['School']

When the above code is executed, it produces the following result −

dict['Age']: 8

dict['School']: DPS School

Delete Dictionary Elements

You can either remove individual dictionary elements or clear the entire contents of a dictionary. You can also delete entire dictionary in a single operation.

To explicitly remove an entire dictionary, just use the del statement. Following is a simple example −

Live Demo

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

del dict['Name']; # remove entry with key 'Name'

dict.clear(); # remove all entries in dict

del dict ; # delete entire dictionary

print "dict['Age']: ", dict['Age']

print "dict['School']: ", dict['School']

This produces the following result. Note that an exception is raised because after del dict dictionary does not exist any more −

dict['Age']:

Traceback (most recent call last):

File "test.py", line 8, in <module>

print "dict['Age']: ", dict['Age'];

TypeError: 'type' object is unsubscriptable

Note − del() method is discussed in subsequent section.

**Properties of Dictionary Keys**

Dictionary values have no restrictions. They can be any arbitrary Python object, either standard objects or user-defined objects. However, same is not true for the keys.

There are two important points to remember about dictionary keys −

(a) More than one entry per key not allowed. Which means no duplicate key is allowed. When duplicate keys encountered during assignment, the last assignment wins. For example −

Live Demo

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Name': 'Manni'}

print "dict['Name']: ", dict['Name']

When the above code is executed, it produces the following result −

dict['Name']: Manni

(b) Keys must be immutable. Which means you can use strings, numbers or tuples as dictionary keys but something like ['key'] is not allowed. Following is a simple example −

Live Demo

#!/usr/bin/python

dict = {['Name']: 'Zara', 'Age': 7}

print "dict['Name']: ", dict['Name']

When the above code is executed, it produces the following result −

Traceback (most recent call last):

File "test.py", line 3, in <module>

dict = {['Name']: 'Zara', 'Age': 7};

TypeError: unhashable type: 'list'

Tuples are immutable which means you cannot update or change the values of tuple elements. You are able to take portions of existing tuples to create new tuples as the following example demonstrates −

Live Demo

#!/usr/bin/python

tup1 = (12, 34.56);

tup2 = ('abc', 'xyz');

# Following action is not valid for tuples

# tup1[0] = 100;

# So let's create a new tuple as follows

tup3 = tup1 + tup2;

print tup3;

When the above code is executed, it produces the following result −

(12, 34.56, 'abc', 'xyz')

Delete Tuple Elements

Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.

To explicitly remove an entire tuple, just use the del statement. For example −

Live Demo

#!/usr/bin/python

tup = ('physics', 'chemistry', 1997, 2000);

print tup;

del tup;

print "After deleting tup : ";

print tup;

This produces the following result. Note an exception raised, this is because after del tup tuple does not exist any more −

('physics', 'chemistry', 1997, 2000)

After deleting tup :

Traceback (most recent call last):

File "test.py", line 9, in <module>

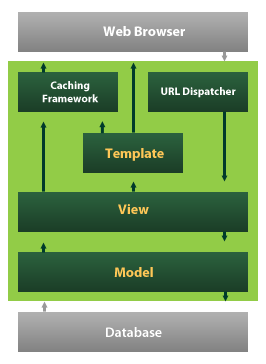
print tup;

NameError: name 'tup' is not defined

**DJANGO**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability)and "pluggability" of components, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself). Python is used throughout, even for settings files and data models.



Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)) and configured via admin models



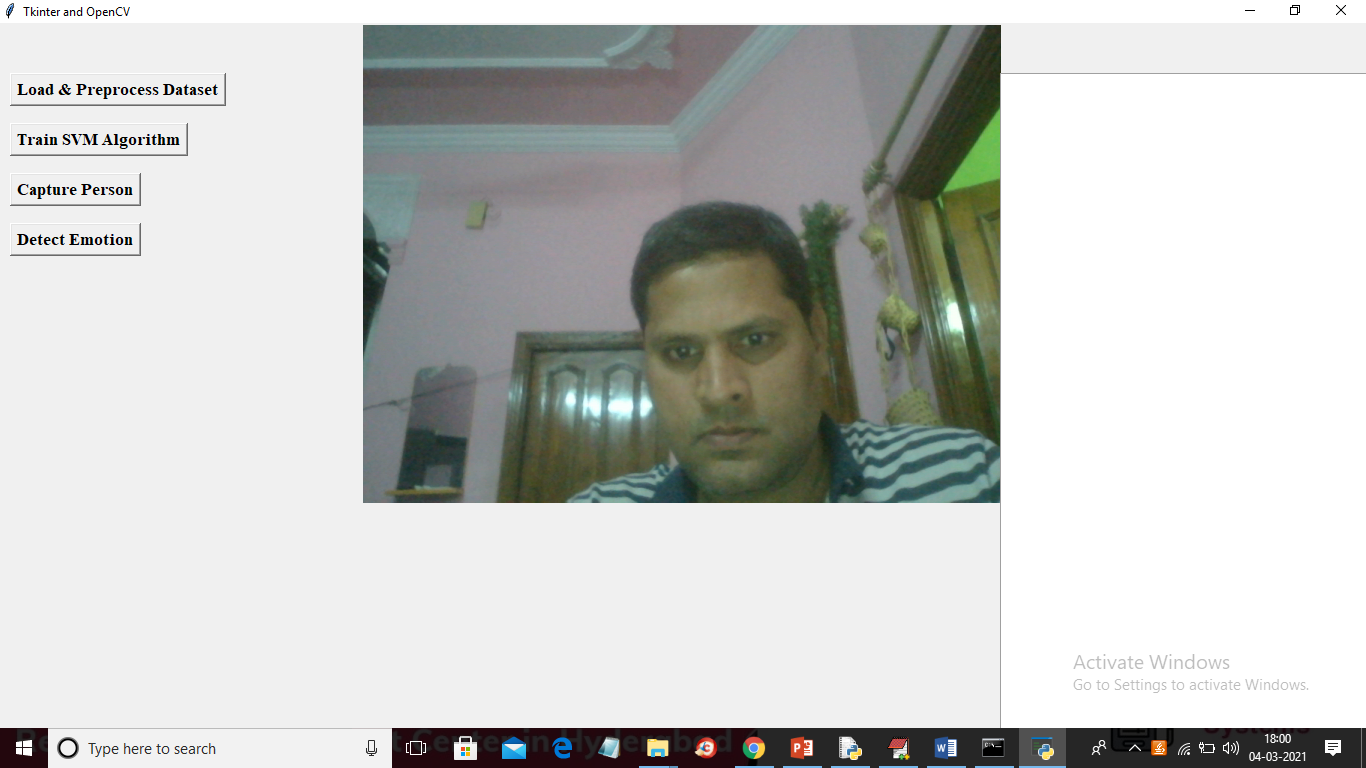
BEHAVIOUR ANALYSIS FOR MENTALLY AFFECTED PEOPLE

In this project student want to detect people’s mental behaviour by analysing there’s emotions and to implement this project we have used Facial Expression dataset and SVM machine learning algorithm. SVM is trained with facial expression dataset and then SVM trained model can be used to detect human mental behaviour. To implement this project we have designed following modules

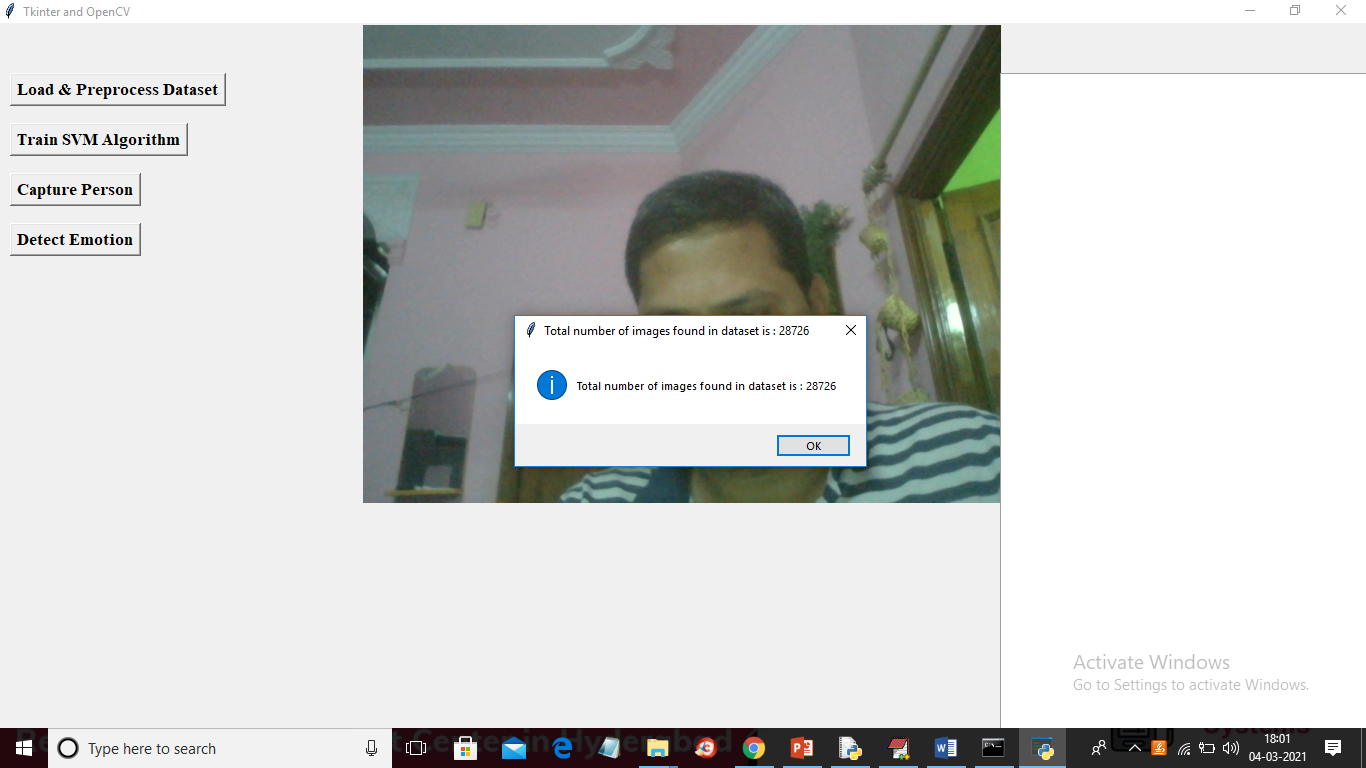
1. Webcam connection: using this module application will be connected to live webcam
2. Load & Preprocess Dataset: using this module application read all dataset images from numpy array and then normalize and extract features from images.
3. Train SVM Algorithm: Extracted features will be used to train SVM algorithm
4. Capture Person: using this module we will capture person image and then detect face from that image
5. Detect Emotion: This module will take detected face as input and then by using SVM algorithm will predict person mental behaviour as SAD, HAPPY, NEUTRAL, ANGRY etc.

SCREN SHOTS

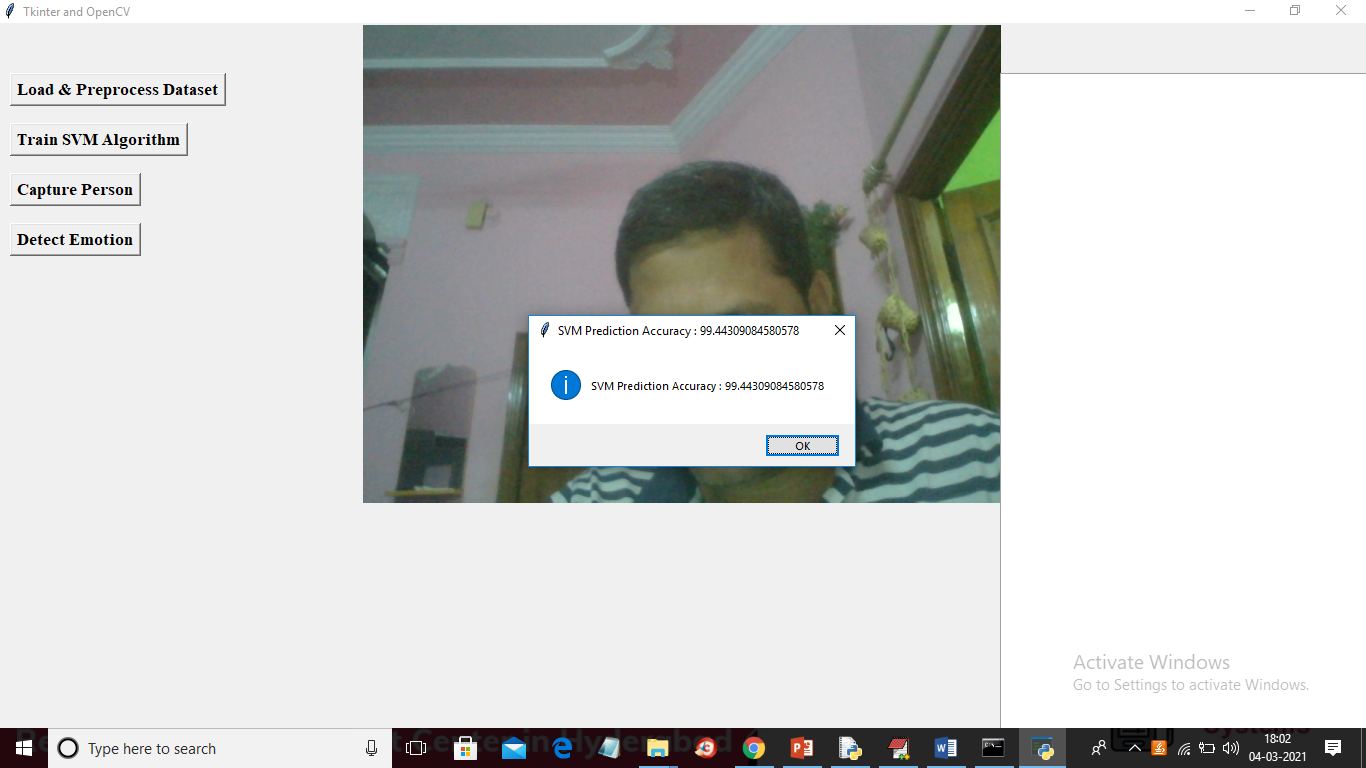
To run project double click on ‘run.bat’ file to get below screen



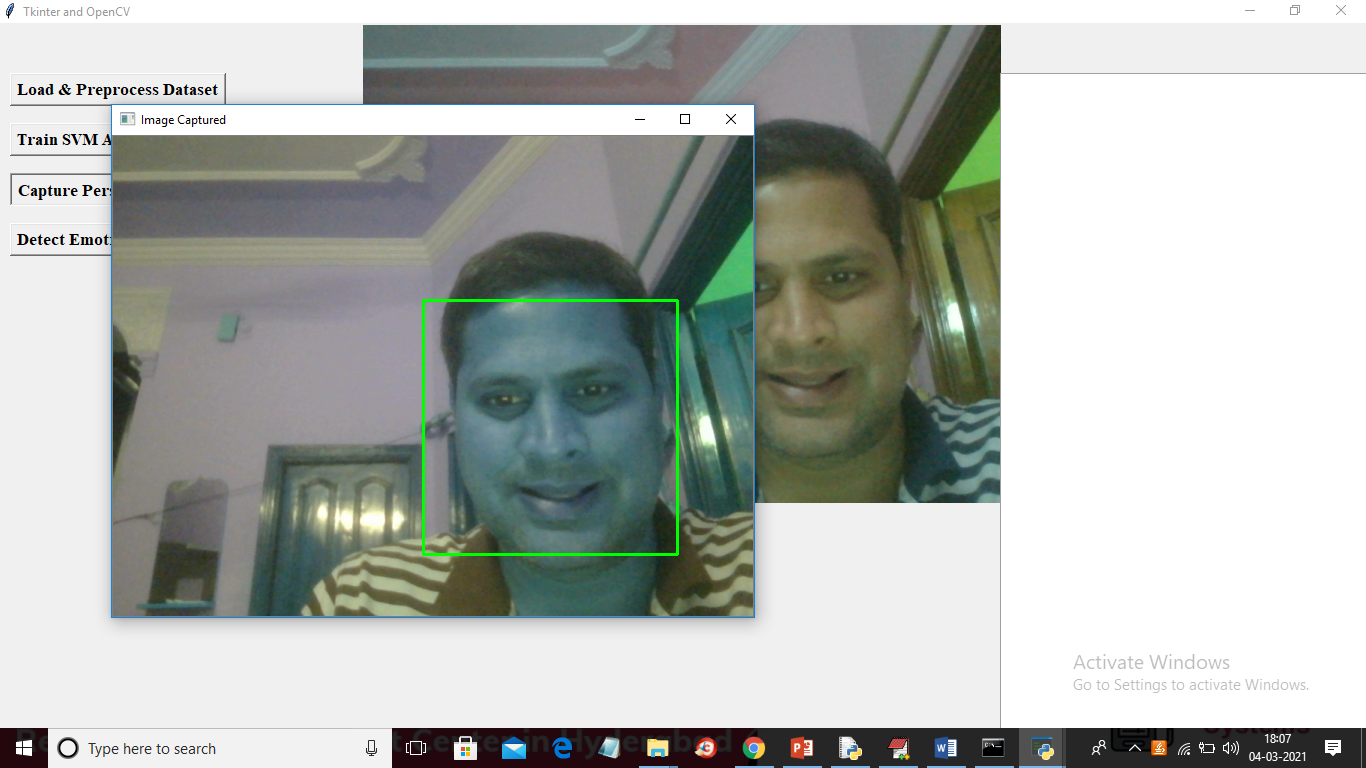
In above screen web cam started and now click on ‘Load & Preprocess Dataset’ button to read images and process them. This process may take 2 to 3 minutes of time



In above screen we can see application process 28726 images and now click on ‘Train SVM Algorithm’ button to train SVM with all those images



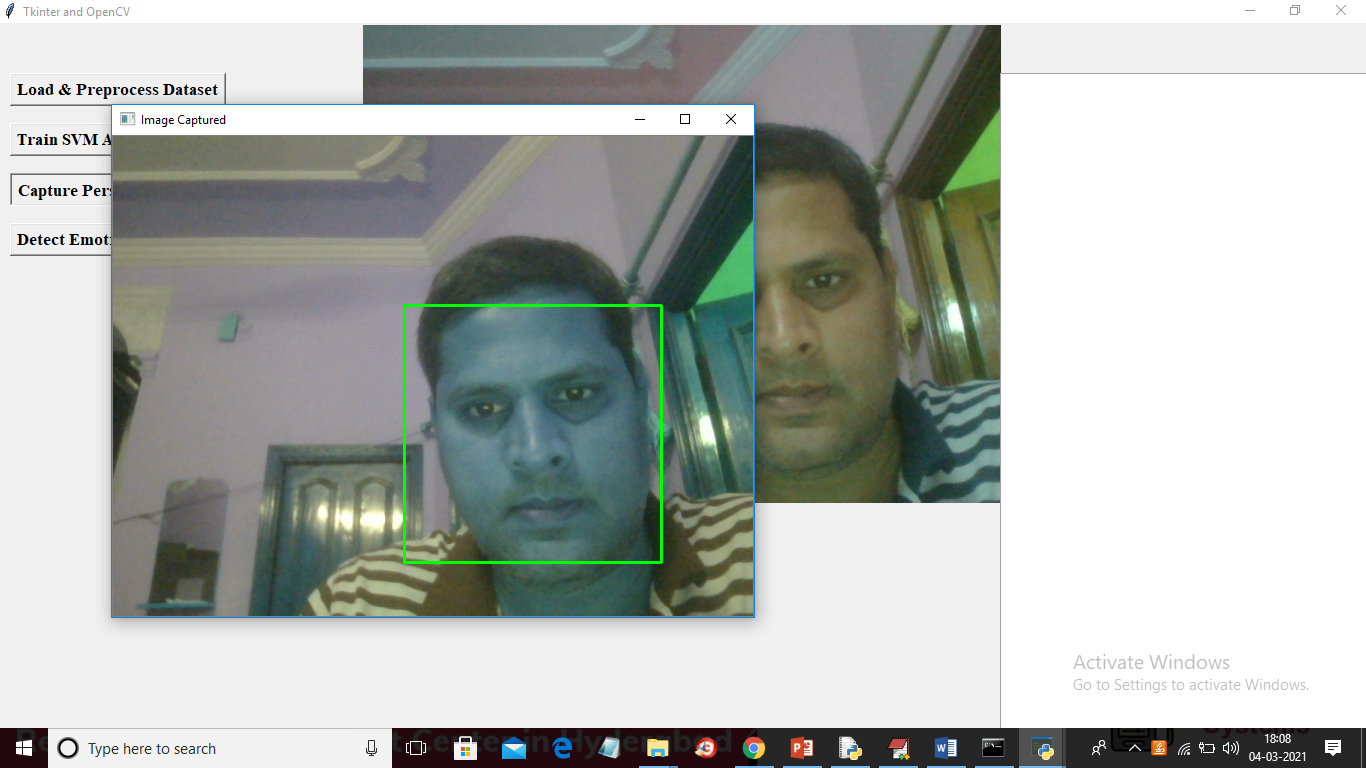
In above screen SVM model trained with prediction accuracy as 99%. Now make some expression and then click on ‘Capture Image’ button to get below screen



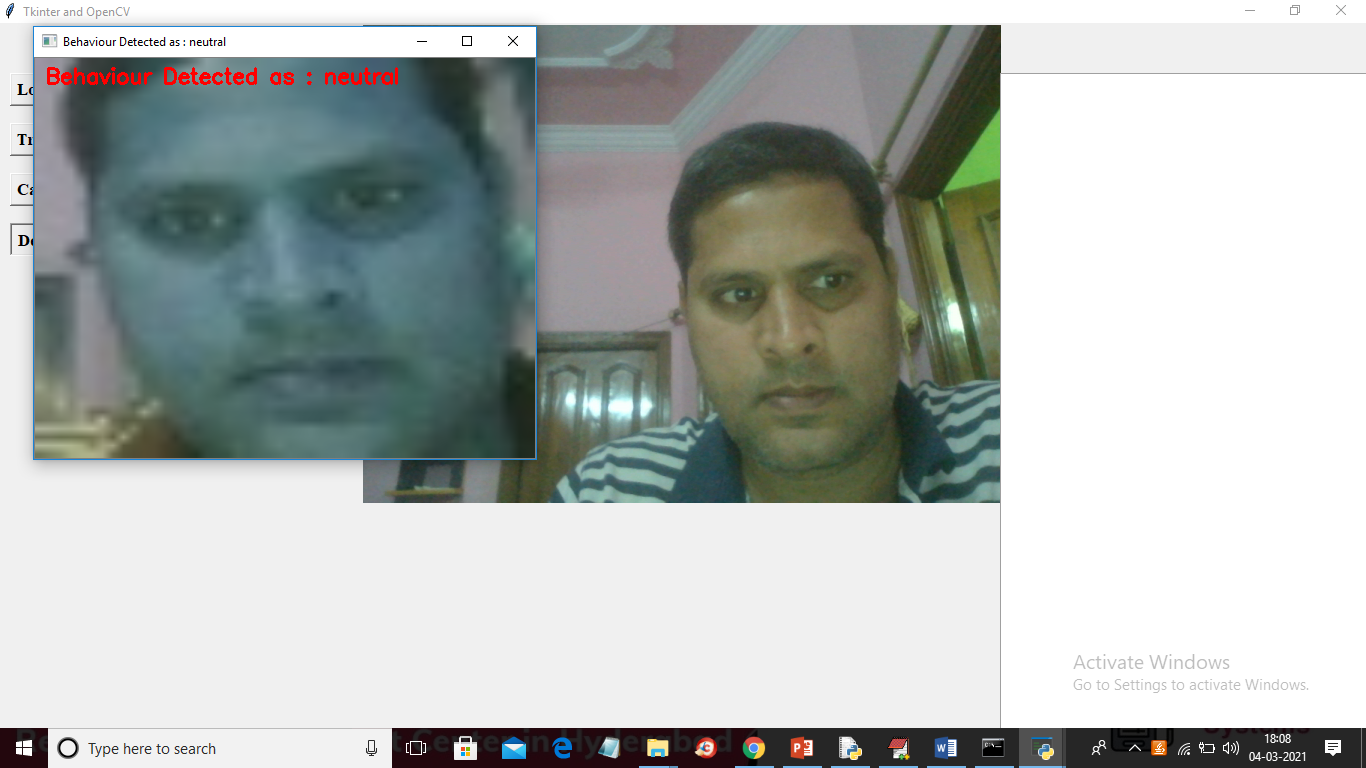
Below is the prediction result

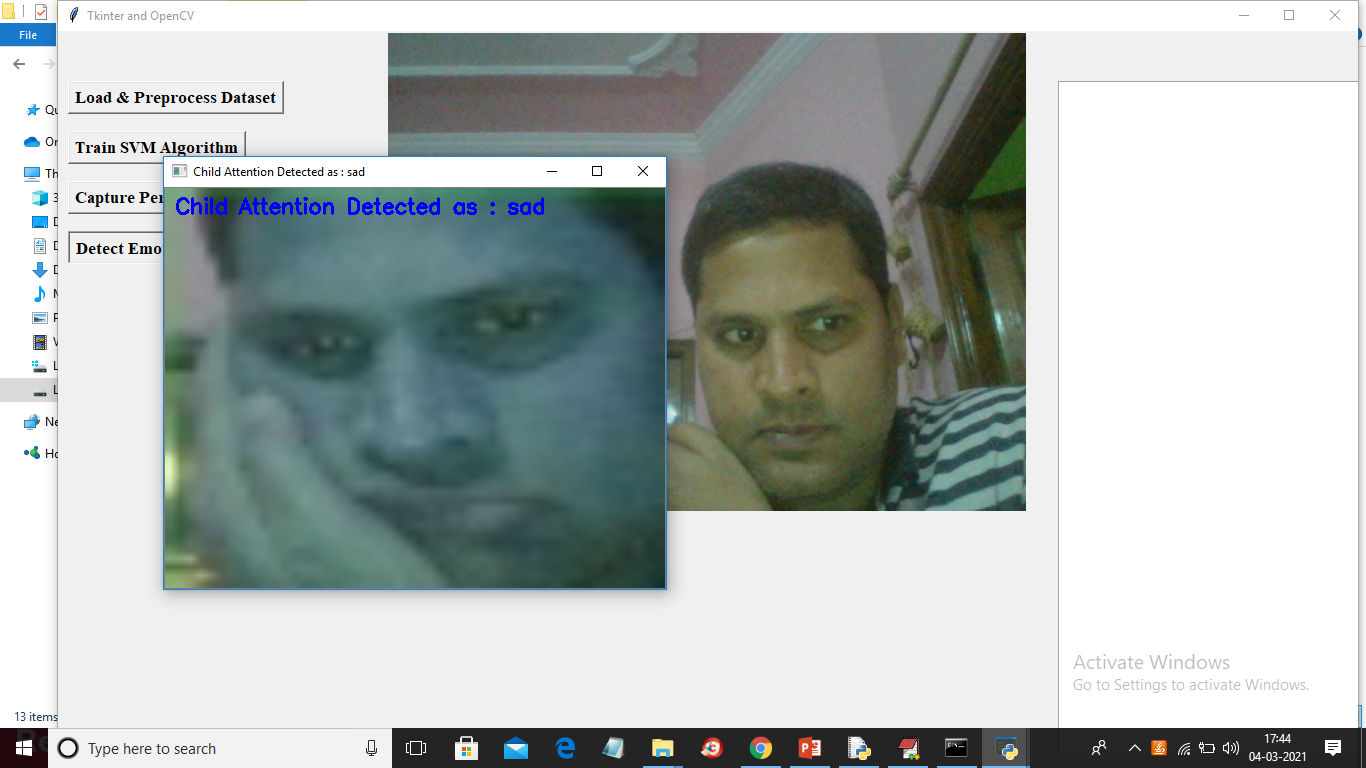


Below is next emotion

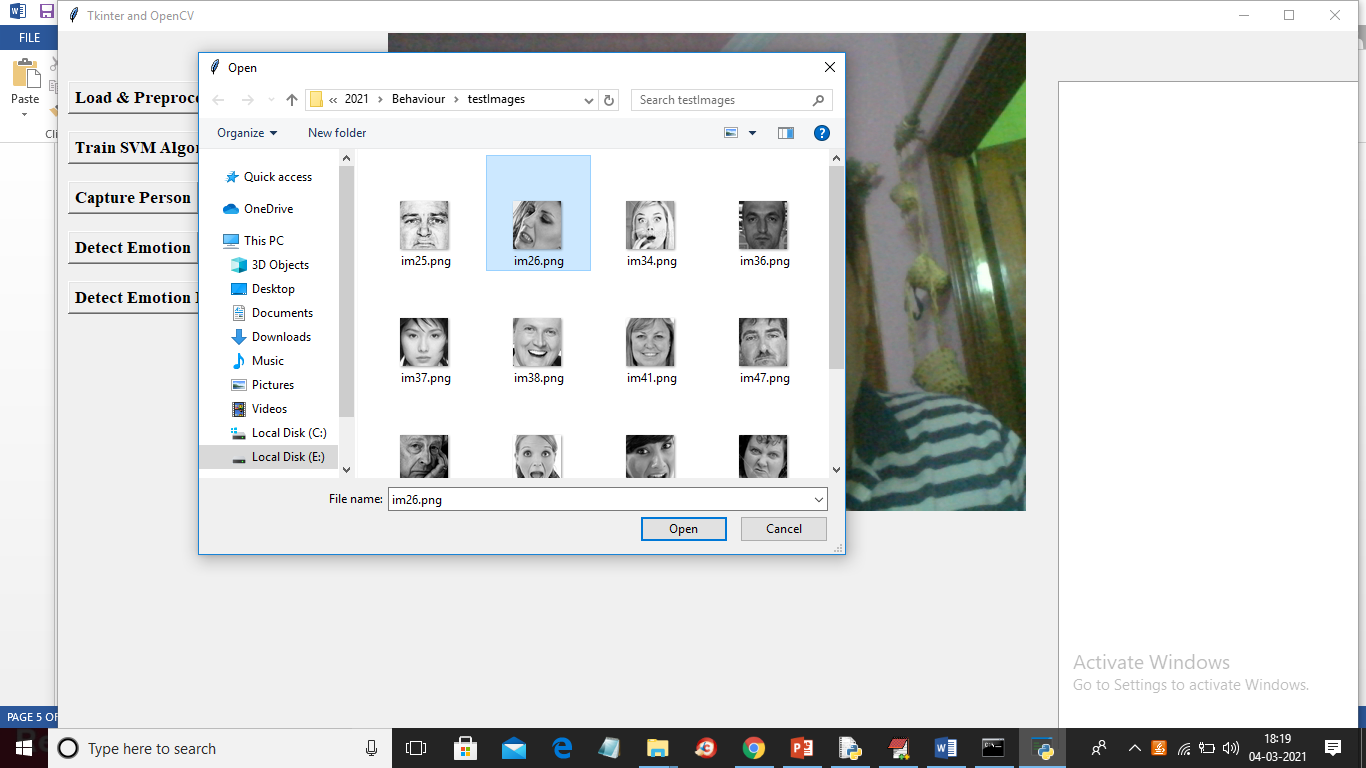


Below is the prediction result

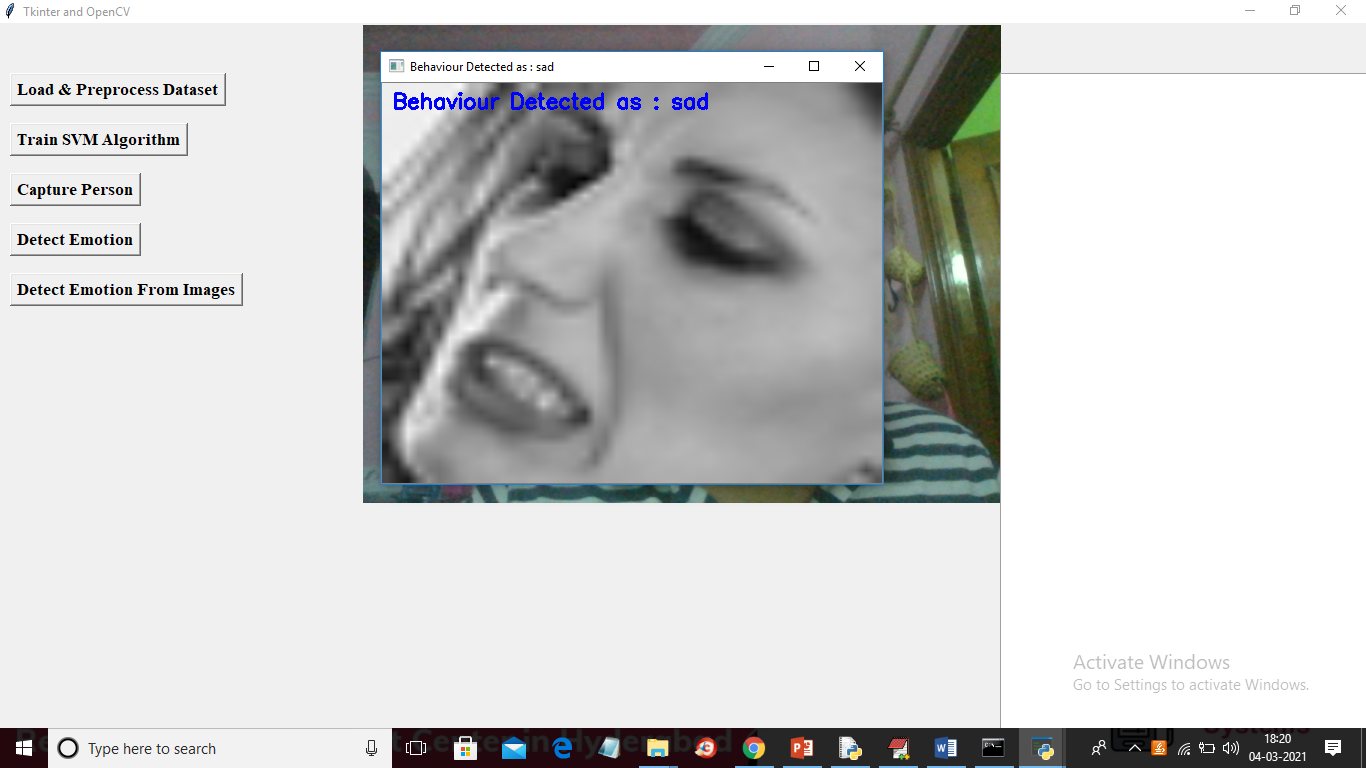


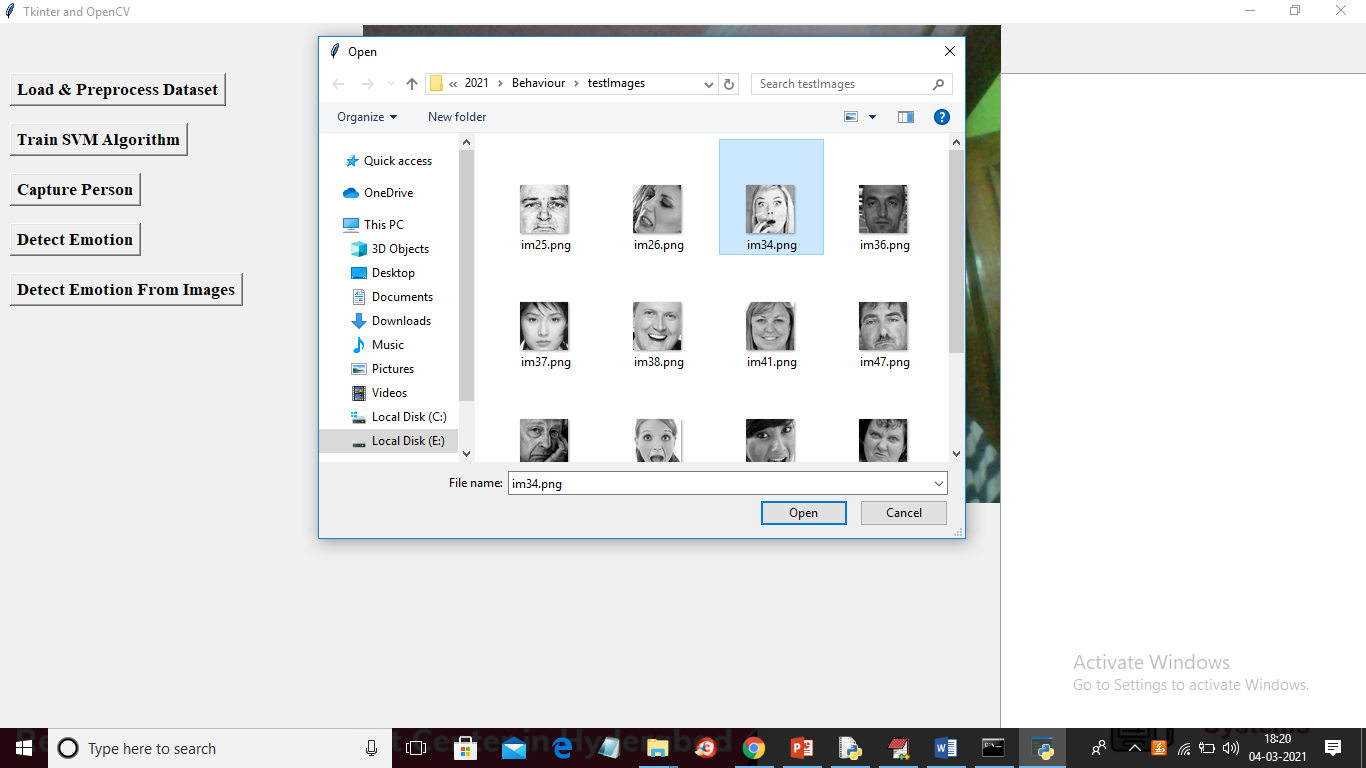


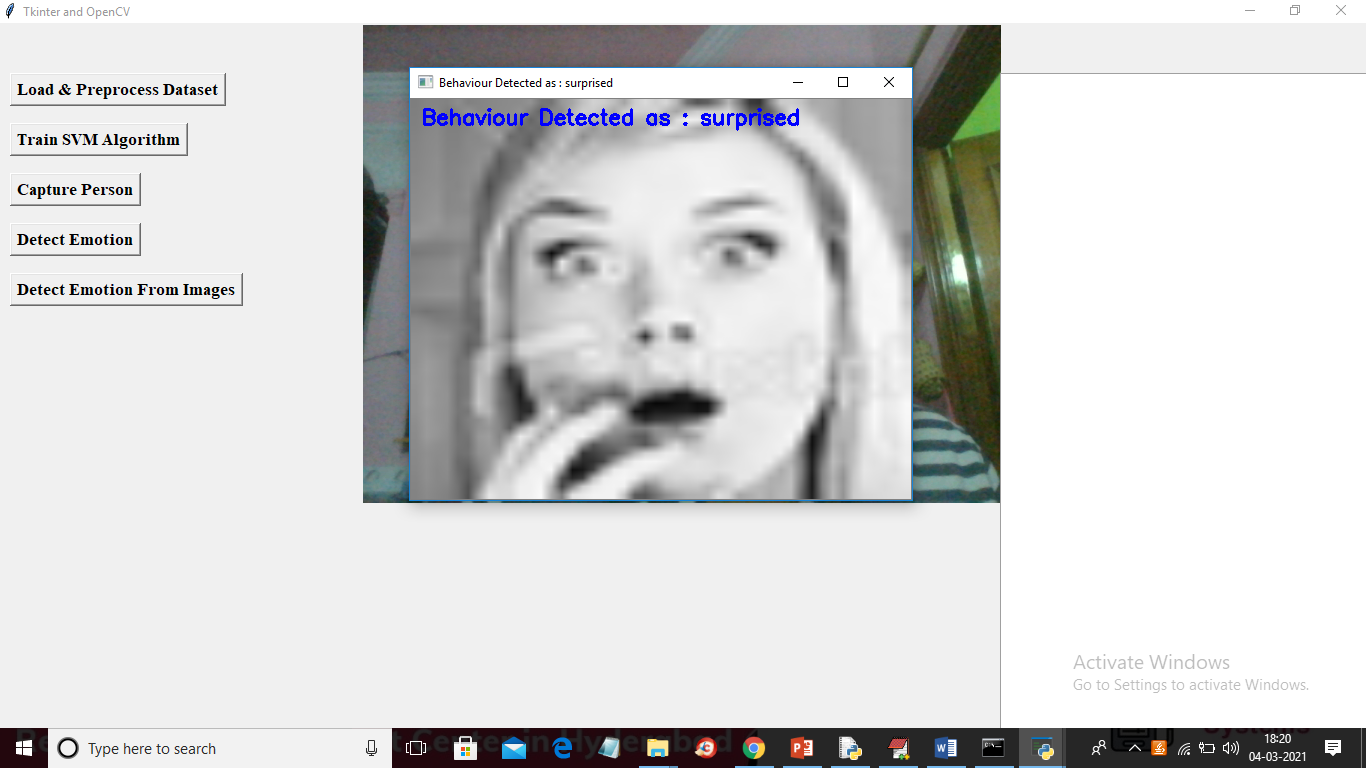
Note: we need to make proper expression to get correct result and you can try with below images also. To test from images click on ‘Detect Emotion From Images’ button and upload image



In above screen selecting and uploading im26.png file and below is the result







So in above screens application giving proper result so from webcam also we need to make exact expression to get proper result

**CONCLUSION**

Although ABA is best known for its achievements in the habilitation of persons with developmental disabilities, this approach has a long history in the treatment of severe mental disorders. Some of the first published examples of ABA were studies that increased appropriate behavior or reduced aberrant behavior of chronic mental patients in psychiatric hospitals. In recent years, ABA interventions for severe mental disorders have grown to include refined functional analyses of problem behavior as well as sophisticated verbal and self-instructional techniques (e.g., habit reversal, acceptance and commitment therapy). Refinement of assessment techniques are intended to rectify the treatment failures, lack of generalization, and poor maintenance sometimes associated with earlier behavioral interventions. Given the extensive history of successful contingency management programs for severe mental disorders and other myriad behavioral problems, it would be prudent to provide this as a treatment component and a foundation for other interventions. Some advantages of contingency management programs are that they make clear, explicit expectations of appropriate client conduct (essential for both client instructional and staff management purposes) and they provide positive reinforcement to strengthen and maintain desired client behavior. Considering the developments in functional analysis and functional assessment, it also would be wise to thoroughly investigate the function of problem behavior before attempting to eliminate it. For example, assessing the function of noncompliance may lead to greater adherence to outpatient medication regimens. If the client's circumstances permit conducting a full functional analysis, this would be the most conclusive method of ascertaining its meaning or the specific environmental stimuli which predict and maintaining the problem behavior. Lacking a functional analysis, a FBA can uncover valuable information suggesting treatment procedures that properly take into account the client's motives. While working with heterogeneous mental disorders ABA has remained a vital and imiovative scientific approach. Although most ABA researchers continue to focus on observable, socially relevant responses, current applications of ABA in mental health involve a broader realm of clinical phenomena that has required conceptual and methodological expansions. ABA practitioners now attend to internal and covert processes during assessment and intervention, as in their use of self-report measures of anxiety and imaginary exercises aimed at changing disturbing thoughts. These radical changes call into question some of the fundamental principles of ABA, and only future research will determine whether these departures represent evolutionary advances of the field. Continued research on integrated models is warranted and will further strengthen the use of ABA within mental health while concurrently providing more efficacious therapies. Although integration of treatment approaches is not completely straightforward because therapies are based on different assumptions and their procedures can be dissimilar, the use of ACT demonstrates how behavior analysis can merge with relational frame theory to create a vibrant approach to mental health issues. The case study of Lucy illustrates how functional assessment can compliment cognitive behavior therapy (CBT). Although challenges in combining varied ABA approaches and clinical practices within mental health are inevitable, the approaches may complement one another raising the likelihood of producing positive and lasting outcomes. Research reviewed in this article demonstrated that integration of behavior analysis within mental health services will expand clinicians' armamentarium and provide more comprehensive assessment and treatment. Behavior analytic techniques, such as FBA can be used to identify environmental stimuli that set the occasion for, elicit, or reinforce problematic behaviors. FBA could also reveal social-environmental variables underlying somatic disorders (as in the case of Lucy described earlier) or somatic variables underlying behavioral disorders (as in the case of the woman with a presumed pelvic infection), thereby facilitating multidisciplinary collaboration and selection of appropriate treatments. Retaining the technology that produced early successes, ABA provides empirically validated instructional procedures and practical methods for engineering a client's environment to promote adaptive behavior (e.g., parent and staff training, token programs) that no other clinical discipline offers.

**REFERENCES**

* American Psychiatric Association. (2000). Diagnostic and statistical manual of mental disorders (4th ed., text revision). Washington, DC: Author.
* Ayllon, T. (1963). Intensive treatment of psychotic behavior by stimulus satiation and food reinforcement. Behaviour Research and Therapy, 1,53-61.
* Ayllon, T., & Azrin, N. H. (1965). The measurement and reinforcement of behavior of psychotics. J OLlrnal of the Experimental Analysis of Behavior, 8, 357-383.
* Ayllon, T., & Azrin, N. H. (1968). The token economy: A motivation systemfor therapy and rehabilitation. Englewood Cliffs, NJ: Prentice Hall.
* Ayllon, T., & Haughton, E. (1964). Modification of symptomatic verbal behavior for mental patients. Behavior Research and Therapy, 2, 87-97.
* Ayllon, T., & Michael, J. (1959). The psychiatric nurse as a behavioral engineer. Journal of the Experimental Analysis of Behavior, 2, 323-334.
* Bach, P. (2004). ACT with the seriously mentally ill.
* In S. C. Hayes, & K. D. Strosahl (Eds.), A practical guide to Acceptance and Commitment Therapy (pp. 185-208). New York: Springer.
* Bach, P., & Hayes, S. C. (2002). The use of acceptance and commitment therapy to prevent the rehospitalization of psychotic patients: A randomized controlled trial. Journal of Consulting and Clinical Psychology, 70, 1129-1139.
* Bach, P. A, Gaudiano, B., Pankey, J., Herbert, J. D., & Hayes, S. C. (2006). Acceptance, mindfulness, values, and psychosis: Applying acceptance and commitment therapy (ACT) to the chronically mentally ill.
* In R. A Baer (Ed.), Mindfulnessbased treatment approaches: Clinician's guide to evidence base and applications (pp. 93-116). Burlington, MA: Academic Press.