**STRESS DETECTION IN IT PROFESSIONAL BY IMAGE PROCESSING**

**1.1. AIM:-**

The aim of the project is to detect the faces and get person behaviors from the human beings using video streaming. Using this we can identify the person behaviors with the file supporting to detect stress.

## 1.2 OBJECTIVE:-

The overall objective is to develop an automated Face based Person behaviors system for detecting stress comprising of a desktop application working in conjunction with deep learning.

**ABSTRACT:-**

The main conception of this paper is to descry stress in the IT professionals with the help of Deep literacy and Image processing ways. This paper is an upgraded interpretation of the old stress discovery systems which barred the live discovery and the particular comforting but this paper comprises of live discovery and periodic analysis of workers and detecting physical as well as internal stress situations in his/ her by furnishing them with proper remedies for managing stress by furnishing check form periodically. This paper substantially focuses on managing stress and making the working terrain healthy and robotic for the workers and to get the stylish out of them during working hours.in this when we detected different type behaviors of human being the person is in abnormal behavior like stress mood then we can control the person with smooth music.

**CHAPTER 1:**

# 1. INTRODUCTION:-

With the advent of modern technology our desires went high and it binds no bounds. In the present era a huge research work is going on in the field of digital image and image processing. The way of progression has been exponential and it is ever increasing. Image Processing is a vast area of research in present day world and its applications are very widespread. Image processing is the field of signal processing where both the input and output signals are images. One of the most important applications of Image processing is Facial expression recognition. Our person behavior is revealed by the expressions in our face. Facial Expressions plays an important role in interpersonal communication. Facial expression is a non verbal scientific gesture which gets expressed in our face based as per our person behaviors. Automatic recognition of facial expression plays an important role in artificial intelligence and robotics and thus it is a need of the generation. Some application related to this includes Personal identification and Access control, Videophone and Teleconferencing, Forensic application, Human-Computer Interaction, Automated Surveillance, Cosmetology and so on. The objective of this project is to develop Automatic Facial Expression Recognition System which can take human facial images containing some expression as input and recognize and classify it into seven different expression. We should the find stress on it professionals.

**Scope of the project:**

The main scope of the project is to detect the human facial expression as image and then image will do the process the filtering which gave human face emotions and then goes on to discrete Wavelet transform is used to compress the image and train the image in neural network .image quality plays an important role in this process. In this we can capture the face image as a real time, what human can express in face. Face emotion detection is used to predict the emotion state of the person based on their face expressions. After that based on stress detection we should give some control options.

**CHAPTER 2:**

**LITERATURE SURVEY**

| **SNO** | **TITLE** | **AUTHOR** | **YEAR** |
| --- | --- | --- | --- |
| **1** | A review of facial  Expression using  Artificial intelligence. | mohammed  mansoor | 2013 |
| **2** | Face recognition  Techniques for  Forsenic identify | Jinhua zeng | **2017** |
| **3** | Comparision of  Four subjective methods for image quality | R.mantiruk | **2012** |
| **4** | Objective  Assesment of  Image quality | r.f.wanger | 2020 |

**paper1:**

**title:** **Behavior Recognition of a Person in a Daily Video Using Joint Position Information**

**author:** **Jeong-Min Seo; Haanju Yoo; Kimin Yun; Hyunil Kim**

**abstract:** In this paper, we propose a method to recognize people's specific behaviors in images acquired in everyday life. The poses that occur when taking a specific action are represented using the joint coordinates of the body. To reduce the influence of the perspective effect on the locations in an image, the coordinates of joints for each person were normalized and the pose was classified using SVM with the normalized coordinates. Experimental results on video images of CCTV camera on ordinary alleys showed that the proposed method relatively recognized a specific behavior.

**keywords:** **Behavior recognition , SVM , joint position**

**paper2:**

**title:** **A Study and Estimation a Lost Person Behavior in Crowded Areas Using Accelerometer Data from Smartphones**

**author:** **Mohammed Balfas; Sheikh Iqbal Ahamed; Chandana Tamma; Muhammad Arif; Ahmed J. Kattan**

**abstract:** As smartphones become more popular, applications are being developed with new and innovative ways to solve problems in the day-to-day lives of users. One area of smartphone technology that has been developed in recent years is human activity recognition (HAR). This technology uses various sensors that are built into the smartphone to sense a person's activity in real time. Applications that incorporate HAR can be used to track a person's movements and are very useful in areas such as health care. We use this type of motion sensing technology, specifically, using data collected from the accelerometer sensor. The purpose of this study is to study and estimate the person who may become lost in a crowded area. The application is capable of estimating the movements of people in a crowded area, and whether or not the person is lost in a crowded area based on his/her movements as detected by the smartphone. This will be a great benefit to anyone interested in crowd management strategies. In this paper, we review related literature and research that has given us the basis for our own research. We also detail research on lost person behavior. We looked at the typical movements a person will likely make when he/she is lost and used these movements to indicate lost person behavior. We then evaluate and describe the creation of the application, all of its components, and the testing process.

**keywords:** **Human Activity Recognition (HAR), Lost Person Behavior and Psychology, Inertial Tracking, Mobile Computing, Ubiquitous Computing, Hajj Smartphones**

**paper3:**

**title:** **Natural person-following behavior for social robots**

**author:** **Rachel Gockley; Jodi Forlizzi; Reid Simmons**

**abstract:** We are developing robots with socially appropriate spatial skills not only to travel around or near people, but also to accompany people side-by-side. As a step toward this goal, we are investigating the social perceptions of a robot's movement as it follows behind a person. This paper discusses our laser-based person-tracking method and two different approaches to person-following: direction-following and path-following. While both algorithms have similar characteristics in terms of tracking performance and following distances, participants in a pilot study rated the direction-following behavior as significantly more human-like and natural than the path-following behavior. We argue that the path-following method may still be more appropriate in some situations, and we propose that the ideal person-following behavior may be a hybrid approach, with the robot automatically selecting which method to use.

**keywords:** **Human-robot interaction , social robots , person tracking , person following**

**paper4:**

**title:** **SmarTV - a multi-person user's behavior analysis and program recommendation system with iTV**

**author:** **Aislan Gomide Foina**

**abstract:** This paper presents a research in progress about a system for multiple person user's behavior analysis and program recommendation for digital TV. Since it uses the spectator cell phone to identify the person, this research proposes an extension of the interactive TV interface based to the user's cell phone as well. The research explore two new approaches related to how TV is watched, using a Bluetooth cell phone and a Bluetooth set-top box connected to an digital TV network with return channel. The first approach is related to user's behavior analysis and program recommendation. Supposing that every viewer has Bluetooth cell phone, each cell phone will identify every spectator to the set-top box wirelessly. The set-top box will inform the TV service provider the list of people who are in front of TV in real-time. Based on the history of programs watched by these people, it will be identified their profile and preferences. The second approach is the use of cell phone as a smart remote control and as an extension of the TV. The cell phone connected to the set-top box can be used by the interactive TV programs to show messages to the viewer or execute interactive applications.

**keywords:** **Bluetooth , Cell Phone , Interactive TV , Program Recommendation , RFID , User Behavior**

**paper5:**

**title:** **Visualization Based on Person Move Similarity for Person Behavior Analysis**

**author: Hidekazu Yanagimoto; Kiyota Hashimoto; Tokuro Matsuo**

**abstract:** This paper proposes new visualization based on distance between persons, which are defined with Normalized Restricted Dynamic Time Warping. In the visualization persons are located based on their distances that are defined by their movement histories with t-SNE. In experiments we visualize academic conference participants with our proposed method and confirm that the proposed method helps us to find participants moving together.

**keywords:** **indoor positioning estimation, visualization person behavior analysis**

**Chapter 3:**

**SYSTEM ANALYSIS**

**Existing system**

The study of face based and its features is an active research area from past few decades. Pose variation, illumination conditions, bad lighting etc., are still challenging factors faced by all algorithms. Face based recognition and person behavior detection system are the major applications of recognition system, in which many algorithms have tried to solve these problems.

* Principal Component Analysis
* Geometric methods
* Support vector machine.

**Drawbacks**

* Low discriminatory power and high computational load
* In geometric based methods, the geometric features like distance between speech signals.

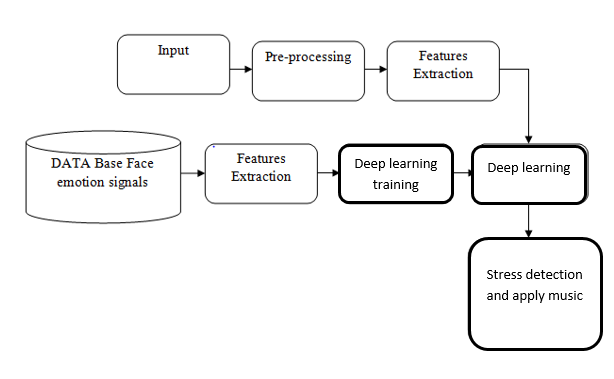
**Proposed method**

* Pre-process
* Deep learning
* Human Behavior model
* Feature extraction

**ADVANTAGE:**

* Can relief the people from abnormal situations
* More accurate results
* Training of features are easy

**BLOCK DIAGRAM**



**Chapter 5**

**PREPROCESSING**

* Image Pre-processing is a common name for operations with images at the lowest level of abstraction. Its input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.
* Image restoration is the operation of taking a corrupted/noisy image and estimating the clean original image. Corruption may come in many forms such as motion blur, noise, and camera misfocus. Image restoration is different from image enhancement in that the latter is designed to emphasize features of the image that make the image more pleasing to the observer, but not necessarily to produce realistic data from a scientific point of view. Image enhancement techniques (like contrast stretching or de-blurring by a nearest neighbor procedure) provided by "Imaging packages" use no a priori model of the process that created the image. With image enhancement noise can be effectively be removed by sacrificing some resolution, but this is not acceptable in many applications. In a Fluorescence Microscope resolution in the z-direction is bad as it is. More advanced image processing techniques must be applied to recover the object. De-Convolution is an example of image restoration method. It is capable of: Increasing resolution, especially in the axial direction removing noise increasing contrast.

**Haar cascade classifier**

**Haar – Cascades**

Haar- like features are  rectangular patterns in data. A cascade is a series of “Haar-like features” that are combined to form a classifier [14]. A Haar wavelet is a mathematical function that produces square wave output.

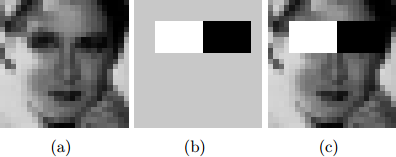


Figure 2-2.  Haar like Features [13]

Figure 2.2 shows Haar like features, the background of a template like (b) is painted gray to highlight the pattern’s support. Only those pixels marked in black or white are used when the corresponding feature is calculated [15].

Since no objective distribution can describe the actual prior probability for a given image to have a face, the algorithm must minimize both the false negative and false positive rates in order to achieve an acceptable performance [16]. This then requires an accurate numerical description of what sets human faces apart from other objects. Characteristics that define a face can be extracted from the images with a remarkable committee learning algorithm called Adaboost [17]. Adaboost (Adaptive boost) relies on a committee of weak classifiers that combine to form a strong one through a voting mechanism [18]. A classifier is weak if, in general, it cannot meet a predefined classification target in error terms [7]. The operational algorithm to be used must also work with a reasonable computational budget. Such techniques as the integral image and attention cascades have made the Viola-Jones algorithm [15] highly efficient: fed with a real time image sequence generated from a standard webcam or camera, it performs well on a standard PC.

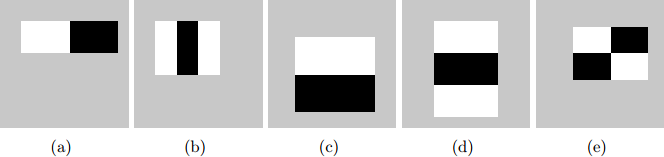


Figure 2-3. Haar-like features with different sizes and orientation [13]

The size and position of a pattern’s support can vary provided its black and white rectangles have the same dimension, border each other and keep their relative positions. Thanks to this constraint, the number of features one can draw from an image is somewhat manageable: a 24 × 24 image, for instance, has 43200, 27600, 43200, 27600 and 20736 features of category (a), (b), (c), (d) and (e) respectively as shown in figure 2.3, hence 162336 features in all[13]. In practice, five patterns are considered. The derived features

are assumed to hold all the information needed to characterize a face. Since faces are large and regular by nature, the use of Haar-like patterns.

## How The HAAR – Like Features Work

A scale is chosen for the features say 24 × 24 pixels. This is then slid across the image. The average pixel values under the white area and the black area are then computed. If the difference between the areas is above some threshold then the feature matches [7].

In face detection, since the eyes are of different color tone from the nose, the Haar feature (b) from Figure 2.3 can be scaled to fit that area as shown below,



Figure 2-4. How the Haar like feature of figure 2.3 can be used to scale the eyes

One Haar feature is however not enough as there are several features that could match it (like the zip drive and white areas at the background of the image of figure 2.4 it is called a “weak classifier.” Haar cascades, the basis of Viola Jones detection framework

[16] therefore consist of a series of weak classifiers whose accuracy is at least 50% correct. If an area passes a single classifier, it moves to the next weak classifier and so on, otherwise, the area does not match.

1. **Cascaded Classifier**

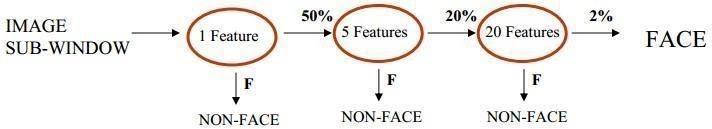


Figure 2-5. several classifiers combined to enhance face detection

From figure 2.5, a 1 feature classifier achieves 100% face detection rate and about 50% false positive rate. A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative). A 20 feature classifier achieves 100% detection rate with 10% false positive rate (2% cumulative)[17].Combining several weak classifiers improves the accuracy of detection.

A training algorithm called Adaboost, short for adaptive boosting [14], which had no application before Haar cascades [14], was utilized to combine a series of weak classifiers in to a strong classifier. Adaboost tries out multiple weak classifiers over several rounds, selecting the best weak classifier in each round and combining the best weak classifier to create a strong classifier [7]. Adaboost can use classifiers that are consistently wrong by reversing their decision [7]. In the design and development, it can take weeks of processing time to determine the final cascade sequence [18].

After the final cascade had been constructed, there was a need for a way to quickly compute the Haar features i.e. compute the differences in the two areas. The integral image was instrumental in this.

#### Integral Image

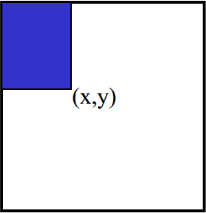


Figure 2-6. Pixel Coordinates of an integral image

The Integral image also known as the “summed area table” developed in 1984 came in to widespread use in 2001 with the Haar cascades [4]. A summed area table is created in a single pass. This makes the Haar cascades fast, since the sum of any region in the image can be computed using a single formula [17].

The integral image computes a value at each pixel (x, y) as is shown in figure 2.6, that is the sum of the pixel values above and to the left of (x, y), inclusive. This can quickly be computed in one pass through the image.

Let A, B, C D be the values of the integral image at the corners of a rectangle as shown in figure 2.7.

The sum of original image values within the rectangle can be computed.

𝑆𝑢= 𝐴 − 𝐵 − 𝐶 + 𝐷 - (2.1)

Only three additions are required for any size of rectangle[17]. This face detection approach minimizes computation time while achieving high detection accuracy[15]. It is now used in many areas of computer vision [4] [7].

Improving Face Detection

Face detection can be improved by tuning the detectors parameters to yield satisfactory results. The parameters to be adjusted are explained as follows.

#### Scale Increase Rate.

The scale increase rate specifies how quickly the face detector function should increase the scale for face detection with each pass it makes over an image. Setting the scale increase rate high makes the detector run faster by running fewer passes. If it is set too high it may jump quickly between the scales and miss the faces. The default increase rate in Open CV is 1.1. This implies that the scale increases by a factor of 10 % each pass .The parameters assume a value of 1.1, 1.2, 1.3 or 1.4.

**Minimum Neighbors Threshold**

The minimum neighbor’s threshold sets the cutoff level for discarding or keeping rectangle groups as either faces or not. This is based on the number of raw detections in the group and its values ranges from zero to four.

When the face detector is called behind the scenes, each positive face region generates many hits from the Haar detector as in Figure 2.8. The face region itself generates a large cluster of rectangles that to a large extend overlap. The isolated detections are usually false detections and are discarded. The multiple face region detections are then merged in to a single detection. The face detection function does all this before returning the list of the detected faces. The merge step groups rectangles that contain a large number of overlaps and then finds the average rectangle.

**Chapter 6**

**DIGITAL IMAGE PROCESSING**

The identification of objects in an image would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures.

The clever bit is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. One reason this is an AI problem is that an object can appear very different when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously but a computer requires skillful programming and lots of processing power to approach human performance. Manipulating data in the form of an image through several possible techniques. An image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen. An image can be processed optically or digitally with a computer.

To digitally process an image, it is first necessary to reduce the image to a series of numbers that can be manipulated by the computer. Each number representing the brightness value of the image at a particular location is called a picture element, or pixel. A typical digitized image may have 512 × 512 or roughly 250,000 pixels, although much larger images are becoming common. Once the image has been digitized, there are three basic operations that can be performed on it in the computer. For a point operation, a pixel value in the output image depends on a single pixel value in the input image. For local operations, several neighbouring pixels in the input image determine the value of an output image pixel. In a global operation, all of the input image pixels contribute to an output image pixel value.

These operations, taken [singly](http://www.answers.com/topic/singly) or in combination, are the means by which the image is enhanced, restored, or compressed. An image is enhanced when it is modified so that the information it contains is more clearly evident, but enhancement can also include making the image more visually appealing.

An example is noise smoothing. To smooth a [noisy](http://www.answers.com/topic/noisy) image, median filtering can be applied with a 3 × 3 pixel window. This means that the value of every pixel in the noisy image is recorded, along with the values of its nearest eight neighbours. These nine numbers are then ordered according to size, and the median is selected as the value for the pixel in the new image. As the 3 × 3 window is moved one pixel at a time across the noisy image, the filtered image is formed.

Another example of enhancement is contrast manipulation, where each pixel's value in the new image depends solely on that pixel's value in the old image; in other words, this is a point operation. Contrast manipulation is commonly performed by adjusting the brightness and contrast controls on a television set, or by controlling the exposure and development time in [printmaking](http://www.answers.com/topic/printmaking). Another point operation is that of [pseudo colouring](http://www.answers.com/topic/pseudocoloring) a black-and-white image, by assigning arbitrary colours to the gray levels. This technique is popular in [thermograph](http://www.answers.com/topic/thermography) (the imaging of heat), where hotter objects (with high pixel values) are assigned one color (for example, red), and cool objects (with low pixel values) are assigned another color (for example, blue), with other colours assigned to intermediate values.

Recognizing object classes in real-world images is a long standing goal in Computer vision. Conceptually, this is challenging due to large appearance variations of object instances belonging to the same class. Additionally, distortions from background clutter, scale, and viewpoint variations can render appearances of even the same object instance to be vastly different. Further challenges arise from interclass similarity in which instances from different classes can appear very similar. Consequently, models for object classes must be flexible enough to accommodate class variability, yet discriminative enough to sieve out true object instances in cluttered images. These seemingly paradoxical requirements of an object class model make recognition difficult. This paper addresses two goals of recognition are image classification and object detection. The task of image classification is to determine if an object class is present in an image, while object detection localizes all instances of that class from an image. Toward these goals, the main contribution in this paper is an approach for object class recognition that employs edge information only. The novelty of our approach is that we represent contours by very simple and generic shape primitives of line segments and ellipses, coupled with a flexible method to learn discriminative primitive combinations. These primitives are complementary in nature, where line segment models straight contour and ellipse models curved contour. We choose an ellipse as it is one of the simplest circular shapes, yet is sufficiently flexible to model curved shapes. These shape primitives possess several attractive properties. First, unlike edge-based descriptors they support abstract and perceptually meaningful reasoning like parallelism and adjacency. Also, unlike contour fragment features, storage demands by these primitives are independent of object size and are efficiently represented with four parameters for a line and five parameters for an ellipse.

Additionally, matching between primitives can be efficiently computed (e.g., with geometric properties), unlike contour fragments, which require comparisons between individual edge pixels. Finally, as geometric properties are easily scale normalized, they simplify matching across scales. In contrast, contour fragments are not scale invariant, and one is forced either to rescale fragments, which introduces aliasing effects (e.g., when edge pixels are pulled apart), or to resize an image before extracting fragments, which degrades image resolution.

In recent studies it is shown that the generic nature of line segments and ellipses affords them an innate ability to represent complex shapes and structures. While individually less distinctive, by combining a number of these primitives, we empower a combination to be sufficiently discriminative. Here, each combination is a two-layer abstraction of primitives: pairs of primitives (termed shape tokens) at the first layer, and a learned number of shape tokens at the second layer. We do not constrain a combination to have a fixed number of shape-tokens, but allow it to automatically and flexibly adapt to an object class. This number influences a combination’s ability to represent shapes, where simple shapes favor fewer shape-tokens than complex ones. Consequently, discriminative combinations of varying complexity can be exploited to represent an object class. We learn this combination by exploiting distinguishing shape, geometric, and structural constraints of an object class. Shape constraints describe the visual aspect of shape tokens, while geometric constraints describe its spatial layout (configurations). Structural constraints enforce possible poses/structures of an object by the relationships (e.g., XOR relationship) between shape-tokens.

**CLASSIFICATION OF IMAGES:**

There are 3 types of images used in Digital Image Processing. They are

1. Binary Image
2. Gray Scale Image
3. Colour Image

**BINARY IMAGE:**

A binary image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that has only two possible values for each [pixel](http://en.wikipedia.org/wiki/Pixel).  Typically the two colours used for a binary image are black and white though any two colours can be used.  The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1).This name black and white, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as [grayscale images](http://en.wikipedia.org/wiki/Grayscale)

Binary images often arise in [digital image processing](http://en.wikipedia.org/wiki/Digital_image_processing) as [masks](http://en.wikipedia.org/w/index.php?title=Mask_(image_processing)&action=edit&redlink=1) or as the result of certain operations such as [segmentation](http://en.wikipedia.org/wiki/Segmentation_(image_processing)), [thresholding](http://en.wikipedia.org/wiki/Thresholding_(image_processing)), and [dithering](http://en.wikipedia.org/wiki/Dither). Some input/output devices, such as [laser printers](http://en.wikipedia.org/wiki/Laser_printer), [fax machines](http://en.wikipedia.org/wiki/Fax), and bi-level [computer displays](http://en.wikipedia.org/wiki/Visual_display_unit), can only handle bi-level images

**GRAY SCALE IMAGE**

A grayscale Image is [digital image](http://en.wikipedia.org/wiki/Digital_image) is an image in which the value of each [pixel](http://en.wikipedia.org/wiki/Pixel) is a single [sample](http://en.wikipedia.org/wiki/Sample_(signal)), that is, it carries only [intensity](http://en.wikipedia.org/wiki/Luminous_intensity) information. Images of this sort, also known as [black-and-white](http://en.wikipedia.org/wiki/Black-and-white), are composed exclusively of shades of [gray](http://en.wikipedia.org/wiki/Gray) (0-255), varying from black (0) at the weakest intensity to white (255) at the strongest.

Grayscale images are distinct from one-bit [black-and-white](http://en.wikipedia.org/wiki/Black-and-white) images, which in the context of computer imaging are images with only the two [colors](http://en.wikipedia.org/wiki/Color), [black](http://en.wikipedia.org/wiki/Black), and [white](http://en.wikipedia.org/wiki/White) (also called bi-level or [binary images](http://en.wikipedia.org/wiki/Binary_image)). Grayscale images have many shades of gray in between. Grayscale images are also called [monochromatic](http://en.wikipedia.org/wiki/Monochromatic), denoting the absence of any [chromatic](http://en.wikipedia.org/wiki/Chromaticity) variation.

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the [electromagnetic spectrum](http://en.wikipedia.org/wiki/Electromagnetic_spectrum) (e.g. [infrared](http://en.wikipedia.org/wiki/Infrared), [visible light](http://en.wikipedia.org/wiki/Visible_spectrum), [ultraviolet](http://en.wikipedia.org/wiki/Ultraviolet), etc.), and in such cases they are monochromatic proper when only a given [frequency](http://en.wikipedia.org/wiki/Frequency) is captured. But also they can be synthesized from a full color image; see the section about converting to grayscale.

**COLOUR IMAGE:**

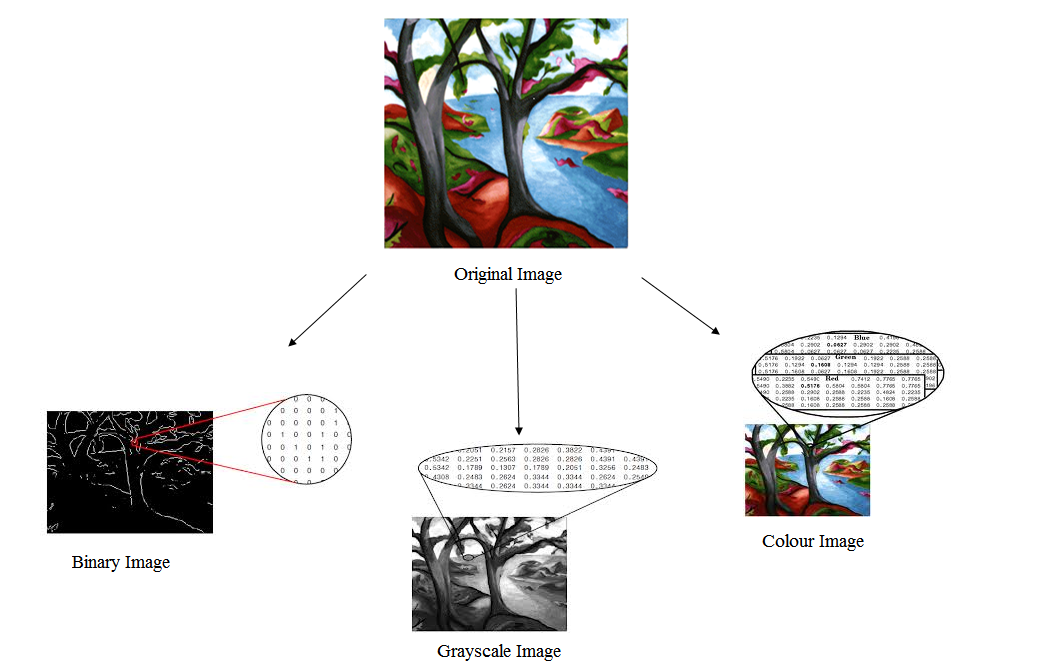
A (digital) color image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that includes [color](http://en.wikipedia.org/wiki/Color) information for each [pixel](http://en.wikipedia.org/wiki/Pixel). Each pixel has a particular value which determines it’s appearing color. This value is qualified by three numbers giving the decomposition of the color in the three primary colours Red, Green and Blue. Any color visible to human eye can be represented this way. The decomposition of a color in the three primary colours is quantified by a number between 0 and 255. For example, white will be coded as R = 255, G = 255, B = 255; black will be known as (R,G,B) = (0,0,0); and say, bright pink will be : (255,0,255).

In other words, an image is an enormous two-dimensional array of color values, pixels, each of them coded on 3 bytes, representing the three primary colours. This allows the image to contain a total of 256x256x256 = 16.8 million different colours. This technique is also known as RGB encoding, and is specifically adapted to human vision

**CLASSIFICATION OF IMAGES:**

There are 3 types of images used in Digital Image Processing. They are

1. Binary Image
2. Gray Scale Image
3. Colour Image



**Figure 1.1**Representation of a image in binary,grayscale and colour form.

**Binary image:**

A binary image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that has only two possible values for each [pixel](http://en.wikipedia.org/wiki/Pixel).  Typically the two colours used for a binary image are black and white though any two colours can be used.  The colour used for the objects inthe image is the foreground colour while the rest of the image is the background colour.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1) [Refer figure 1.1].This name black and white, monochrome or monochromatic are often used for this concept, but also designate any images that have only one sample per pixel, such as [grayscale images](http://en.wikipedia.org/wiki/Grayscale).

Binary images often arise in [digital image processing](http://en.wikipedia.org/wiki/Digital_image_processing) as [masks](http://en.wikipedia.org/w/index.php?title=Mask_(image_processing)&action=edit&redlink=1) or as the result of certain operations such as [segmentation](http://en.wikipedia.org/wiki/Segmentation_(image_processing)), [thresholding](http://en.wikipedia.org/wiki/Thresholding_(image_processing)) and [dithering](http://en.wikipedia.org/wiki/Dither). Some input/output devices, such as [laser printers](http://en.wikipedia.org/wiki/Laser_printer), [fax machines](http://en.wikipedia.org/wiki/Fax), and bi-level [computer displays](http://en.wikipedia.org/wiki/Visual_display_unit), can only handle bi-level images.

**Gray scale image:**

A grayscale Image is [digital image](http://en.wikipedia.org/wiki/Digital_image) is an image in which the value of each [pixel](http://en.wikipedia.org/wiki/Pixel) is a single [sample](http://en.wikipedia.org/wiki/Sample_(signal)), that is, it carries only [intensity](http://en.wikipedia.org/wiki/Luminous_intensity) information. Images of this sort, also known as [black-and-white](http://en.wikipedia.org/wiki/Black-and-white), are composed exclusively of shades of [gray](http://en.wikipedia.org/wiki/Gray) (0-255), varying from black (0) at the weakest intensity to white (255) at the strongest [Refer figure 1.1].

Grayscale images are distinct from one-bit [black-and-white](http://en.wikipedia.org/wiki/Black-and-white) images, which in the context of computer imaging are images with only the two [colours](http://en.wikipedia.org/wiki/Color), [black](http://en.wikipedia.org/wiki/Black), and [white](http://en.wikipedia.org/wiki/White) (also called bi-level or [binary images](http://en.wikipedia.org/wiki/Binary_image)). Grayscale images have many shades of gray in between. Grayscale images are also called[monochromatic](http://en.wikipedia.org/wiki/Monochromatic), denoting the absence of any [chromatic](http://en.wikipedia.org/wiki/Chromaticity) variation.

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the [electromagnetic spectrum](http://en.wikipedia.org/wiki/Electromagnetic_spectrum) (e.g. [infrared](http://en.wikipedia.org/wiki/Infrared), [visible light](http://en.wikipedia.org/wiki/Visible_spectrum), [ultraviolet](http://en.wikipedia.org/wiki/Ultraviolet), etc.), and in such cases they are monochromatic in nature when only a given [frequency](http://en.wikipedia.org/wiki/Frequency) is captured.

**Colour image:**

A colour image is a [digital image](http://en.wikipedia.org/wiki/Digital_image) that includes [colour](http://en.wikipedia.org/wiki/Color) information for each [pixel](http://en.wikipedia.org/wiki/Pixel). Each pixel has a particular value which determines it’s appearing colour. This value is qualified by three numbers giving the decomposition of the colour in the three primary colours Red, Green and Blue. Any colour visible to human eye can be represented this way. The decomposition of a colour in the three primary colours is quantified by a number between 0 and 255. For example, white will be coded as R = 255, G = 255, B = 255; black will be known as (R,G,B) = (0,0,0)[Refer figure 1.1].In other words, an image is an enormous two-dimensional array of colour values, pixels, each of them coded on 3 bytes, representing the three primary colours. This allows the image to contain a total of 256x256x256 = 16.8 million different colours. This technique is also known as RGB encoding, and is specifically adapted to human vision.

**APPLICATIONS**

**Digital camera images**

Digital cameras generally include dedicated digital image processing chips to convert the raw data from the [image sensor](http://en.wikipedia.org/wiki/Image_sensor) into a [colour-corrected](http://en.wikipedia.org/wiki/Color_correction) image in a standard [image file format](http://en.wikipedia.org/wiki/Image_file_formats). Images from digital cameras often receive further processing to improve their quality, a distinct advantage that digital cameras have over [film](http://en.wikipedia.org/wiki/Photographic_film) cameras. The digital image processing typically is executed by special software programs that can manipulate the images in many ways. Many digital cameras also enable viewing of [histograms](http://en.wikipedia.org/wiki/Histogram) of images, as an aid for the photographer to understand the rendered brightness range of each shot more readily.

**Intelligent transportation systems**

Digital image processing has wide applications in intelligent transportation systems, such as [automatic number plate recognition](http://en.wikipedia.org/wiki/Automatic_number_plate_recognition) and [traffic sign recognition](http://en.wikipedia.org/wiki/Traffic_sign_recognition).

**Chapter 7**

**Hardware and software requirement**

**Hardware requirement:**

* Hdd=1tb

**Software requirement**

* Python ,
* anaconda navigator

**4.2.2PYTHON**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language) for programming Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

**PYTHON FEATURES:**

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, including object-oriented , [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 library. Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support functional programming and aspect [-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by meta [programming](https://en.wikipedia.org/wiki/Metaprogramming) and [meta objects](https://en.wikipedia.org/wiki/Metaobject) (magic methods)). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).

**PYTHON LIBRARIES**

Python's large [standard library](https://en.wikipedia.org/wiki/Standard_library), commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications,

18

many standard formats and protocols such as [MIME](https://en.wikipedia.org/wiki/MIME) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) are supported. It

includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary precision decimals, manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testin](https://en.wikipedia.org/wiki/Unit_testing)g.

As of March 2018, the [Python Package Index](https://en.wikipedia.org/wiki/Python_Package_Index) (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration
* Scientific computing
* Text processing
* Image processing

**DEPLOYMENT ENVIRONMENT**

Most Python implementations (including CPython) include a [read–eval–print loop](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) (REPL), permitting them to function as a [command line interpreter](https://en.wikipedia.org/wiki/Command_line_interpreter) for

which the user enters statements sequentially and receives results immediately.

**OPENCV-PYTHON**

 Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

Since OpenCV is an open source initiative, all are welcome to make contributions to this library. And it is same for this tutorial also. So, if you find any mistake in this tutorial (whether it be a small spelling mistake or a big error in code or concepts, whatever), feel free to correct it

And that will be a good task for freshers who begin to contribute to open source projects. Just fork the OpenCV in github, make necessary corrections and send a pull request to OpenCV. OpenCV developers will check your pull request, give you important feedback and once it passes the approval of the reviewer, it will be merged to OpenCV. Then you become a open source contributor. Similar is the case with other tutorials, documentation etc. As new modules are added to OpenCV-Python, this tutorial will have to be expanded. So those who knows about particular algorithm can write up a tutorial which includes a basic theory of the algorithm and a code showing basic usage of the algorithm and submit it to OpenCV. Remember, we together can make this project a great success !!!

**ARRAY ATTRIBUTES**

Array attributes reflect information that is intrinsic to the array itself. Generally, accessing an array through its attributes allows you to get and sometimes set intrinsic properties of the array without creating a new array. The exposed attributes are the core parts of an array and only some of them can be reset meaningfully without creating a new array.

For a 1-D array, this has no effect. (To change between column and row vectors, first cast the 1-D array into a matrix object.) For a 2-D array, this is the usual matrix transpose. For an n-D array, if axes are given, their order indicates how the axes are permuted (see Examples). If axes are not provided and a.shape = (i[0], i[1], ... i[n-2], i[n-1]), then a.transpose().shape = (i[n-1], i[n-2], ... i[1], i[0]).

**SCALARS**

Python defines only one type of a particular data class (there is only one integer type, one floating-point type, etc.). This can be convenient in applications that don’t need to be concerned with all the ways data can be represented in a computer. For scientific computing, however, more control is often needed. In NumPy, there are 24 new fundamental Python types to describe different types of scalars. These type descriptors are mostly based on the types available in the C language that CPython is written in, with several additional types compatible with Python’s types.

**METHODS**

Array scalars have exactly the same methods as arrays. The default behavior of these methods is to internally convert the scalar to an equivalent 0-dimensional array and to call the corresponding array method. In addition, math operations on array scalars are defined so that the same hardware flags are set and used to interpret the results as for ufunc, so that the error state used for ufuncs also carries over to the math on array scalars.

**DATA TYPE OBJECTS (DTYPE)**

A data type object (an instance of numpy.dtype class) describes how the bytes in the fixed-size block of memory corresponding to an array item should be interpreted. It describes the following aspects of the data: 1. Type of the data (integer, float, Python object, etc.) 2. Size of the data (how many bytes is in e.g. the integer) 3. Byte order of the data (little-endian or big-endian) 4. If the data type is structured, an aggregate of other data types, (e.g., describing an array item consisting of an integer and a float), (a) what are the names of the “fields” of the structure, by which they can be accessed, (b) what is the data-type of each field, and (c) which part of the memory block each field takes. 5. If the data type is a sub-array, what is its shape and data type. To describe the type of scalar data, there are several built-in scalar types in Numpy for various precision of integers, floating-point numbers, etc. An item extracted from an array, e.g., by indexing, will be a Python object whose type is the scalar type associated with the data type of the array. Note that the scalar types are not dtype objects, even though they can be used in place of one whenever a data type specification is needed in Numpy. Structured data types are formed by creating a data type whose fields contain other data types. Each field has a name by which it can be accessed. The parent data type should be of sufficient size to contain all its fields; the parent is nearly always based on the void type which allows an arbitrary item size. Structured data types may also contain nested structured sub-array data types in their fields. Finally, a data type can describe items that are themselves arrays of items of another data type. These sub-arrays must, however, be of a fixed size. If an array is created using a data-type describing a sub-array, the dimensions of the sub-array are appended to the shape of the array when the array is created. Sub-arrays in a field of a structured type behave differently, see Field Access. Sub-arrays always have a C-contiguous memory layout

**OpenCV**

OpenCV was started at Intel in 1999 by **Gary Bradsky** and the first release came out in 2000. **Vadim Pisarevsky** joined Gary Bradsky to manage Intel’s Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations.

OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.

**OpenCV-Python**

Python is a general purpose programming language started by **Guido van Rossum**, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

And the support of Numpy makes the task more easier. **Numpy** is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

**OpenCV-Python Tutorials**

OpenCV introduces a new set of tutorials which will guide you through various functions available in OpenCV-Python. **This guide is mainly focused on OpenCV 3.x version** (although most of the tutorials will work with OpenCV 2.x also).

A prior knowledge on Python and Numpy is required before starting because they won’t be covered in this guide. **Especially, a good knowledge on Numpy is must to write optimized codes in OpenCV-Python.**

This tutorial has been started by Abid Rahman K. as part of Google Summer of Code 2013 program, under the guidance of Alexander Mordvintsev.

**OpenCV Needs You !!!**

Since OpenCV is an open source initiative, all are welcome to make contributions to this library. And it is same for this tutorial also.

So, if you find any mistake in this tutorial (whether it be a small spelling mistake or a big error in code or concepts, whatever), feel free to correct it.

And that will be a good task for freshers who begin to contribute to open source projects. Just fork the OpenCV in github, make necessary corrections and send a pull request to OpenCV. OpenCV developers will check your pull request, give you important feedback and once it passes the approval of the reviewer, it will be merged to OpenCV. Then you become a open source contributor. Similar is the case with other tutorials, documentation etc.

As new modules are added to OpenCV-Python, this tutorial will have to be expanded. So those who knows about particular algorithm can write up a tutorial which includes a basic theory of the algorithm and a code showing basic usage of the algorithm and submit it to OpenCV.

Remember, we **together** can make this project a great success !!!

**Contributors**

Below is the list of contributors who submitted tutorials to OpenCV-Python.

1. Alexander Mordvintsev (GSoC-2013 mentor)
2. Abid Rahman K. (GSoC-2013 intern)

**Additional Resources**

1. A Quick guide to Python - [A Byte of Python](http://swaroopch.com/notes/python/)
2. [Basic Numpy Tutorials](http://wiki.scipy.org/Tentative_NumPy_Tutorial)
3. [Numpy Examples List](http://wiki.scipy.org/Numpy_Example_List)
4. [OpenCV Documentation](http://docs.opencv.org/)
5. [OpenCV Forum](http://answers.opencv.org/questions/)

# Install OpenCV-Python in Windows

## Goals

**In this tutorial**

* We will learn to setup OpenCV-Python in your Windows system.

Below steps are tested in a Windows 7-64 bit machine with Visual Studio 2010 and Visual Studio 2012. The screenshots shows VS2012.

## Installing OpenCV from prebuilt binaries

1. Below Python packages are to be downloaded and installed to their default locations.

1.1. [Python-2.7.x](http://python.org/ftp/python/2.7.5/python-2.7.5.msi).

1.2. [Numpy](http://sourceforge.net/projects/numpy/files/NumPy/1.7.1/numpy-1.7.1-win32-superpack-python2.7.exe/download).

1.3. [Matplotlib](https://downloads.sourceforge.net/project/matplotlib/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe) (Matplotlib is optional, but recommended since we use it a lot in our tutorials).

1. Install all packages into their default locations. Python will be installed to **C:/Python27/**.
2. After installation, open Python IDLE. Enter import numpy and make sure Numpy is working fine.
3. Download latest OpenCV release from [sourceforge site](http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.6/OpenCV-2.4.6.0.exe/download) and double-click to extract it.
4. Goto **opencv/build/python/2.7** folder.
5. Copy **cv2.pyd** to **C:/Python27/lib/site-packeges**.
6. Open Python IDLE and type following codes in Python terminal.
7. >>> import cv2
8. >>> print cv2.\_\_version\_\_

If the results are printed out without any errors, congratulations !!! You have installed OpenCV-Python successfully.

## Building OpenCV from source

1. Download and install Visual Studio and CMake.

1.1. [Visual Studio 2012](http://go.microsoft.com/?linkid=9816768)

1.2. [CMake](http://www.cmake.org/files/v2.8/cmake-2.8.11.2-win32-x86.exe)

1. Download and install necessary Python packages to their default locations

2.1. [Python 2.7.x](http://python.org/ftp/python/2.7.5/python-2.7.5.msi)

2.2. [Numpy](http://sourceforge.net/projects/numpy/files/NumPy/1.7.1/numpy-1.7.1-win32-superpack-python2.7.exe/download)

2.3. [Matplotlib](https://downloads.sourceforge.net/project/matplotlib/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe) (Matplotlib is optional, but recommended since we use it a lot in our tutorials.)

### Computer Vision Resources

##### Packages And Frameworks

[Opencv](https://opencv.org/) – “Opencv Was Designed For Computational Efficiency And With A Strong Focus On Real-Time Applications. Adopted All Around The World, Opencv Has More Than 47 Thousand People Of User Community And Estimated Number Of Downloads Exceeding 14 Million. Usage Ranges From Interactive Art, To Mines Inspection, Stitching Maps On The Web Or Through Advanced Robotics.”

[Simplecv](http://simplecv.org/) – “Simplecv Is An Open Source Framework For Building Computer Vision Applications. With It, You Get Access To Several High-Powered Computer Vision Libraries Such As Opencv – Without Having To First Learn About Bit Depths, File Formats, Color Spaces, Buffer Management, Eigenvalues, Or Matrix Versus Bitmap Storage.”

[Mahotas](http://mahotas.readthedocs.io/en/latest/) – “Mahotas Is A Computer Vision And Image Processing Library For Python. It Includes Many Algorithms Implemented In C++ For Speed While Operating In Numpy Arrays And With A Very Clean Python Interface. Mahotas Currently Has Over 100 Functions For Image Processing And Computer Vision And It Keeps Growing.

* **NumPy:**
* Numpy, Which Stands For Numerical Python, Is A Library Consisting Of Multidimensional Array Objects And A Collection Of Routines For Processing Those Arrays. Using Numpy, Mathematical And Logical Operations On Arrays Can Be Performed. This Tutorial Explains The Basics Of Numpy Such As Its Architecture And Environment. It Also Discusses The Various Array Functions, Types Of Indexing, Etc. An Introduction To Matplotlib Is Also Provided. All This Is Explained With The Help Of Examples For Better Understanding.
* Audience
* This Tutorial Has Been Prepared For Those Who Want To Learn About The Basics And Various Functions Of Numpy. It Is Specifically Useful For Algorithm Developers. After Completing This Tutorial, You Will Find Yourself At A Moderate Level Of Expertise From Where You Can Take Yourself To Higher Levels Of Expertise.
* Prerequisites
* You Should Have A Basic Understanding Of Computer Programming Terminologies. A Basic Understanding Of Python And Any Of The Programming Languages Is A Plus.
* Numpy Is A Python Package. It Stands For 'Numerical Python'. It Is A Library Consisting Of Multidimensional Array Objects And A Collection Of Routines For Processing Of Array.

**Numeric**, The Ancestor Of Numpy, Was Developed By Jim Hugunin. Another Package Numarray Was Also Developed, Having Some Additional Functionalities. In 2005, Travis Oliphant Created Numpy Package By Incorporating The Features Of Numarray Into Numeric Package. There Are Many Contributors To This Open Source Project.

## Operations Using Numpy

Using Numpy, A Developer Can Perform The Following Operations −

* Mathematical And Logical Operations On Arrays.
* Fourier Transforms And Routines For Shape Manipulation.
* Operations Related To Linear Algebra. Numpy Has In-Built Functions For Linear Algebra And Random Number Generation.

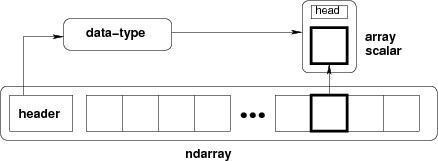
## Numpy – A Replacement ForMatlab

Numpy Is Often Used Along With Packages Like **Scipy** (Scientific Python) And **Mat−Plotlib** (Plotting Library). This Combination Is Widely Used As A Replacement For Matlab, A Popular Platform For Technical Computing. However, Python Alternative To Matlab Is Now Seen As A More Modern And Complete Programming Language.

The Most Important Object Defined In Numpy Is An N-Dimensional Array Type Called **Ndarray**. It Describes The Collection Of Items Of The Same Type. Items In The Collection Can Be Accessed Using A Zero-Based Index.

Every Item In An Ndarray Takes The Same Size Of Block In The Memory. Each Element In Ndarray Is An Object Of Data-Type Object (Called **Dtype**).

Any Item Extracted From Ndarray Object (By Slicing) Is Represented By A Python Object Of One Of Array Scalar Types. The Following Diagram Shows A Relationship BetweenNdarray, Data Type Object (Dtype) And Array Scalar Type −



An Instance Of Ndarray Class Can Be Constructed By Different Array Creation Routines Described Later In The Tutorial. The Basic Ndarray Is Created Using An Array Function In Numpy As Follows −

Numpy.Array

It Creates An Ndarray From Any Object Exposing Array Interface, Or From Any Method That Returns An Array.

**Imutils:**

A Series Of Convenience Functions To Make Basic Image Processing Operations Such As Translation, Rotation, Resizing, Skeletonization, And Displaying Matplotlib Images Easier With Opencv And Python.

Transalation

Translation Is The Shifting Of An Image In Either The X Or Y Direction. To Translate An Image In Opencv You Need To Supply The (X, Y)-Shift, Denoted As (Tx, Ty) To Construct The Translation Matrix M:

translation_eq

And From There, You Would Need To Apply The Cv2.Warpaffine  Function.

Instead Of Manually Constructing The Translation Matrix M And Calling Cv2.Warpaffine , You Can Simply Make A Call To The Translate  Function Of Imutils

To get the pip package manager, you first need to install Python. Download the latest version of Python from the official Python website and install it.

Once the installation completes, check for the version of pip running on your system. To do so, go to the command prompt and type:

$ pip3 --version

Since you have installed the latest version of Python, that is, Python 3.x, you have pip3, and not pip. The latter was used with Python 2.7.

It is now finally time to install TensorFlow. Run the windows command prompt as an administrator. To do so, go to the start menu on your Windows machine, search for "cmd", right click it and choose "Run as administrator".

After that, you only have to run one simple command to install TensorFlow. Here is the command:

$ pip3 install --upgrade tensorflow

The command will take some time to execute, so remain patient. With pip, you can install TensorFlow with GPU support as follows:

$ pip3 install tensorflow-gpu

And that's it! You can now skip to the section "Verifying the Installation" below to make sure it installed correctly.

Or else use **pip install tensorflow==2.3.0**

Keras installation is quite easy. Follow below steps to properly install Keras on your system.

Step 1: Create virtual environment

Virtualenv is used to manage Python packages for different projects. This will be helpful to avoid breaking the packages installed in the other environments. So, it is always recommended to use a virtual environment while developing Python applications.

**Windows**

Windows user can use the below command,

**Pip install keras==2.1.0.**

**Dlib flie:**

**Install delib for the deep learning and go to chrome type links for dlib. Select required package like**

[dlib-19.7.0-cp36-cp36m-win\_amd64.whl](https://files.pythonhosted.org/packages/da/06/bd3e241c4eb0a662914b3b4875fc52dd176a9db0d4a2c915ac2ad8800e9e/dlib-19.7.0-cp36-cp36m-win_amd64.whl#sha256=d71c78dce0f7614e05b92518cf4ad654af9e8bfc9e15dea6af3c4dbb306bcd09).

download it and save in scripts.copy the path and install in command prompt **pip install** [dlib-19.7.0-cp36-cp36m-win\_amd64.whl](https://files.pythonhosted.org/packages/da/06/bd3e241c4eb0a662914b3b4875fc52dd176a9db0d4a2c915ac2ad8800e9e/dlib-19.7.0-cp36-cp36m-win_amd64.whl#sha256=d71c78dce0f7614e05b92518cf4ad654af9e8bfc9e15dea6af3c4dbb306bcd09).

**Conclusion and Future Scope**

The aim of the project is to face recognition and face emotion by using deep learning technique. In this we proposed real time video surveillance, what human face expresses it in front of camera and they were recognising the face.after completion we should find exactly stress detection on face.

**Future scope**:

In future we have increase the accuracy rate based on stress detection using hardware with music.

**Chapter 9:**

**Reference**

[1] H. L. Wagner, R. Buck, M. Winterbotham, "Communication of specific person behaviors: Ge

nder differences in sending accuracy and communication measures," J. Nonverbal. Behav., vol.17, pp.29-52, 1993.

[2] J. G. Allen, D. M. Haccoun, "Sex differences in person behaviorality: A multidimensional approach," Hum. Relat., vol. 29, pp. 711- 722, 1976.

 [3] C. L. Barr, R. E. Kleck, Self-other perception of the intensity of facial expressions of person behavior: Do we know what we show," J. Pers. Soc. Psychol.vol. 68, pp.608-618, 1995.

 [4] A. M. Kring, A. H. Gordon, "Sex differences in person behavior: Expression, experience, and physiology," J. Pers. Soc. Psychol. vol. 74, pp. 686-703, 1998.

[5] L. R. Brody, Gender, person behavioral expression, and parent-child boundaries. In R. D. Kavanaugh, B. Zimmerberg, & S. Fein (Eds.), Person behavior: Interdisciplinary perspectives, 1996, pp. 139-170. Mahwah, NJ: Lawrence Erlbaum.

 [6] A. Campbell, "Staying alive: Evolution, culture and intra-female aggression," Behav. Brain. Sci., vol. 22, pp. 203-252, 1999.

 [7] C. Kirschbaum, D. H. Hellhammer, "Salivary cortisol in psychoneuroendocrine research: Recent developments and applications." psychoneuroendocrino, vol.19, pp. 313–333, 1994.

 [8] O. Houstis, S. Kiliaridis, "Gender and age differences in facial expressions," Eur. J. Orthod., vol. 31(5), pp. 459-466, 2009.

 [9] C. Sforza, A. Mapelli, D. Galante, S. Moriconi, T. Ibba, L. Ferraro,V. Ferrario, "The effect of age and sex on facial mimicry: a threedimensional study in healthy adults,", Int. J. Oral. Maxillofac. Surg., vol. 39(10), pp.990-999, 2010.

 [10] D. McDuff, E. Kodra, R. Kaliouby, M. LaFrance, "A large-scale analysis of sex differences in facial expressions," PLoS. One., vol. 12(4),2107.

[11] P. Rai, P. Khanna,"Gender Classification Techniques: A Review," Adv. Intel. Soft. Compu., vol. 166, pp. 51–59, 2012.

[12] S. A. Khan, M. Nazir, S. Akram, N. Riaz, "Gender classification using image processing techniques: A survey," In Proceedings of the 2011 IEEE 14th International Multitopic Conference (INMIC) , pp. 25–30, 2011.

 [13] W. S. Chu, C. R. Huang,C. S. Chen, "Gender classification from unaligned facial images using support subspaces," Inform. Sciences., vol. 221, pp. 98–109 , 2013.

[14] B. Patel, R. P. Maheshwari, R. Balasubramanian, "Multi-quantized local binary patterns for facial gender classification,", Comput. Electr. Eng., vol. 54, pp. 271-284, 2016.

 [15] <http://pics.stir.ac.uk/2D_face_sets.htm>

 [16] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009.

 [17] www.Neurotechnology.com