Cairo University

Faculty of Engineering.
Computer Engineering Department.

Project title:

Face recognition.

Presented by :

Name	section	BN
Nada Ahmed Fawzy	33	17
Nada Elsayed Mohammed	33	20
Noha Ayman	33	35
Nada Mamdouh	33	33
Nada Hossam	33	21
Nadeen Ayman	33	5
Mohammed nour	33	

In partial fulfillment of the Requirement for
GNE001 Practical and Engineering Application (physics part)

Table of contents

Contents

List of figure:

Introduction:

Over the last ten years or so, face recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding. Because of the nature of the problem, not only computer science researchers are interested in it, but researchers from various fields also. The goal is to implement the system (model) for a particular face and distinguish it from a large number of stored faces with some real-time variations as well. It gives us efficient way to find the lower dimensional space. Further this algorithm can be extended to recognize the gender of a person or to interpret the facial expression of a person. Recognition could be carried out under widely varying conditions like frontal view, a 45° view, scaled frontal view, subjects with spectacles etc are tried, while the training data set covers limited views.

The aim of this project is to study and develop an efficient MATLAB program for face recognition using principal component analysis and to perform test for program.

One of the simplest and most effective PCA approaches used in face recognition systems is the so-called eigenface approach. This approach transforms faces into a small set of essential characteristics, eigenfaces, which are the main components of the initial set of learning images (training set). Recognition is done by projecting a new image in the eigenface subspace, after which the person is classified by comparing its position in eigenface space with the position of known individuals. The advantage of this approach over other face recognition systems is in its simplicity, speed and insensitivity to small or gradual changes on the face. The problem is limited to files that can be used to recognize the face. Namely, the images must be vertical frontal views of human faces. The whole recognition process involves two steps:

A. Initialization process

B. Recognition process

The Initialization process involves the following operations:

i. Acquire the initial set of face images called as training set.
ii. Calculate the Eigenfaces from the training set, keeping only the highest eigenvalues. These M images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.

iii. Calculate distribution in this M-dimensional space for each known person by projecting his or her face images onto this face-space.

Machine Learning

- **Definition:**

It's the science of continuously feeding data to a machine so that it can interpret this data, understand it, and detect patterns and useful insides to help it solve problems. The aim is for the machine to be able to solve problems without being explicitly programmed.

- **Categories:**

- Supervised learning.

- Unsupervised learning.

- Reinforcement learning.

  - **Supervised Learning– Train Me! :**

Is method in which we teach the machine using labeled data, where we explicitly tell it the input and the correct output. It is similar to being guided by a teacher.

Types of supervised learning problems:

✓ **Classification:** When the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.

Example: classifying an email as spam or no-spam.

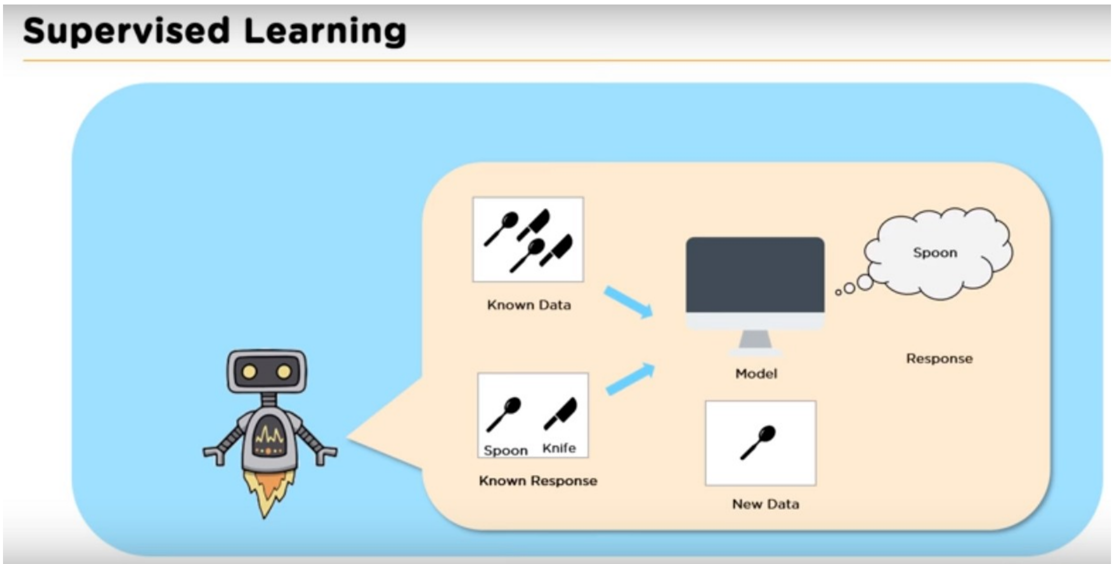


figure 1 classification supervised learning

- ✓ **Regression:** Relationship between two or more variable where a change in one variable is associated with a change in other variable. It is used to predict a continuous variable.

Example: predicting a person's weight. predicting stock prices.

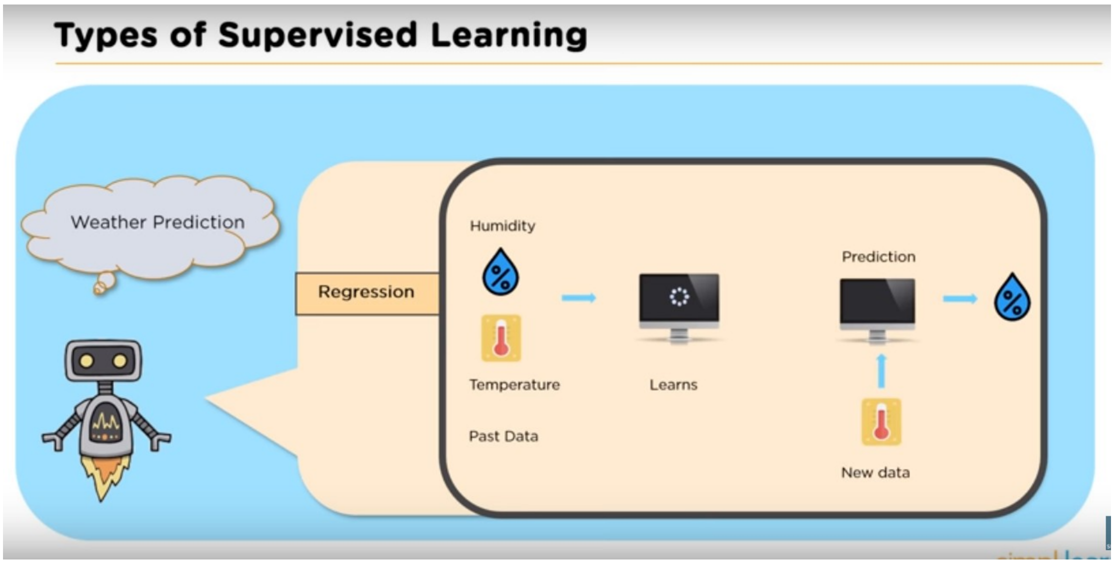


figure 2 regression supervised learning

- ### Unsupervised Learning – I am self sufficient in learning:

“The method to reach outcome or output for the given inputs is unknown”, here input data is given and the model is run on it, the model isn't guided and must find patterns inside the data to help it correctly predict the output.

Types of unsupervised learning problems:

- ✓ **Clustering:** the method of dividing the objects into clusters where elements of each cluster are similar between them and dissimilar to the objects belonging to another cluster.

Example: websites that group users of similar interests together and suggests them as friends.

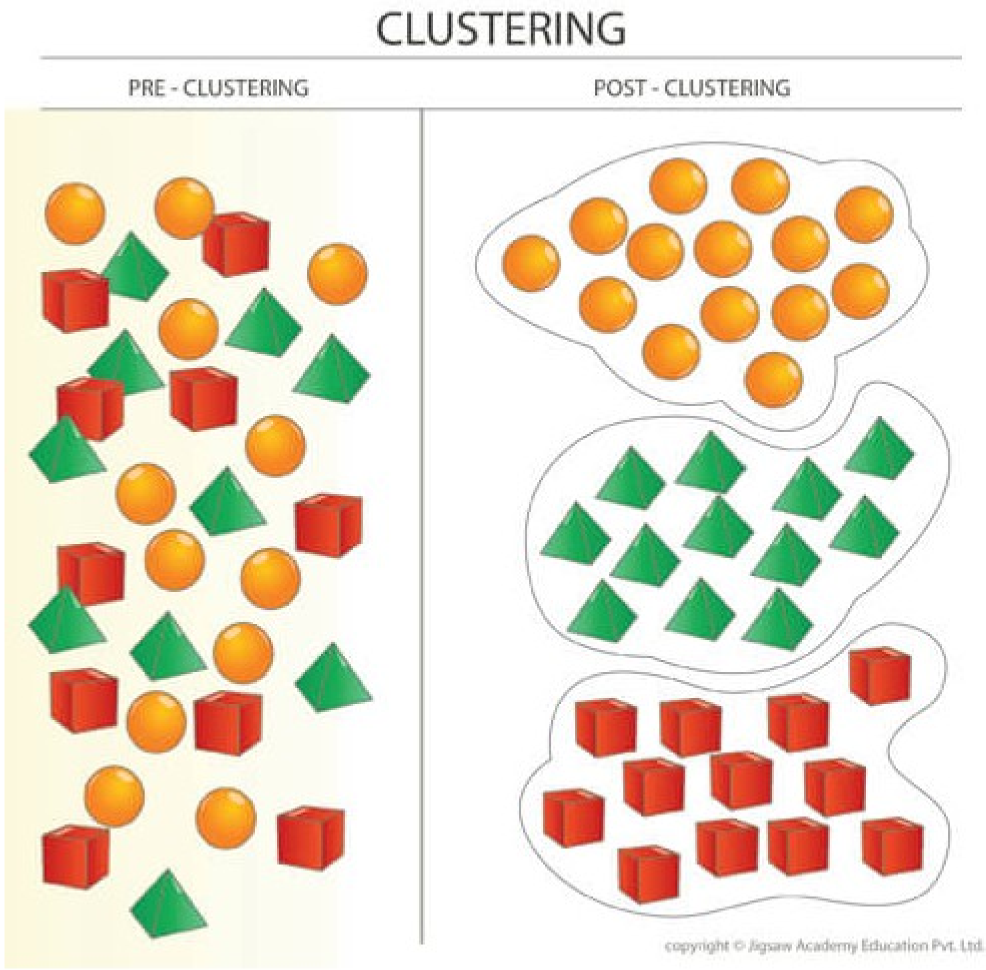


figure 3 clustring unsupervised learning

- ✓ Association: An association rule learning problem is where you want to discover rules that describe the relations between variables of your data that are similar or co-occur.

Example: people tend to buy product X and Y together, so we use association technique to find the relation between X and Y.

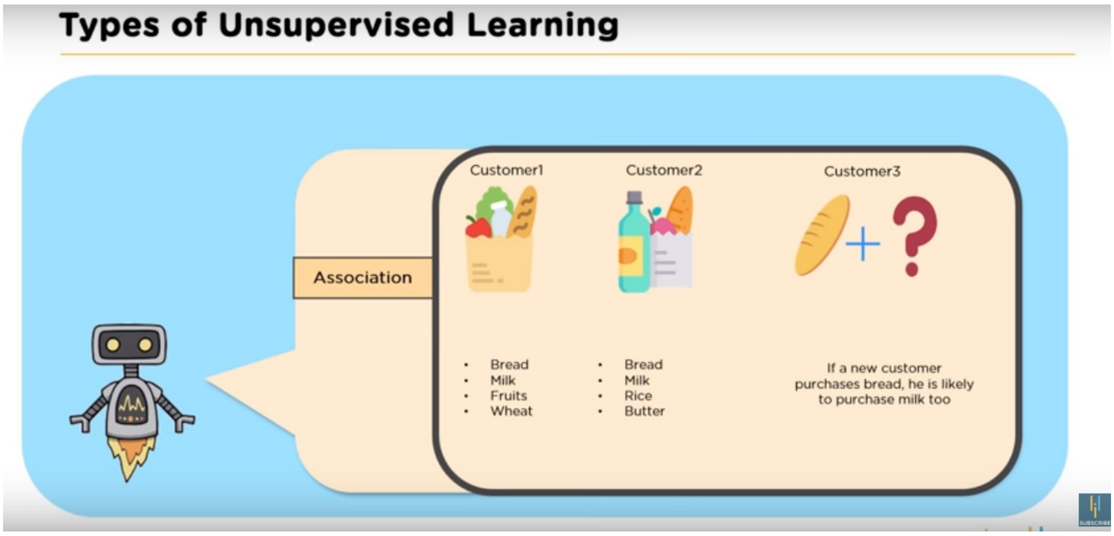


figure 4 Association unsupervised learning

o Reinforcement Learning– My life My rules! (Hit & Trial):

It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.

The agent (machine) is thrown in an unfamiliar environment and learns what to do by taking actions where it discovers errors and rewards. The agent then decides the path to follow to avoid errors and receive maximum rewards.

Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task.

Facial Expression Recognition

- **Definition:**

Face recognition is a method that uses biometric markers to identify the identity of an individual and his emotions. Face recognition systems can be used to identify people in photos, video, or in real-time.

- **Importance:**

Facial expressions and other gestures convey nonverbal communication cues that play an important role in interpersonal relations. These cues complement speech by helping the listener to interpret the intended meaning of spoken words. Therefore, facial expression recognition, because it extracts and analyzes information from an image or video feed, it is able to deliver unfiltered, unbiased emotional responses as data.

• How it works:

The software identifies 80 nodal points on a human face. In this context, nodal points are endpoints used to measure variables of a person's face, such as the length or width of the nose, the depth of the eye sockets and the shape of the cheekbones. The system works by capturing data for nodal points on a digital image of an individual's face and storing the resulting data as a faceprint. The faceprint is then used as a basis for comparison with data captured from faces in an image or video.

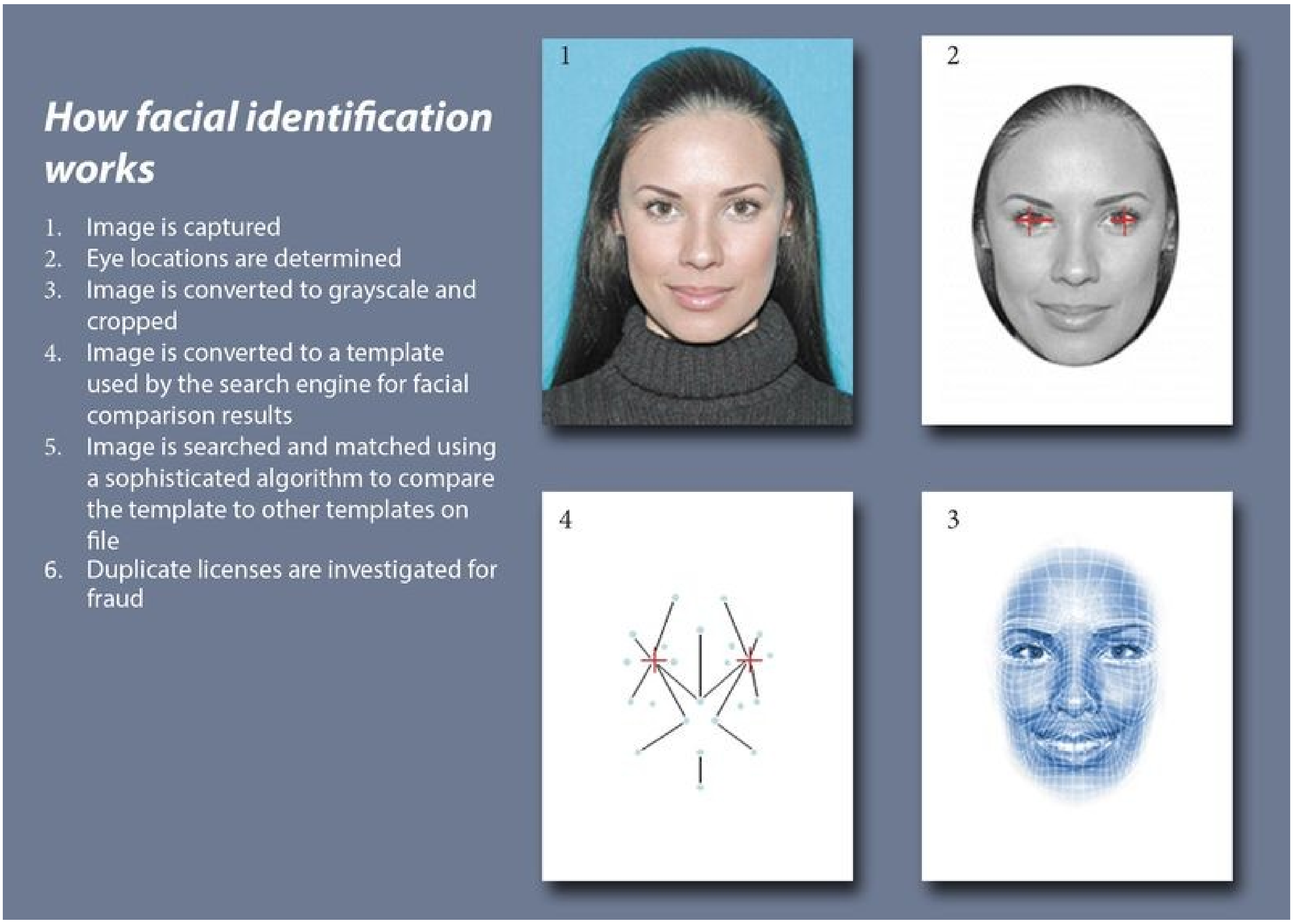


figure 5 How facial identification works

- ## Errors in this technology:

face recognition systems vary in their ability to identify people under challenging conditions such as poor lighting, low quality image resolution, and suboptimal angle of view.

When it comes to errors, there are two key concepts to understand:

- ✓ **False negative:** is when the face recognition system fails to match a person's face to an image that is, in fact, contained in a database. In other words, the system will erroneously return zero results in response to a query.

- ✓ **False positive:** is when the face recognition system does match a person's face to an image in a database, but that match is actually incorrect.

- # Applications:

  - ✓ **Market research:** employs behavioral methods where user's reactions are observed, while interacting with a brand or a product. facial expression recognition technology allows companies to conduct market research and measure moment-by-moment facial expressions of emotions automatically, making it easy to aggregate the results.
  - ✓ **Video game testing:** helps game developers gain insights and draw conclusions about the emotions experienced during game play and incorporate that feedback in the making of the final product.
  - ✓ **Making cars safer:** Using facial emotion detection smart cars can alert the driver when he is feeling drowsy.
  - ✓ **Facial emotion detection in interviews:** A candidate-interviewer interaction is susceptible to many categories of judgment and subjectivity. Such subjectivity makes it hard to determine whether candidate's personality is a good fit for the job. Face expression recognition helps assess the candidate personality traits which helps employers make better decisions while hiring.

- # Organizations that use face recognition:

  - ✓ **U.S. government at airports:** Facial recognition systems can monitor people coming and going in airports. The Department of Homeland Security has used the technology to identify people who have overstayed their visas or may be under criminal investigation

  - ✓ **makers in products:** Apple first used facial recognition to unlock its iPhone X, and continues with the iPhone XS. Face ID authenticates — it makes sure you're you when you access your phone. Apple says the chance of a random face unlocking your phone is about one in 1 million.

  - ✓ **Social media companies on websites:** Facebook uses an algorithm to spot faces when you upload a photo to its platform. The social media company asks if you want to tag people in your photos. If you say yes, it creates a link to their profiles. Facebook can recognize faces with 98 percent accuracy.

Image Processing :

- **Definition:**

- The basic definition of image processing refers to processing of digital image, i.e removing the noise and any kind of irregularities present in an image using the digital computer in order to get an enhanced image or to extract some useful information from it. The noise or irregularity may creep into the image either during its formation or during transformation etc. For mathematical analysis, an image may be defined as a two dimensional function f(x,y) where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are all finite, discrete quantities, we call the image a digital image. It is very important that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, and pixels. Pixel is the most widely used term to denote the elements of a digital image.

it consist of several stages:

1-image import

2-analysis

3-manipulation

4-image output

- **DIGITAL IMAGE PROCESSING**

The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principle advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision. The various Image Processing techniques are:

1. Image preprocessing
2. Image enhancement
3. Image segmentation
4. Feature extraction
5. Image classification

IMAGE PREPROCESSING

In image preprocessing, image data recorded by sensors on a satellite restrain errors related to geometry and brightness values of the pixels. These errors are corrected using appropriate mathematical models which are either definite or statistical models. Image enhancement is the modification of image by changing the pixel brightness values to improve its visual impact. Image enhancement involves a collection of techniques that are used to improve the visual appearance of an image, or to convert the image to a form which is better suited for human or machine interpretation.

Sometimes images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have different types of noise. In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display. Examples include contrast and edge enhancement, pseudo-coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application dependent. Some of the enhancement techniques are:

a. Contrast Stretching

b. Noise Filtering

c. Histogram modification

a. Contrast Stretching

Some images (eg. over water bodies, deserts, dense forests, snow, clouds and under hazy conditions over heterogeneous regions) are homogeneous i.e., they do not have much change in their levels. In terms of histogram representation, they are characterized as the occurrence of very narrow peaks. The homogeneity can also be due to the incorrect illumination of the scene. Ultimately the images hence obtained are not easily interpretable due to poor human perceptibility. This is because there exists only a narrow range of gray-levels in the image having provision for wider range of gray-levels. The contrast stretching methods are designed exclusively for frequently encountered situations. Different stretching techniques have been developed to stretch the narrow range to the whole of the available dynamic range.

b. Noise Filtering

Noise Filtering is used to filter the unnecessary information from an image. It is also used to remove various types of noises from the images. Mostly this feature is interactive. Various filters like low pass, high pass, mean, median etc., are available.

c. Histogram Modification

Histogram has a lot of importance in image enhancement. It reflects the characteristics of image. By modifying the histogram, image characteristics can be modified. One such example is Histogram Equalization. Histogram equalization is a nonlinear stretch that redistributes pixel values so that there is approximately the same number of pixels with each value within a range. The result approximates a flat histogram. Therefore, contrast is increased at the peaks and lessened at the tails.

- **There are two methods of image processing:**
  - ✓ Digital and Analogue

What are the major techniques of digital image processing??

1-image editing, which basically means altering digital images by means of graphic software tools

2-image restoration, refers to the estimation of a clean original image out of the corrupt image taken in order to get back the information lost.

3-independent component analysis, separates a multivariate signal computationally into additive subcomponent.

4-anisstropic diffusion, which is often known as perona-malik diffusion, makes it possible to reduce image noise without having to remove important parts of the image.

5- linear filtering, it's another digital image processing technique, refers to processing time varying input signals and producing output signals that are subject to the constraint of linearity.

6-neural networks, which are computational models widely used in machine for solving various tasks.

7-pixelation,which often refers to turning printed images into digitized ones (such as GIF).

8-principal components analysis , which can be used for feature extraction.

9-partial differential equations, is dealing with effectively de-noising image.

10-hidden markov models, a technique used for image analysis in two dimensions.

11-wavelets,stands for a mathematical function that's used in image compression.

12- self-organizing maps, for classifying images into a number of classes .

13-Rescaling Image (Digital Zoom), Rescaling or re-sampling is the technique used to create a new version of an image with a different size. Increasing the size of the image is called up-sampling, and reducing the size of an image is called down-sampling.

14-Correcting Illumination ,The goal of illumination correction is to remove uneven illumination of the image caused by sensor defaults , non uniform illumination of the scene, or orientation of the objects surface.

prospective Illumination correction is based on background subtraction. This type of correction assumes the scene is composed of an homogeneous background and relatively small objects brighter or darker than the background. There are two major types of background subtraction techniques depending on whether the illumination model of the images can be given as additional images or not: Prospective correction, retrospective correction.

1**-Prospective correction:** Prospective correction uses additional images obtained at the time of image capture. Two types of additional images can be used: A dark image and A bright image

2-Retrospective correction ,When additional image are not available, an ideal illumination model has to be estimated to define the bright image. Therefore, retrospective correction can use the same background subtraction method than the prospective correction with the estimated bright image.

There are different algorithms for estimating the bright image. All of them assume the scene background corresponds to the low frequencies and the objects to the high frequencies. The retrospective correction techniques consist in removing the objects from the background to build the bright image, and then apply the same techniques as the prospective correction.

Applications:

Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecrafts, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics software etc. Image Processing is used in various applications such as:

1 Removing Straight Lines

2 Separating Aggregate of Objects

3 Detecting License Plate

4 Scanning Whiteboard Contents

5 Detecting Text in Still Images

6 Enhancing X-Ray Images

7 Removing Moiré Pattern from Scanned Photos

8 Extracting Urban Areas in Google Maps Aerial Images

9 Extracting Forest Areas in Google Maps Aerial Images

10 Extracting Agricultural Fields in Google Maps Aerial Images

11 Extracting Serous Cell Nuclei

12 Detecting Template in Image

13 Detecting Racing bib number

Naive Bayes

- **What is a classifier?**

A classifier is a machine learning model that is used to discriminate different objects based on certain features.

- **Bayes Theorem:**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

figure 6 bayes theorem

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

- ✓ P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The **evidence** is an attribute value of an unknown instance(here, it is event B).
- ✓ P(A|B) is a **posteriori** probability of B, i.e. probability of event after evidence is seen.

- **Principle of Naive Bayes Classifier:**

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. It works on the principles of conditional probability as given by Bayes theorem.

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

$$P(C_j \mid A_1, A_2, \ldots, A_n) = \frac{\left(\prod_{i=1}^{n} P(A_i \mid C_j)\right) P(C_j)}{P(A_1, A_2, \ldots, A_n)}$$

figure 7 posterior

Types of Naive Bayes classifier:

- **Multinomial Naive Bayes:**

This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

- **Bernoulli Naive Bayes:**

This is similar to the multinomial Naive Bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

- **Gaussian Naive Bayes:**

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

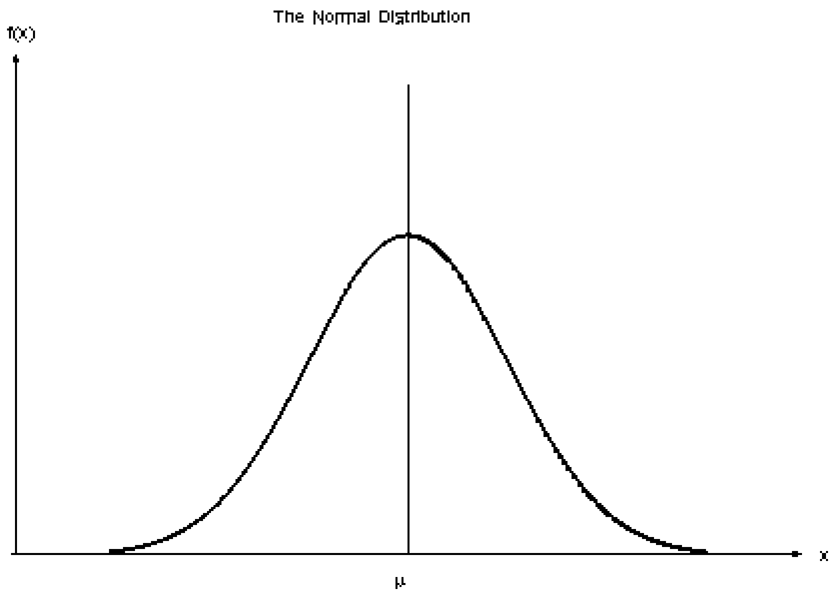


figure 8 Gaussian Distribution(Normal Distribution)

Since the way the values are present in the dataset changes, the formula for conditional probability changes too.

- ## Applications of Naive Bayes Classifier:

✓ Face recognition.
✓ Weather prediction.
✓ News classification.
✓ Medical diagnosis.
✓ Spam filtering.

- ## Advantages of Naive Bayes Classifier:

✓ Simple and easy to implement.
✓ Needs less training data.
✓ Can handle both discrete and continuous data.
✓ Not sensitive to irrelevant features.

Conclusion:

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

Principle Component Analysis (PCA)

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

- **2D Example:**

First, consider a dataset in only two dimensions, like (height, weight). This dataset can be plotted as points in a plane. But if we want to tease out variation, PCA finds a new coordinate system in which every point has a new (x,y) value. The axes don't actually mean anything physical; they're combinations of height and weight called "principal components" that are chosen to give one axes lots of variation.

figure 9 how PCA work

- ## Properties of Principal component (PC):

Technically, a principal component can be defined as a linear combination of optimally-weighted observed variables. The output of PCA are these principal components, the number of which is less than or equal to the number of original variables. Less, in case when we wish to discard or reduce the dimensions in our dataset. The PCs possess some useful properties which are listed below:

1. The PCs are essentially the linear combinations of the original variables, the weights vector in this combination is actually the eigenvector found which in turn satisfies the principle of least squares
2. The PCs are orthogonal.
3. The variation present in the PCs decrease as we move from the 1st PC to the last one, hence the importance.

- ## Advantages of PCA:

1. Principal components help reduce the number of dimensions down to 2 or 3, making it possible to see strong patterns.

2. Principal components take all dimensions and data points into account.
3. Since PC1 and PC2 are perpendicular to each other, we can rotate them and make them straight. These are the axes of our pretty PCA plot.
    - **Principal component 1 (PC1)** :

is a line that goes through the center of a cloud of data and describes it best. It is a line that, if you project the original dots on it, two things happen:

1. The total distance among the projected points is maximum. This means they can be distinguished from one another as clearly as possible. This helps us in comparing data easily.
2. The total distance from the original points to their corresponding projected points is minimum. This means we have a representation that is as close to the original data as possible.

In other words, the best line — our PC1 — must convey the *maximum variation* among data points and contain *minimum error*.

Conclusion:

To sum up, principal component analysis (PCA) is a way to bring out strong patterns from large and complex datasets. The essence of the data is captured in a few principal components, which themselves convey the most variation in the dataset. PCA reduces the number of dimensions without selecting or discarding them. Instead, it constructs principal components that focus on variation and account for the varied

influences of dimensions. Such influences can be traced back from the PCA plot to find out what produces the differences among clusters.

Neural Network

- **What is meant by neural network ?**

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Neural networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on. (Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification and regression.)

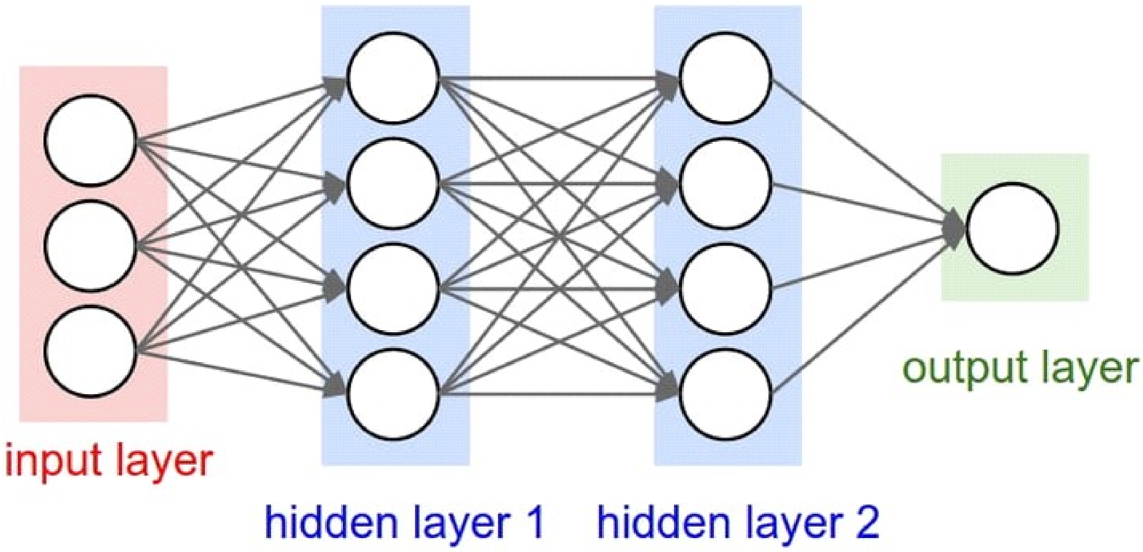


figure 10 Basic idea of how a neural network learns:

Imagine a factory line. After the raw materials (the data set) are input, they are then passed down the conveyer belt, with each subsequent stop or layer extracting a different set of high-level features. If the network is intended to recognize an object, the first layer might analyze the brightness of its pixels.

The next layer could then identify any edges in the image, based on lines of similar pixels. After this, another layer may recognize textures and shapes, and so on. By the time the fourth or fifth layer is reached, the deep learning net will have created complex feature detectors. It can figure out that certain image elements (such as a pair of eyes, a nose, and a mouth) are commonly found together.

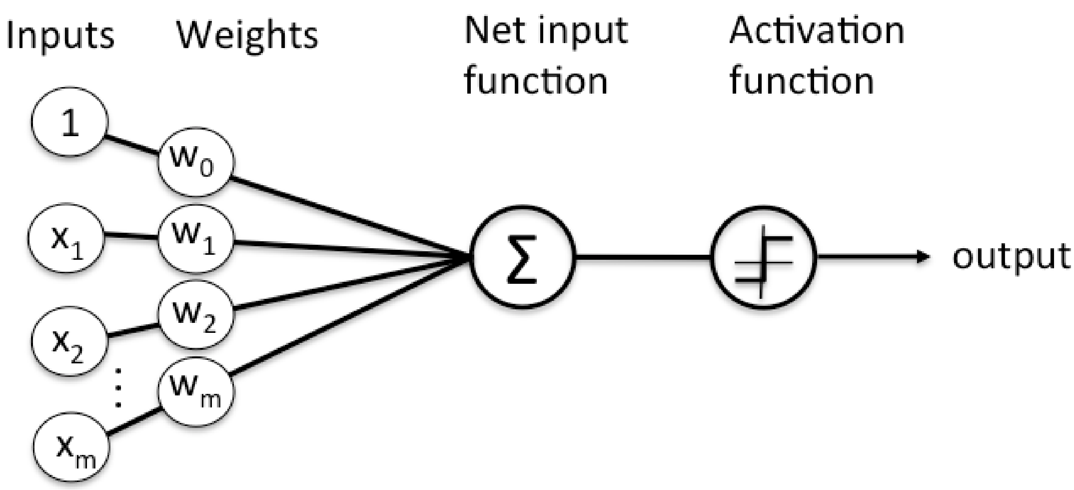


figure 11 example of Neural Network

The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs with regard to the task the algorithm is trying to learn; e.g. which input is most helpful is classifying data without error? These input-weight products are summed and then the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome, say, an act of classification. If the signals passes through, the neuron has been "activated."

Node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

- ## How neural network are trained ?

A neural network is initially trained or fed large amounts of data. Training consists of providing input and telling the network what the output should be. For example, to build a network to identify the faces of actors, initial training might be a series of pictures of actors, nonactors, masks, statuary, animal faces and so on. Each input is accompanied by the matching identification, such as actors' names, "not actor" or "not human" information. Providing the answers allows the model to adjust its internal weightings to learn how to do its job better. For example, if nodes David, Dianne and Dakota tell node Ernie the current input image is a picture of Brad Pitt, but node Durango says it is Betty White, and the training program confirms it is Pitt, Ernie will decrease the weight it assigns to Durango's input and increase the weight it gives to that of David, Dianne and Dakota.

In defining the rules and making determinations -- that is, each node decides what to send on to the next tier based on its own inputs from the previous tier -- neural networks use several principles. These include gradient-based training, fuzzy logic, genetic algorithms and Bayesian methods. They may be given some basic rules about object relationships in the space being modeled. For example, a facial recognition system might be instructed, "Eyebrows are found above eyes," or, "Moustaches are below a nose. Moustaches are above and/or beside a mouth." Preloading rules can make training faster and make the model more powerful sooner.

KNN 'lazy learner'

- **What is KNN algorithm?**

K Nearest Neighbors, is one of the simplest supervised machine learning algorithm mostly used for classification and regression.

- **Why do we need KNN?**

Because KNN is based on feature of similarity, we can do classification using KNN classifier.

- ✓ The simple version of the K-nearest neighbor classifier algorithms is to predict the target label by finding the nearest neighbor class. The closest class will be identified using the distance measures like Euclidean distance.
- ✓ KNN stores all available cases and classifies new cases based on a similarity measure
- ✓ K is a parameter that refers to the number of nearest neighbors to include in the majority voting process
- ✓ A data point is classified by majority votes from its K nearest neighbors

- **How do we choose K?**

A small value of K means that noise will have a higher influence on the result i.e., the probability of overfitting is very high. A large value of K makes it computationally expensive and defeats the basic idea behind KNN (that points that are near might have similar classes). A simple approach to select k is k = n^(1/2).

To optimize the results, we can use Cross Validation. Using the cross-validation technique, we can test KNN algorithm with different values of K. The model which gives good accuracy can be considered to be an optimal choice.

It depends on individual cases, at times best process is to run through each possible value of k and test our result.

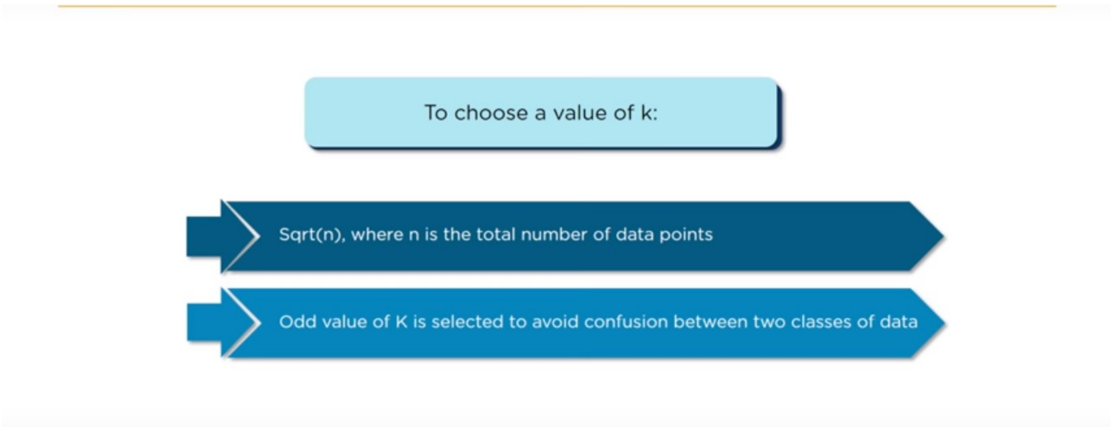


figure 12 properties of K

- ## How does the KNN algorithm work?

In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. Suppose P1 is the point, for which label needs to predict. First, you find the one closest point to P1 and then the label of the nearest point assigned to P1.

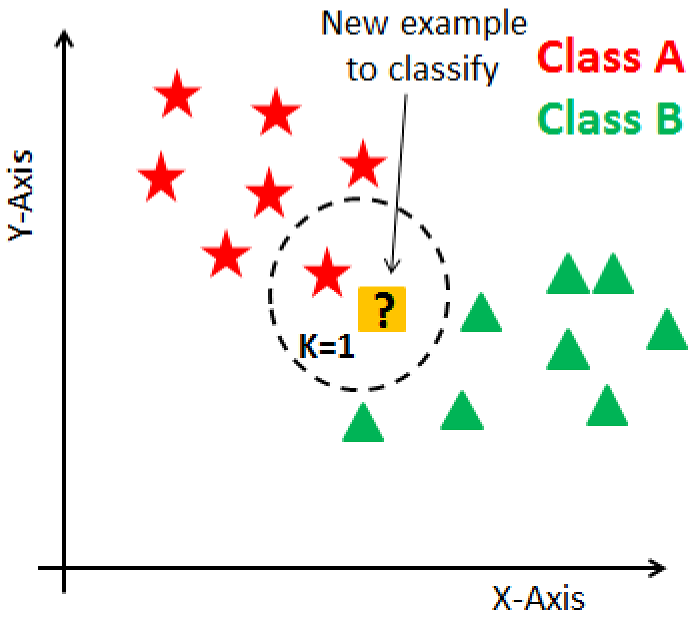


figure 13 example of KNN algorithm

Suppose P1 is the point, for which label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbors. Each object votes for their class and the class with the most votes is taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance. KNN has the following basic steps:

1. Calculate distance
2. Find closest neighbors
3. Vote for labels

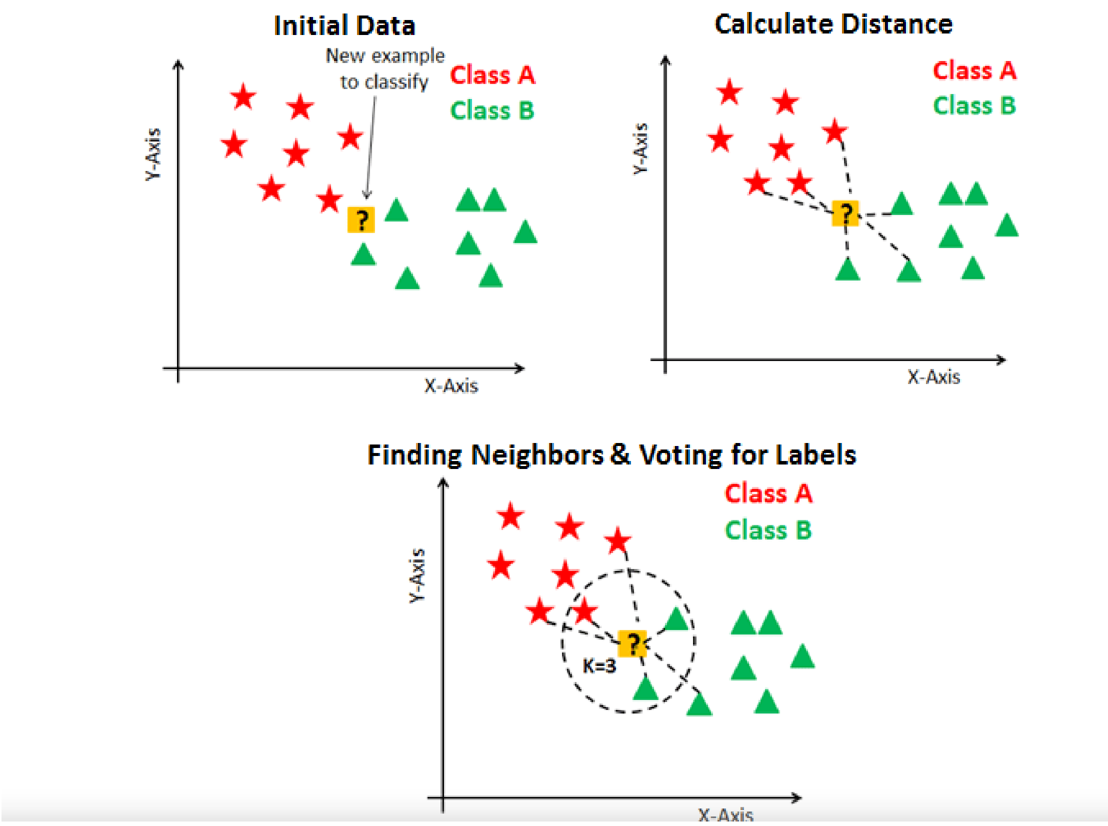


figure 14 how KNN workes

- **When we use KNN?**

1. Data is labeled.
2. Dataset is smal.
3. Data is noise free.

- # Pros and cons of KNN:

  ✓ **Pros**:

a) No assumptions about data—useful, for example, for nonlinear data
b) Simple algorithm—to explain and understand/interpret
c) High accuracy (relatively)—it is pretty high but not competitive in comparison to better supervised learning models
d) Versatile—useful for classification or regression

✓ Cons:

a) Computationally expensive _because the algorithm stores all of the training data_.
b) High memory requirement.
c) Stores all (or almost all) of the training data.
d) Prediction stage might be slow (with big N).
e) Sensitive to irrelevant features and the scale of the data.

- **KNN classification application:**

Let's assume a money lending company "XYZ" like UpStart, IndiaLends, etc. Money lending XYZ company is interested in making the money lending system comfortable & safe for lenders as well as for borrowers. The company holds a database of customer's details.

Using customer's detailed information from the database, it will calculate a credit score(discrete value) for each customer. The calculated credit score helps the company and lenders to understand the credibility of a customer clearly. So they can simply take a decision whether they should lend money to a particular customer or not.

• The customer's details could be:

✓ Educational background details.

a) Highest graduated degree.
b) Cumulative grade points average (CGPA) or marks percentage.
c) The reputation of the college.
d) Consistency in his lower degrees.
e) Whether to take the education loan or not.
f) Cleared education loan dues.

✓ Employment details.

a) Salary.
b) Year of experience.
c) Got any onsite opportunities.
d) Average job change duration.

The company(XYZ) use's these kinds of details to calculate credit score of a customer. The process of calculating the credit score from the customer's details is expensive. To reduce the cost of predicting credit score, they realized that the customers with similar background details are getting a similar credit score.

So, they decided to use already available data of customers and predict the credit score using it by comparing it with similar data. These kinds of problems are handled by the k-nearest neighbor classifier for finding the similar kind of customers.

- **K-nearest neighbor (KNN) algorithm pseudocode:**

Let (X_i, C_i) where i = 1, 2……., n be data points. X_i denotes feature values &C_i denotes labels for $X_i$for each i.
Assuming the number of classes as 'c'
$C_i \in \{1, 2, 3, \ldots\ldots, c\}$ for all values of i

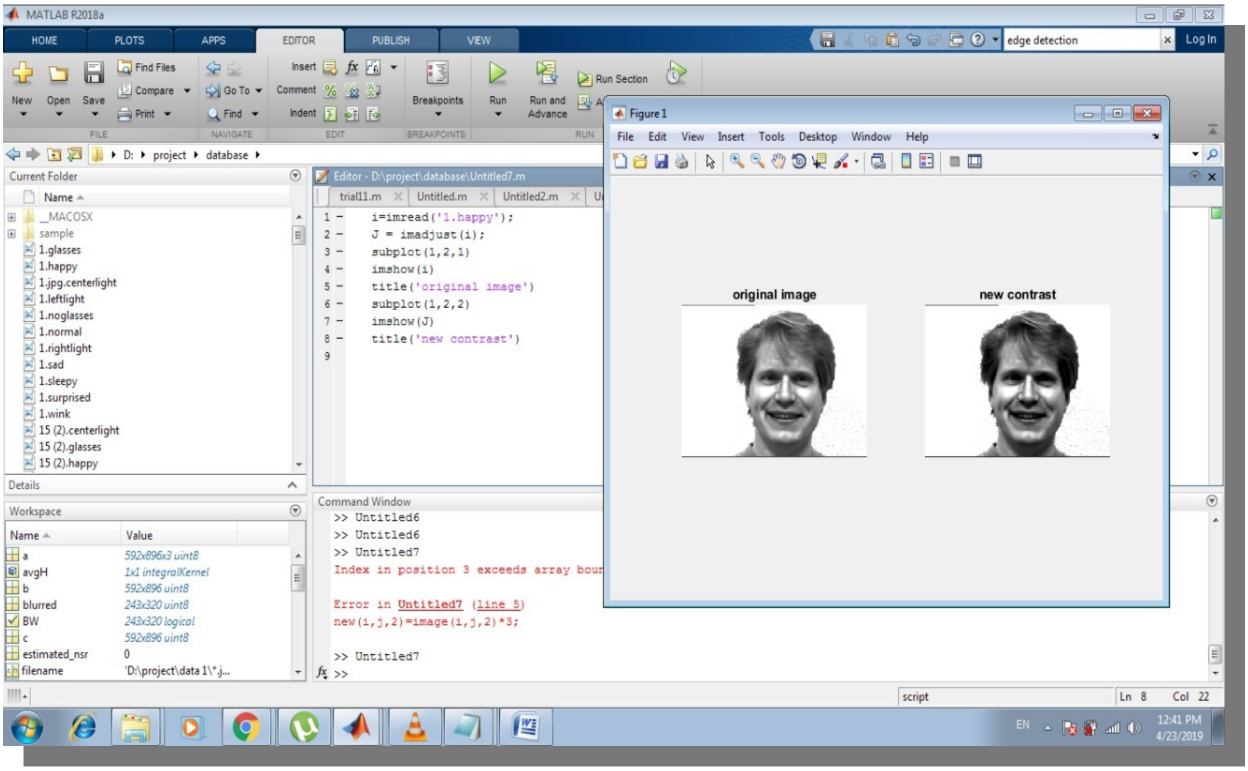
Let x be a point for which label is not known, and we would like to find the label class using k-nearest neighbor algorithms.

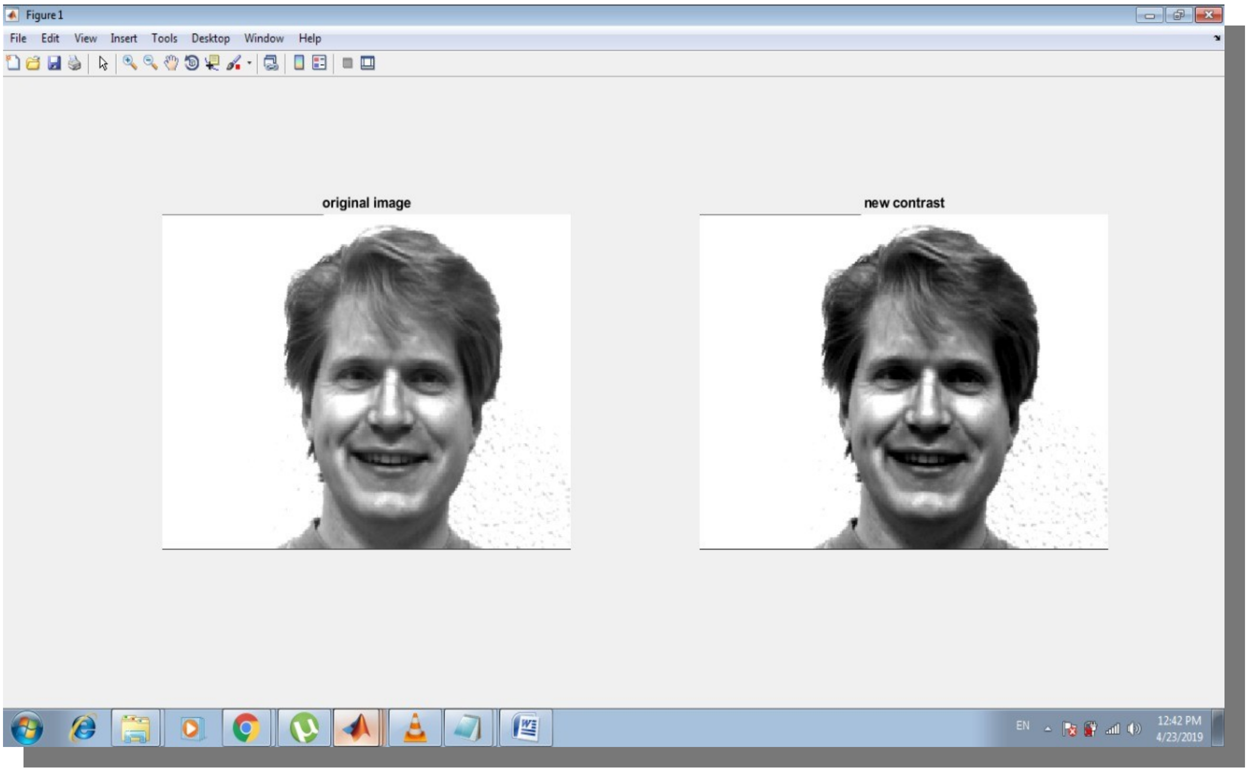
- **KNN Algorithm pseudocode:**

1. Calculate "$d(x, x_i)$" i =1, 2, ….., **n**; where **d** denotes the Euclidean distance between the points.
2. Arrange the calculated **n** Euclidean distances in non-decreasing order.
3. Let **k** be a +ve integer, take the first **k** distances from this sorted list.
4. Find those **k**-points corresponding to these **k**-distances.
5. Let $\mathbf{k}_i$ denotes the number of points belonging to the i^{th} class among **k** points i.e. $k \geq 0$
6. If $k_i > k_j \ \forall \ i \neq j$ then put x in class i.

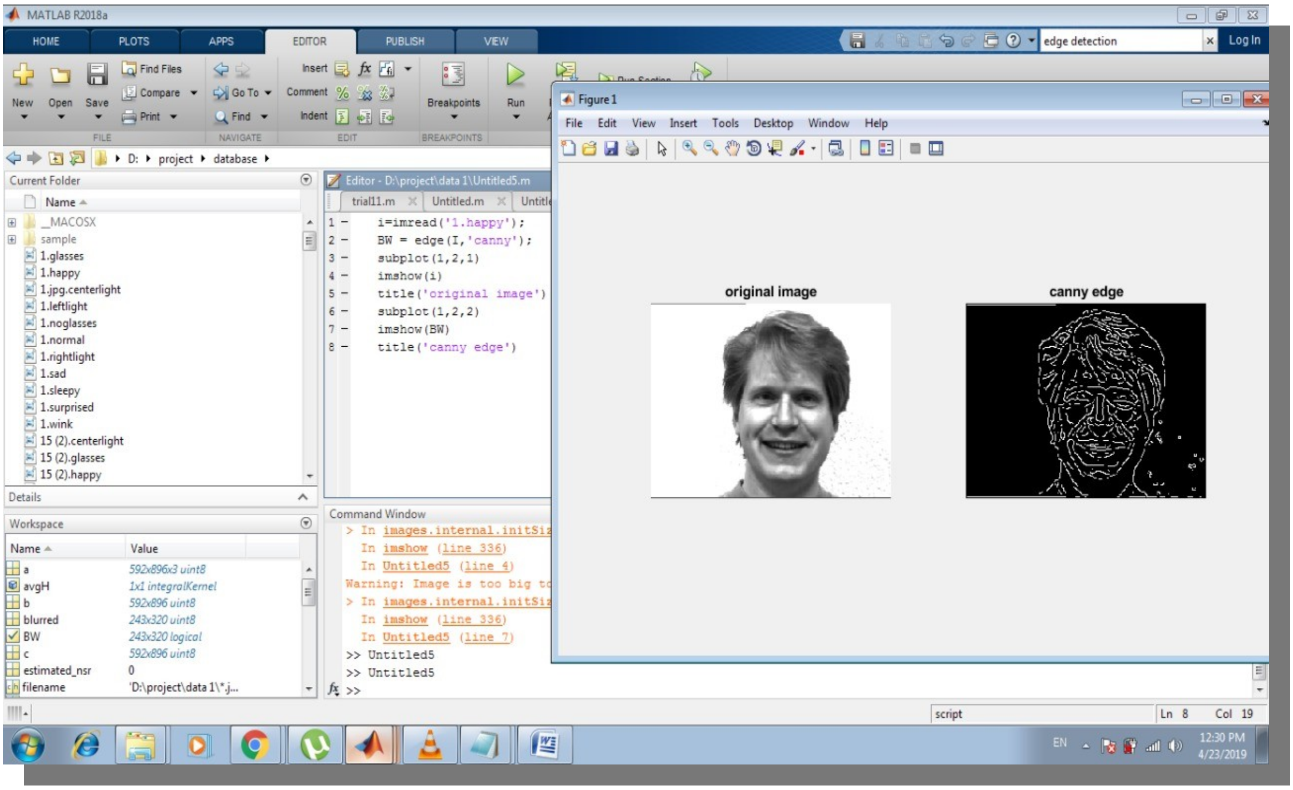
Code Filters

Changing the contrast of an image:





Edge detection:



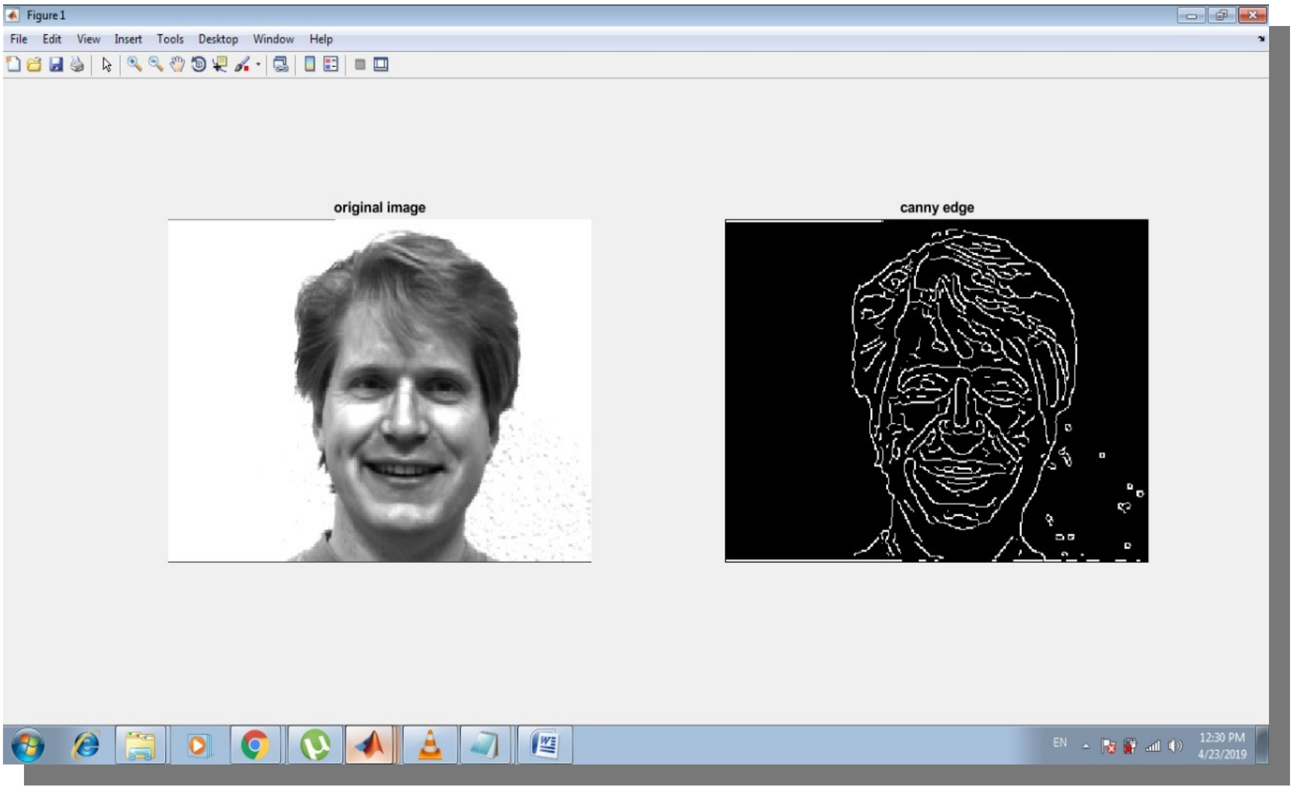
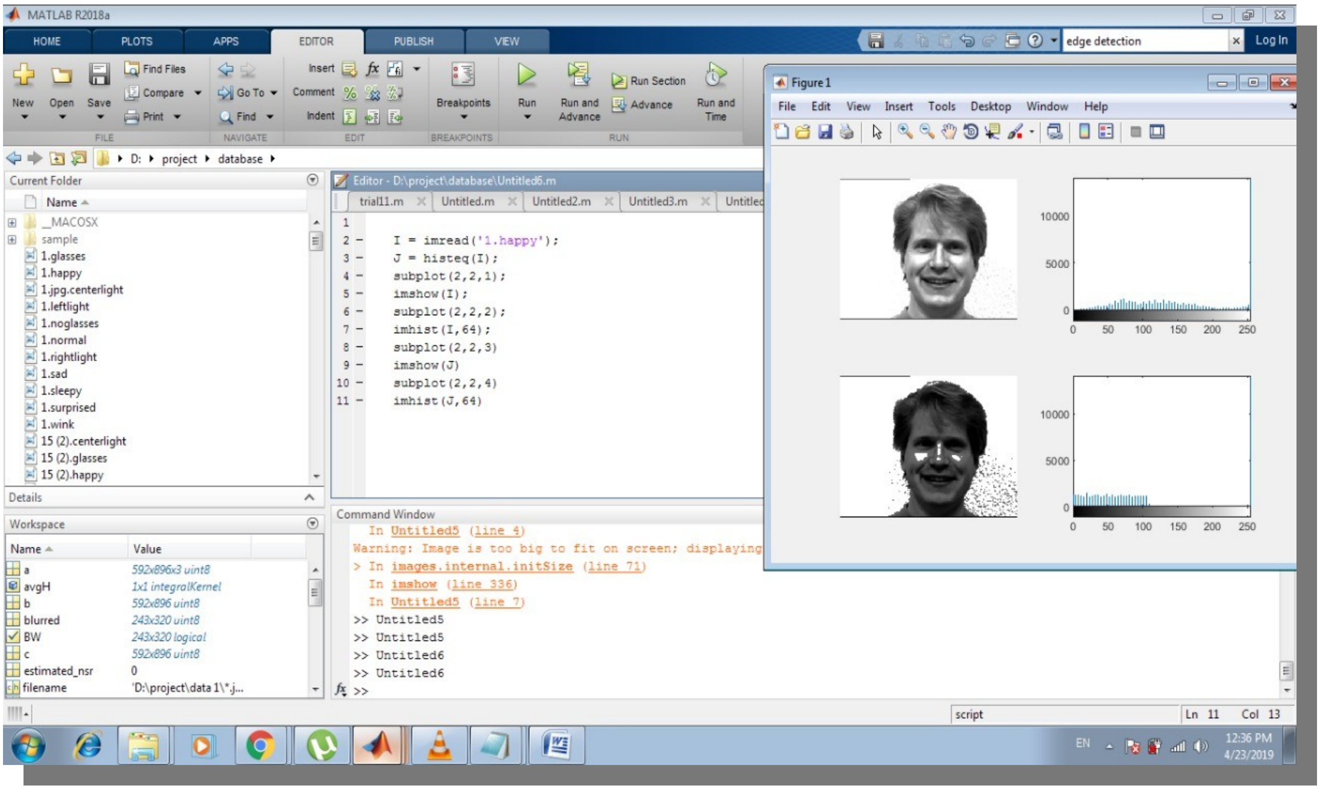
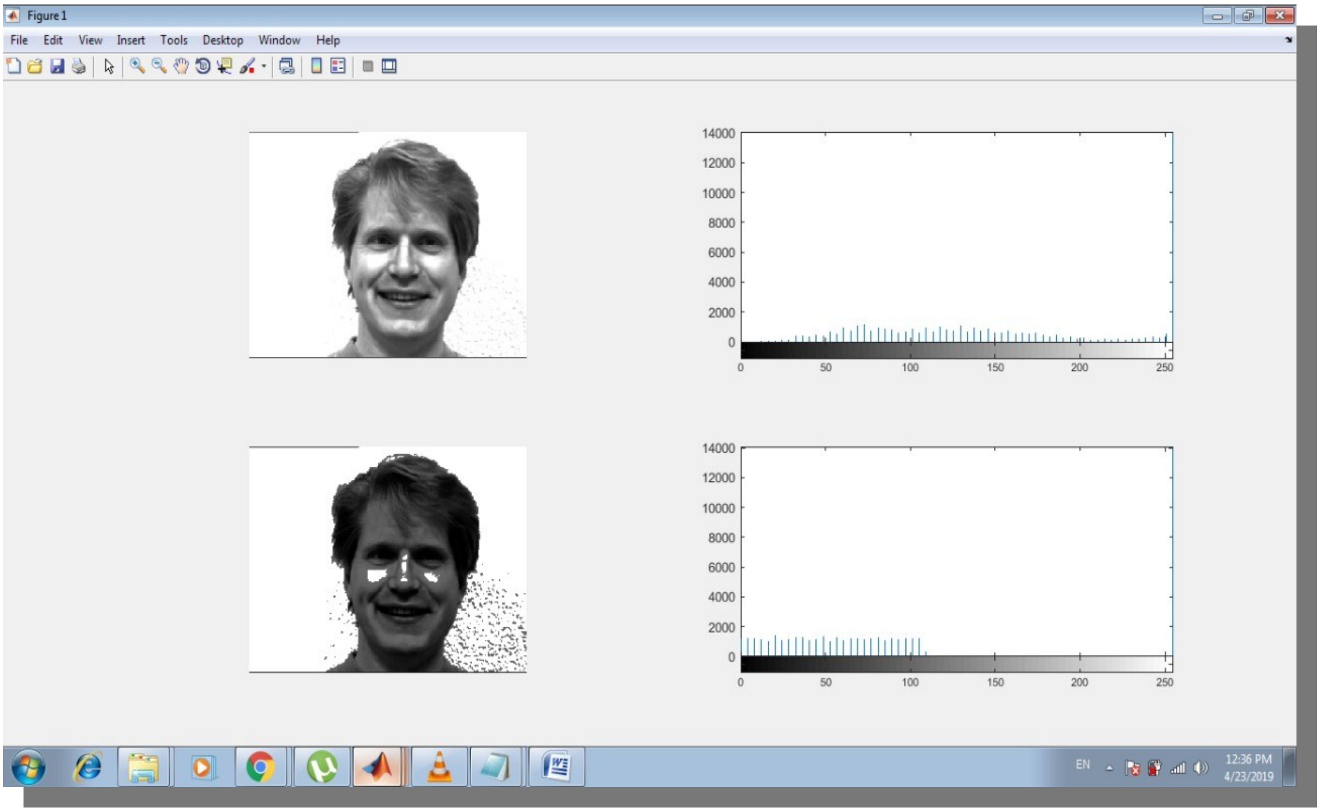


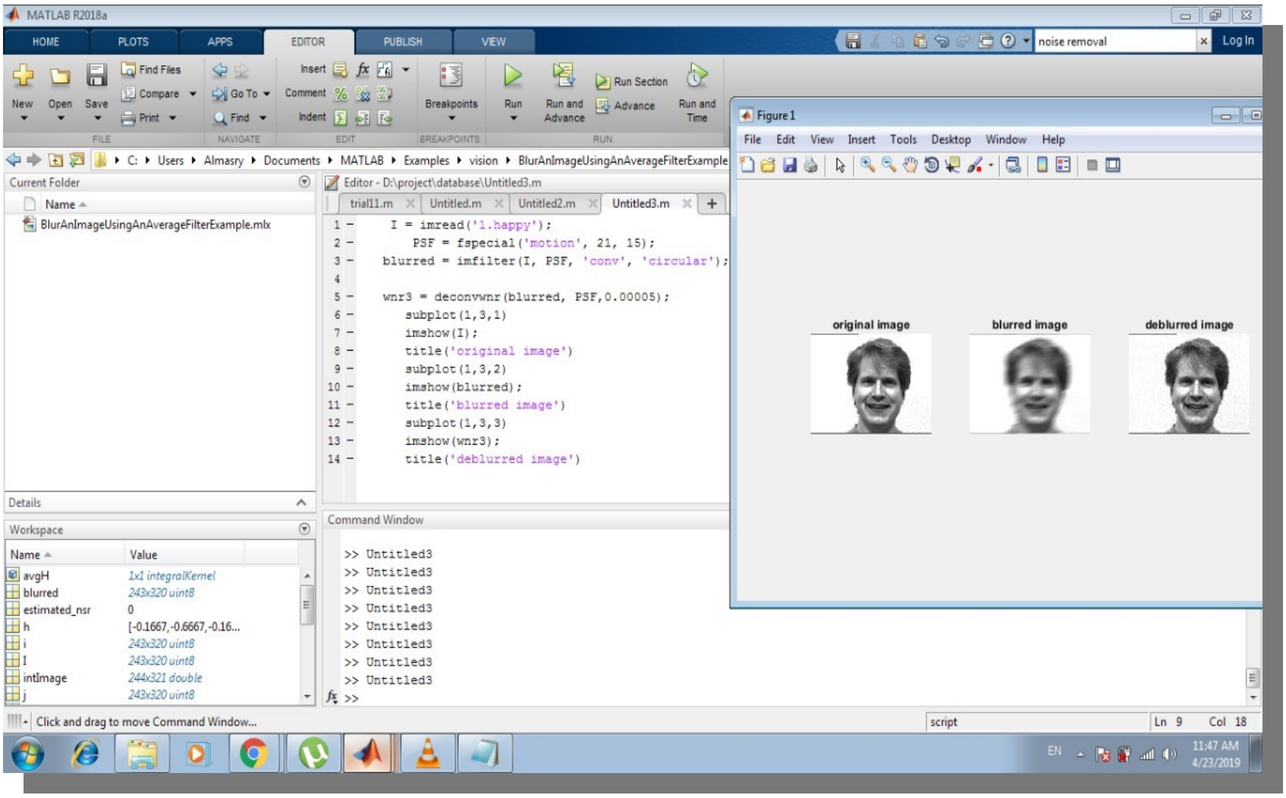
Figure 1
File Edit View Insert Tools Desktop Window Help
original image
canny edge
EN
12:30 PM
4/23/2019

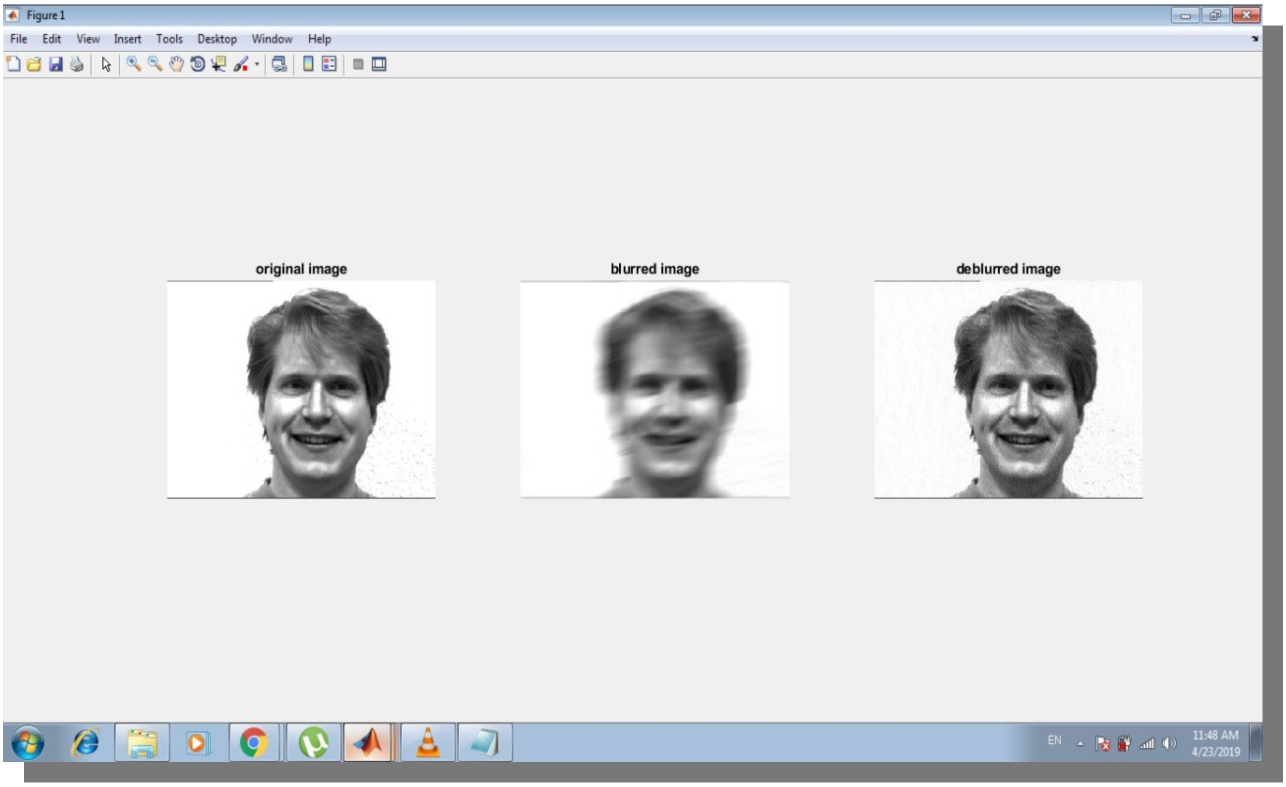
Histogram Equalization:





Blurring and de-blurring on an image:

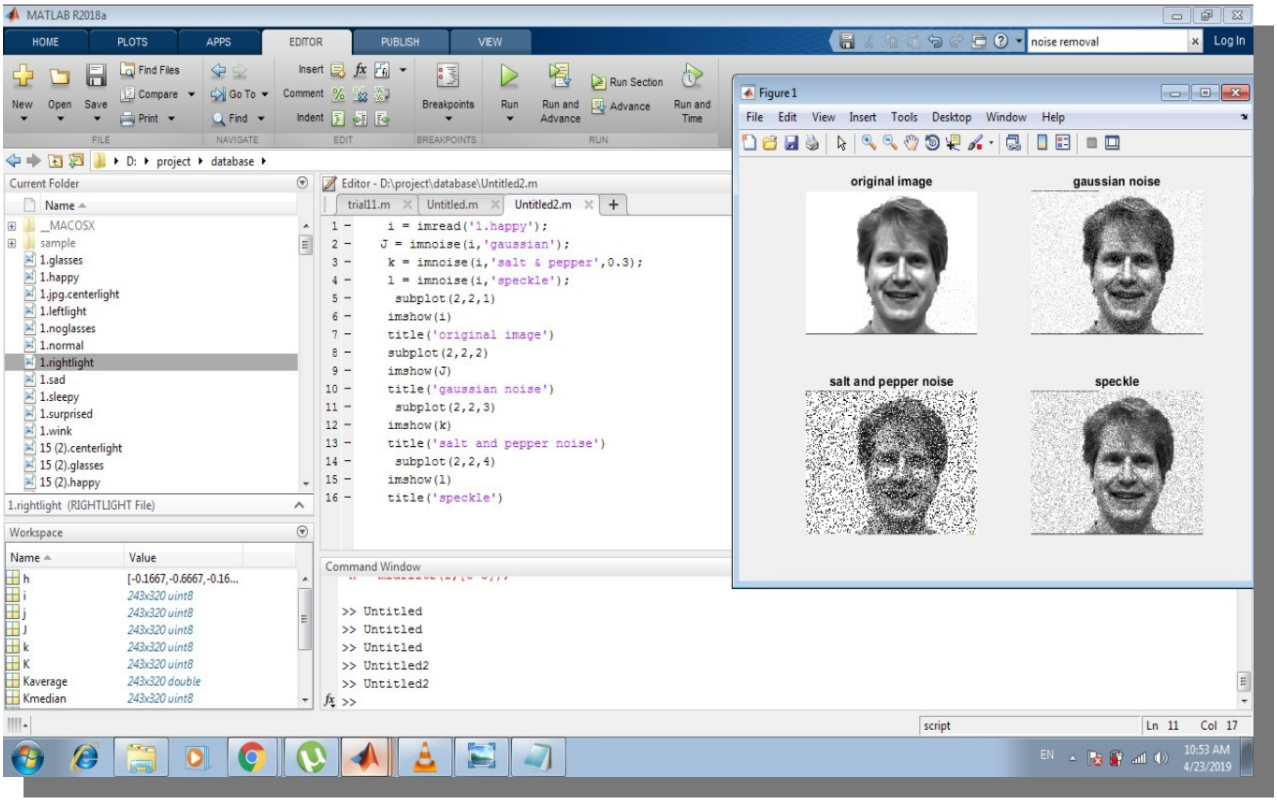


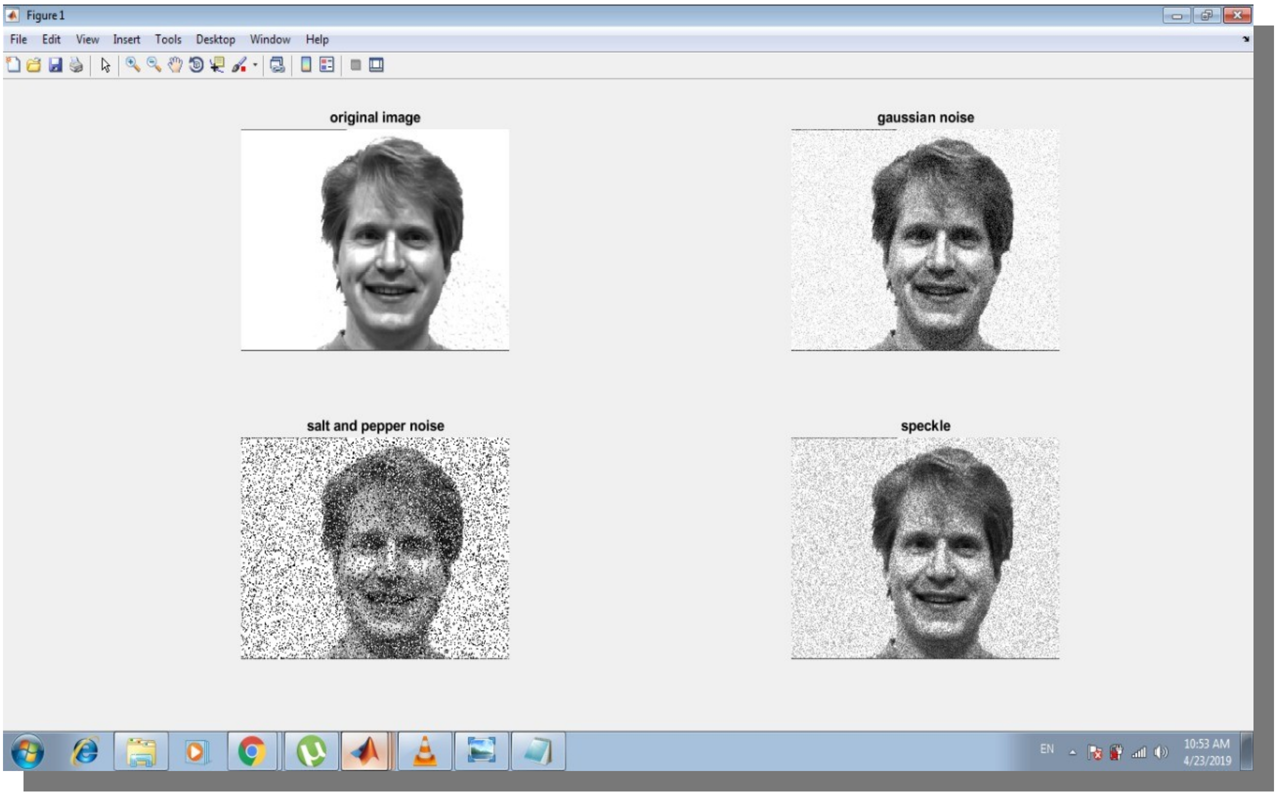


Applying and Removing noise:

Applying Noise:

- **salt and pepper noise:** An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions . This type of noise can be caused by dead pixels, analog-to-digital converter errors, bit errors in transmission, etc. This can be eliminated in large part by using dark frame subtraction and by interpolating around dark/bright pixels.
- **Gaussian noise**: is a major part of the "read noise" of an image sensor, that is, of the constant noise level in dark areas of the image .
- **Speckle noise:** Speckle noise is a granular noise that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images. Speckle noise in conventional radar results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of a local area.



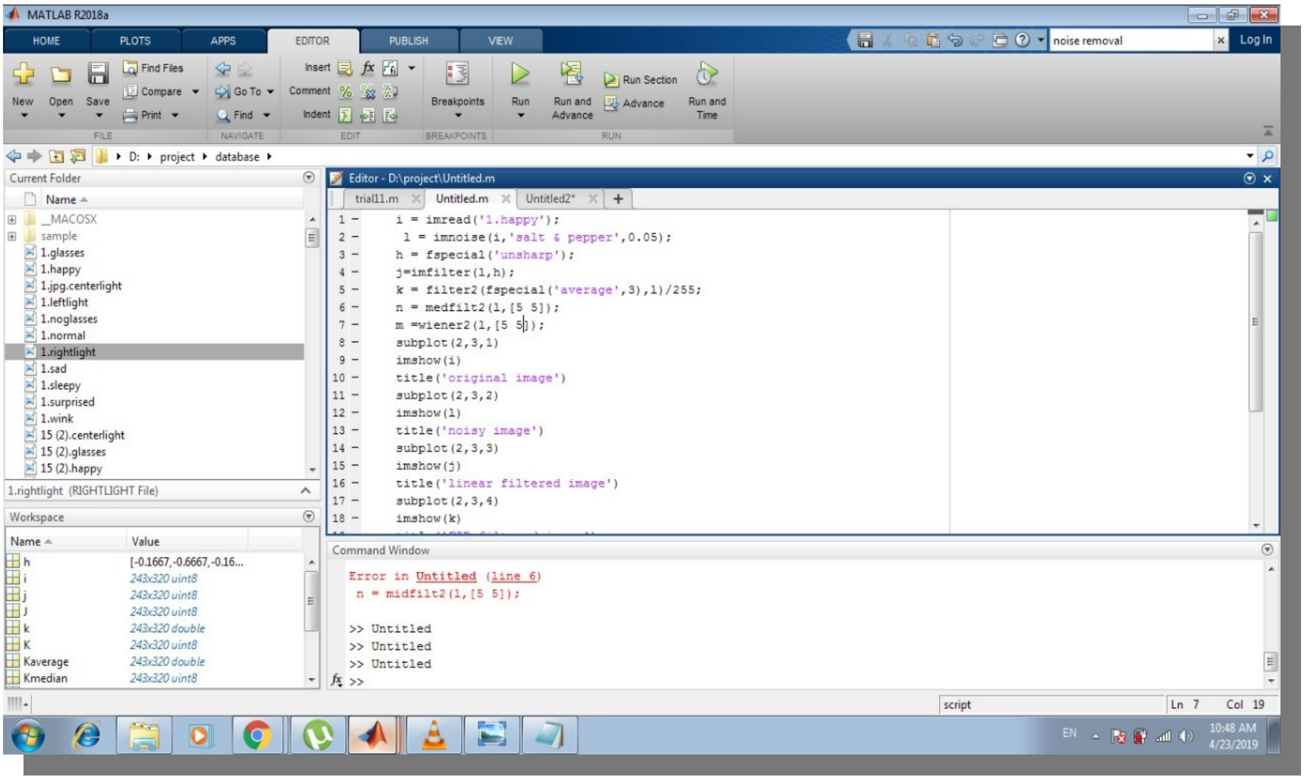


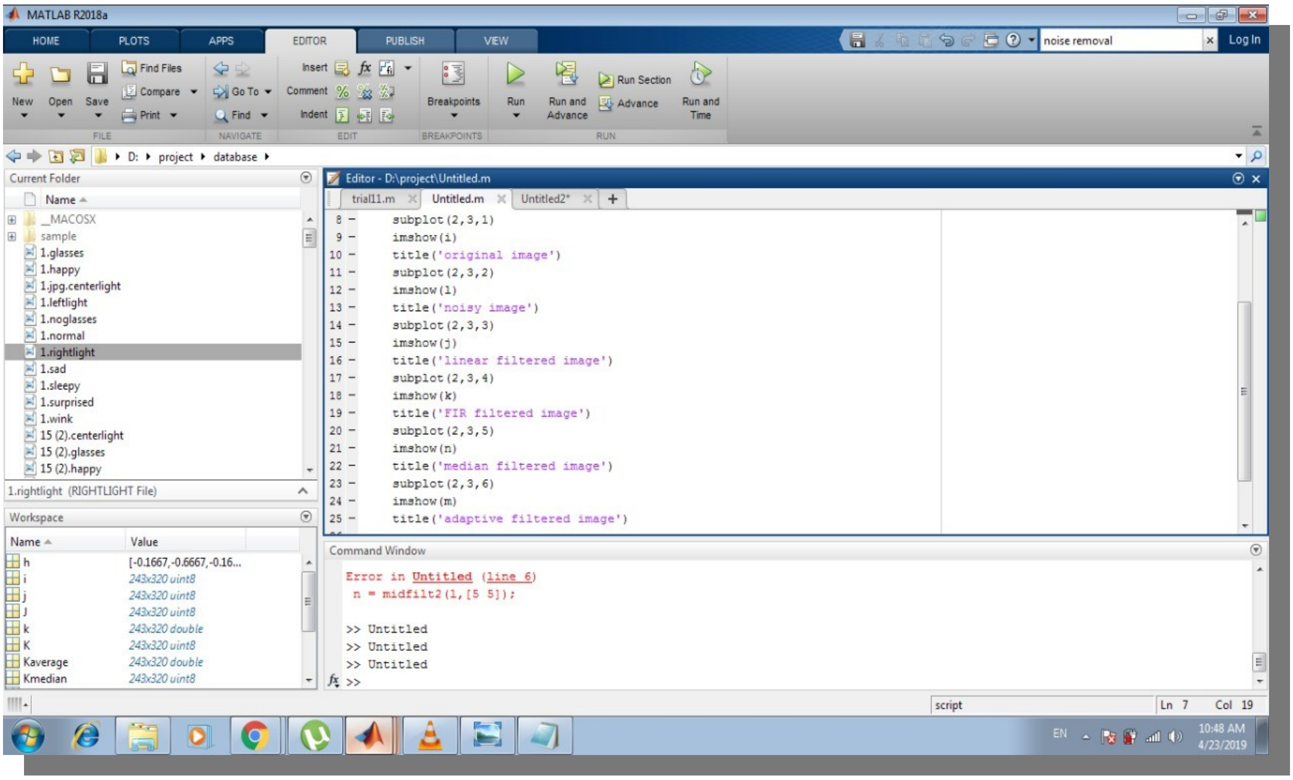
Removing Noise:

Linear Filtering: We can use linear filtering to remove certain types of noise. Certain filters, such as averaging or Gaussian filters, are appropriate for this purpose. For example, an averaging filter is useful for removing grain noise from a photograph.

Adaptive Filtering: Median filtering is similar to using an averaging filter, in that each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the median of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called outliers). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image. The medfilt2 function implements median filtering.

Median Filtering: The wiener2 function applies a Wiener filter (a type of linear filter) to an image adaptively, tailoring itself to the local image variance. Where the variance is large, wiener2 performs little smoothing. Where the variance is small, wiener2 performs more smoothing. This approach often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high-frequency parts of an image. wiener2 works best when the noise is constant-power ("white") additive noise, such as Gaussian noise




MATLAB R2018a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
noise removal
Log In
Editor - D:\project\Untitled.m
triall1.m
Untitled.m
Untitled2*
subplot(2,3,1)
imshow(i)
title('original image')
subplot(2,3,2)
imshow(l)
title('noisy image')
subplot(2,3,3)
imshow(j)
title('linear filtered image')
subplot(2,3,4)
imshow(k)
title('FIR filtered image')
subplot(2,3,5)
imshow(n)
title('median filtered image')
subplot(2,3,6)
imshow(m)
title('adaptive filtered image')
Command Window
Error in Untitled (line 6)
n = midfilt2(l,[5 5]);
>> Untitled
>> Untitled
>> Untitled
Current Folder
Workspace

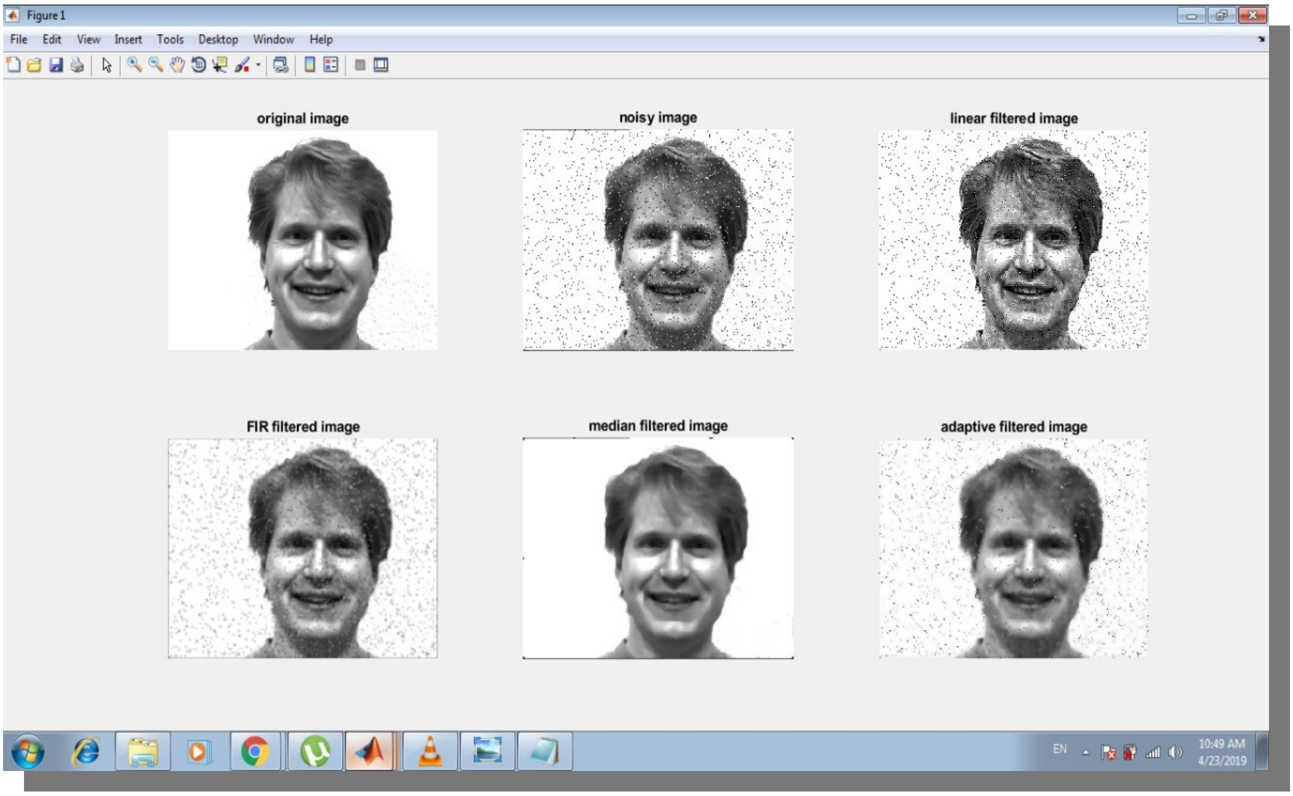
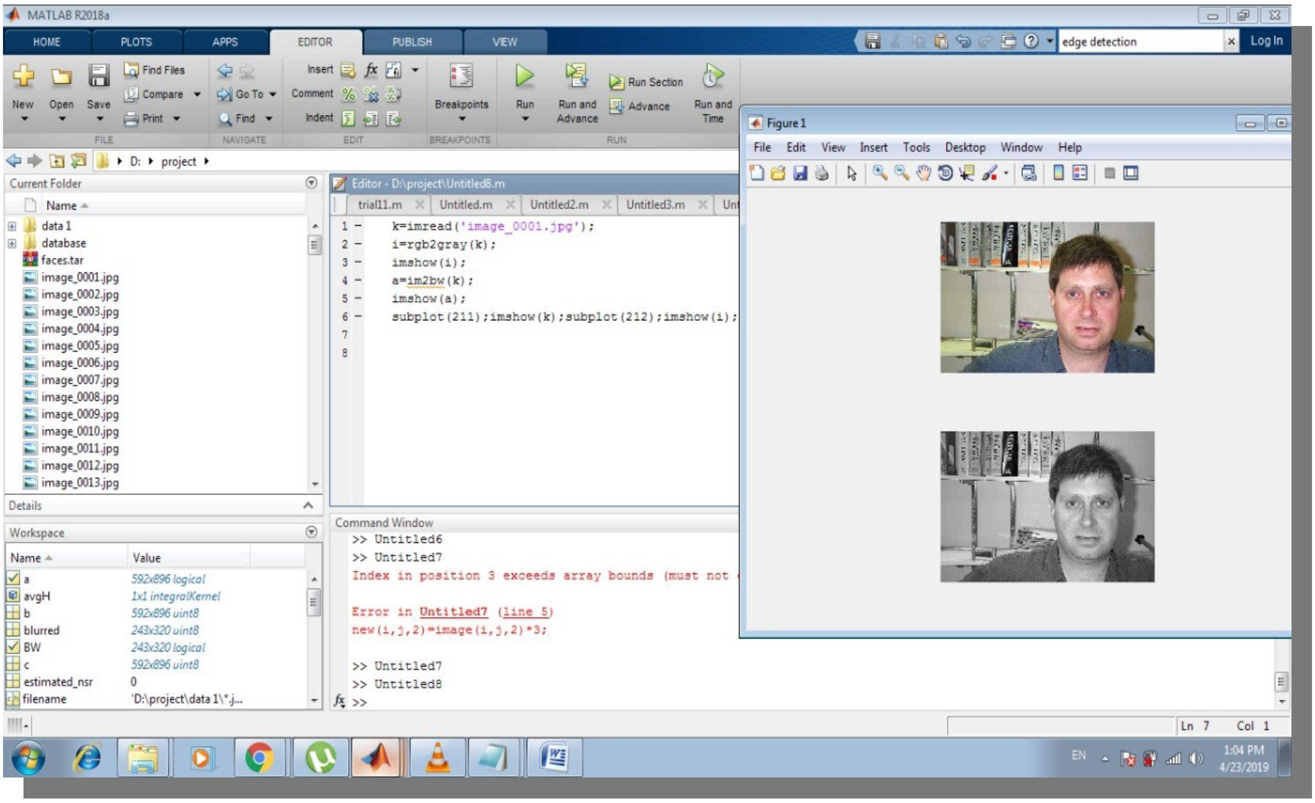


Figure 1
File Edit View Insert Tools Desktop Window Help
original image
noisy image
linear filtered image
FIR filtered image
median filtered image
adaptive filtered image

Changing an image from RGB to black and white:



Database:

The Yale Face Database contains 127 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, with no glasses, normal, right-light, sad, sleepy, surprised, and wink.

Expressions:

The yale database contains a variety of expressions that allow us to properly train our program to recognize the person in the picture and the his expression.

Final code:

Let a face image Γ(x, y) be a two dimensional M by N array of intensity values. In this thesis.

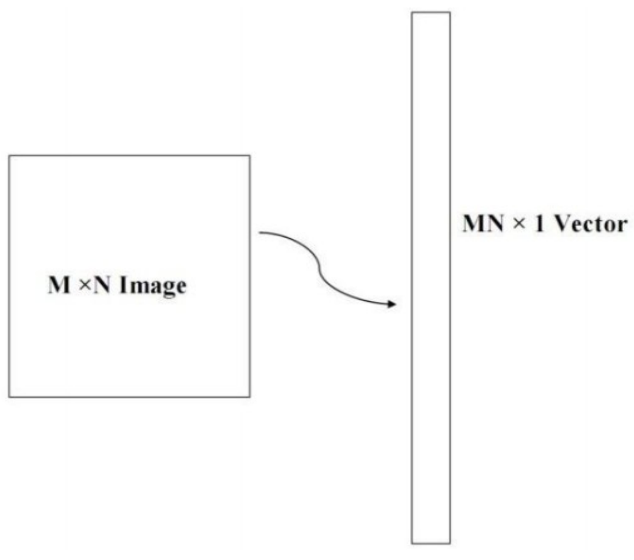


figure 15 1 Conversion of M × N image into MN ×1 vector

- **Step1:** prepare the training faces Obtain face images I1, I2, I3, I4 , ... IM (training faces), The face images must be centered and of the same size, , I used a set of image by 243×320 pixels. An image may also be considered as a vector of dimension M × N, so that a typical image of size 243× 320 becomes a vector of dimension 77,760 or equivalently a point in a 77,760 dimensional space.
- **Step 2:** Prepare the data set Each face image Ii in the database is transformed into a vector , using reshape function ,and placed into a training set S.

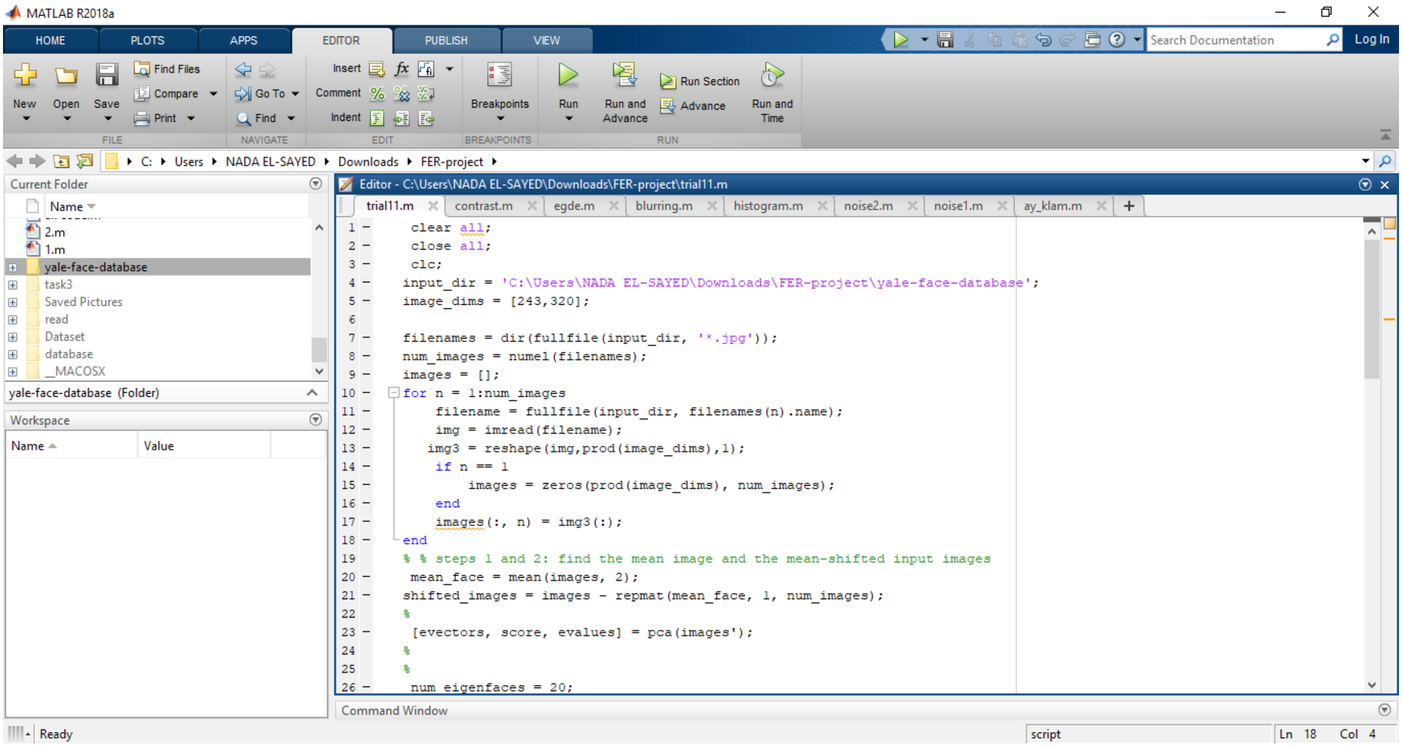
S=[I1, I2, I3, I4 , ... IM]

In My example M = 134. Each image is transformed into a vector of size MN × 1 and placed into the set.

- **Step 3**: compute the average face vector The average face mean(M) has to be calculated by using mean function on MATLAB.
- **Step 4:** The mean face vector is subtracted from the original faces and the result stored in the variable (shifted_images) .
- **Step 5:** find first 20commen features using PCA function**.**
- **Step 6:** project the images into the subspace to generate the eigenvectors.
- **Step 7:** input test image .
- **Step 8:** reshape the image to vector .
- **Step 9:** The mean face vector is subtracted from the test image.
- **Step 10:** match the test image.

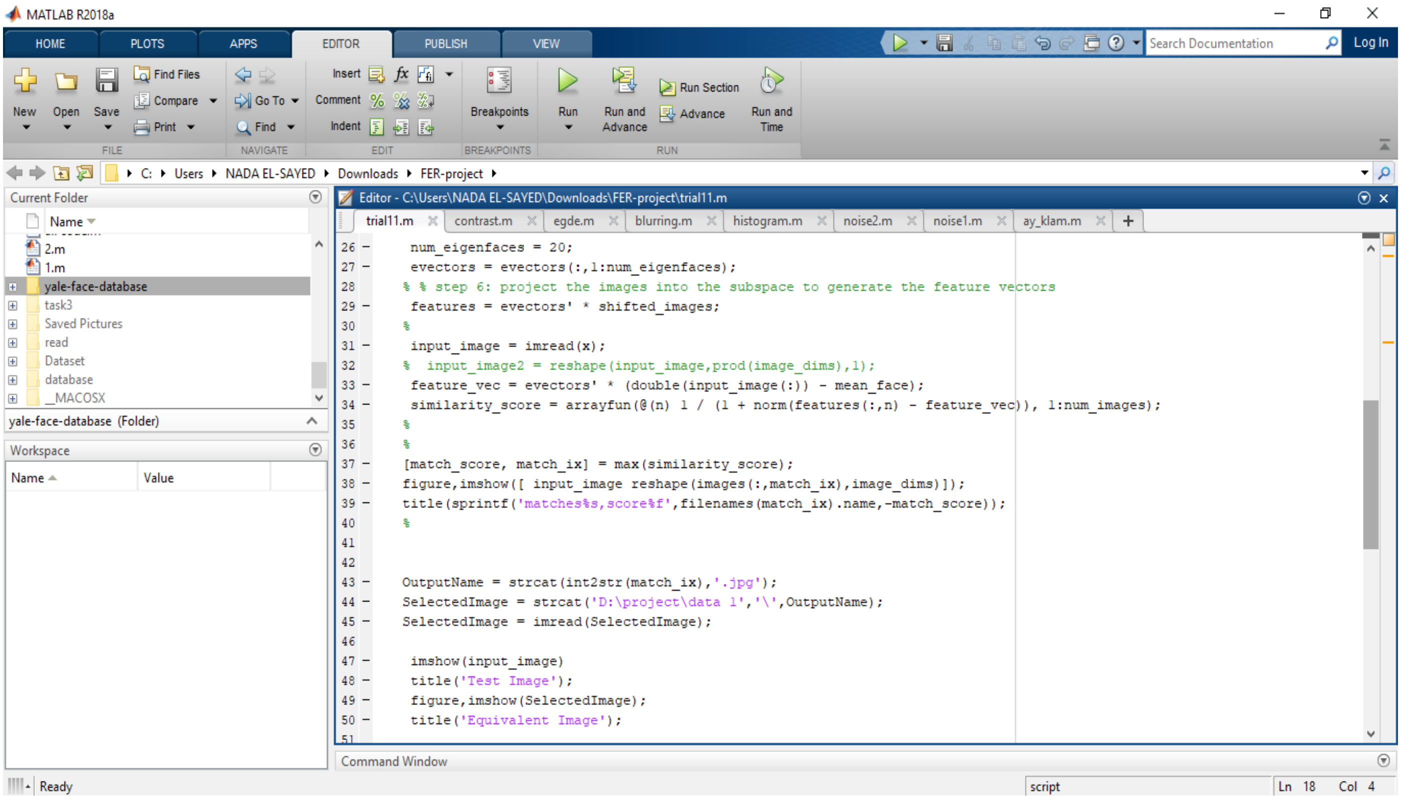
we trained our program in two different ways.

1. to recognize the person with the same expression.
2. to recognize the same person with a different expression.



```
clear all;
close all;
clc;
input_dir = 'C:\Users\NADA EL-SAYED\Downloads\FER-project\yale-face-database';
image_dims = [243,320];

filenames = dir(fullfile(input_dir, '*.jpg'));
num_images = numel(filenames);
images = [];
for n = 1:num_images
    filename = fullfile(input_dir, filenames(n).name);
    img = imread(filename);
   img3 = reshape(img,prod(image_dims),1);
    if n == 1
        images = zeros(prod(image_dims), num_images);
    end
    images(:, n) = img3(:);
end
% % steps 1 and 2: find the mean image and the mean-shifted input images
 mean_face = mean(images, 2);
shifted_images = images - repmat(mean_face, 1, num_images);
%
 [evectors, score, evalues] = pca(images');
%
%
 num_eigenfaces = 20;
```



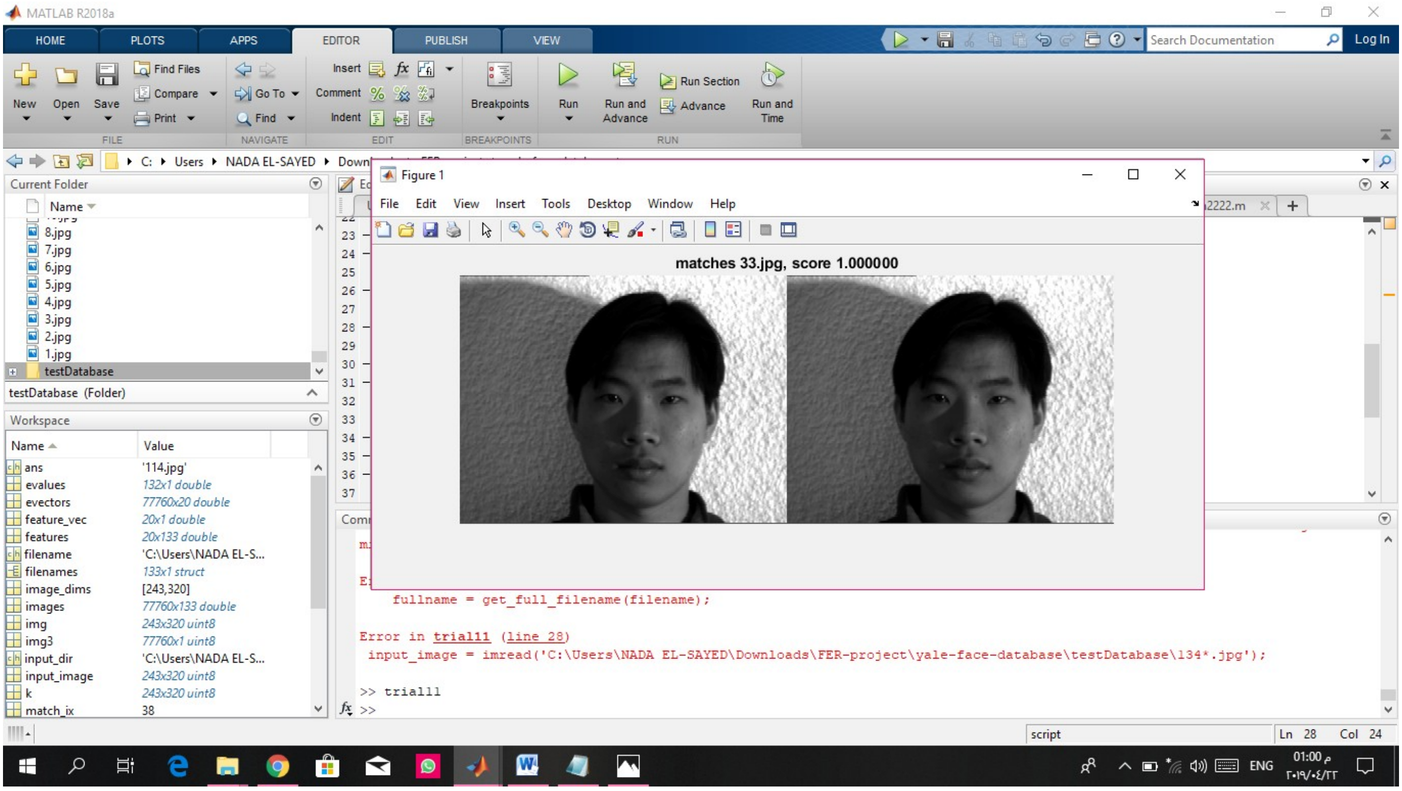
```
 num_eigenfaces = 20;
 evectors = evectors(:,1:num_eigenfaces);
% % step 6: project the images into the subspace to generate the feature vectors
 features = evectors' * shifted_images;
%
 input_image = imread(x);
%  input_image2 = reshape(input_image,prod(image_dims),1);
 feature_vec = evectors' * (double(input_image(:)) - mean_face);
 similarity_score = arrayfun(@(n) 1 / (1 + norm(features(:,n) - feature_vec)), 1:num_images);
%
%
[match_score, match_ix] = max(similarity_score);
figure,imshow([ input_image reshape(images(:,match_ix),image_dims)]);
title(sprintf('matches%s,score%f',filenames(match_ix).name,-match_score));
%


OutputName = strcat(int2str(match_ix),'.jpg');
SelectedImage = strcat('D:\project\data 1','\',OutputName);
SelectedImage = imread(SelectedImage);

 imshow(input_image)
 title('Test Image');
 figure,imshow(SelectedImage);
 title('Equivalent Image');
```

Results:

1. Here we tested the ability of the program to identify the right person with the same expression.



2. Here we tested the ability of the program to identify the right person with a different expression. we did so by removing the test images from the database and locating them in a different folder. the program successfully identified the right person from the database to fit the test image.

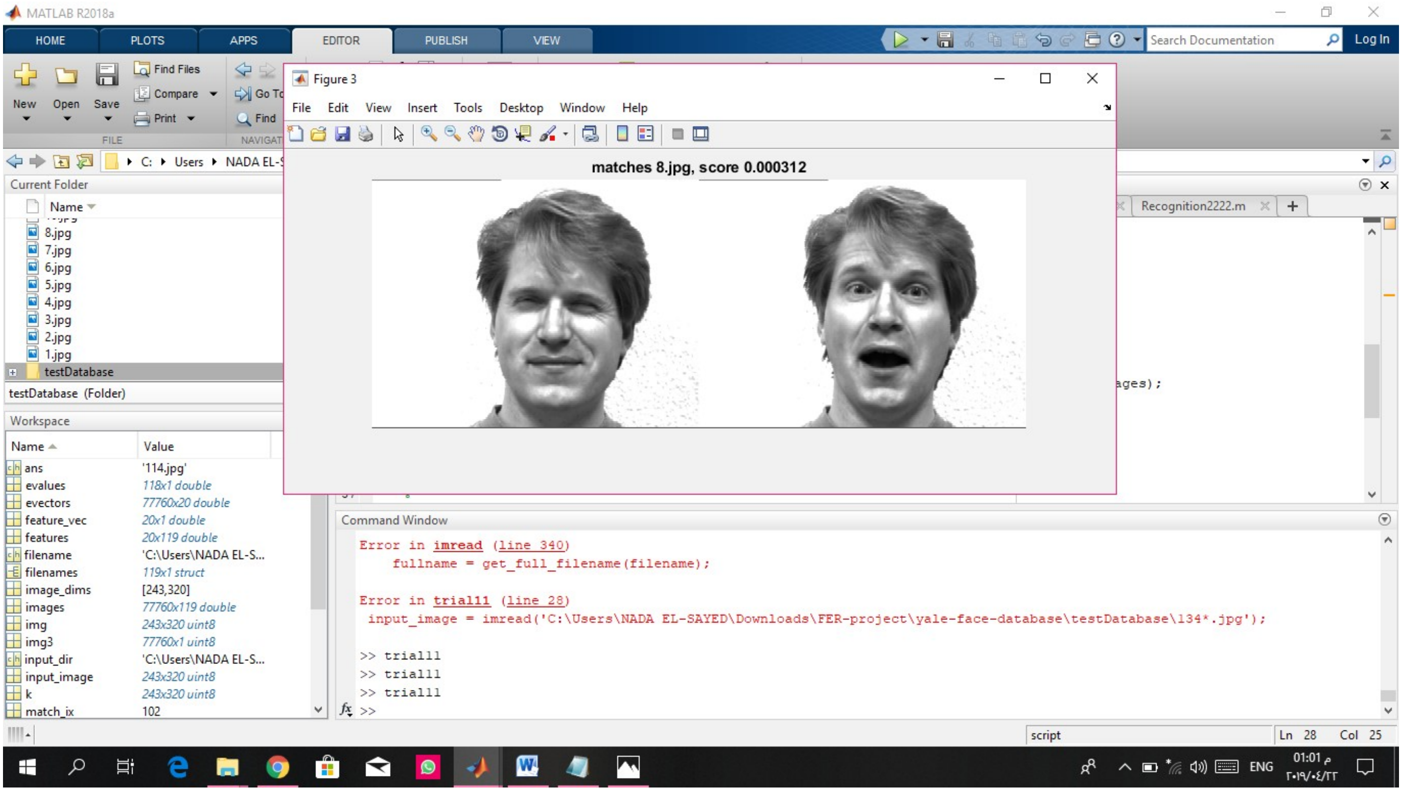
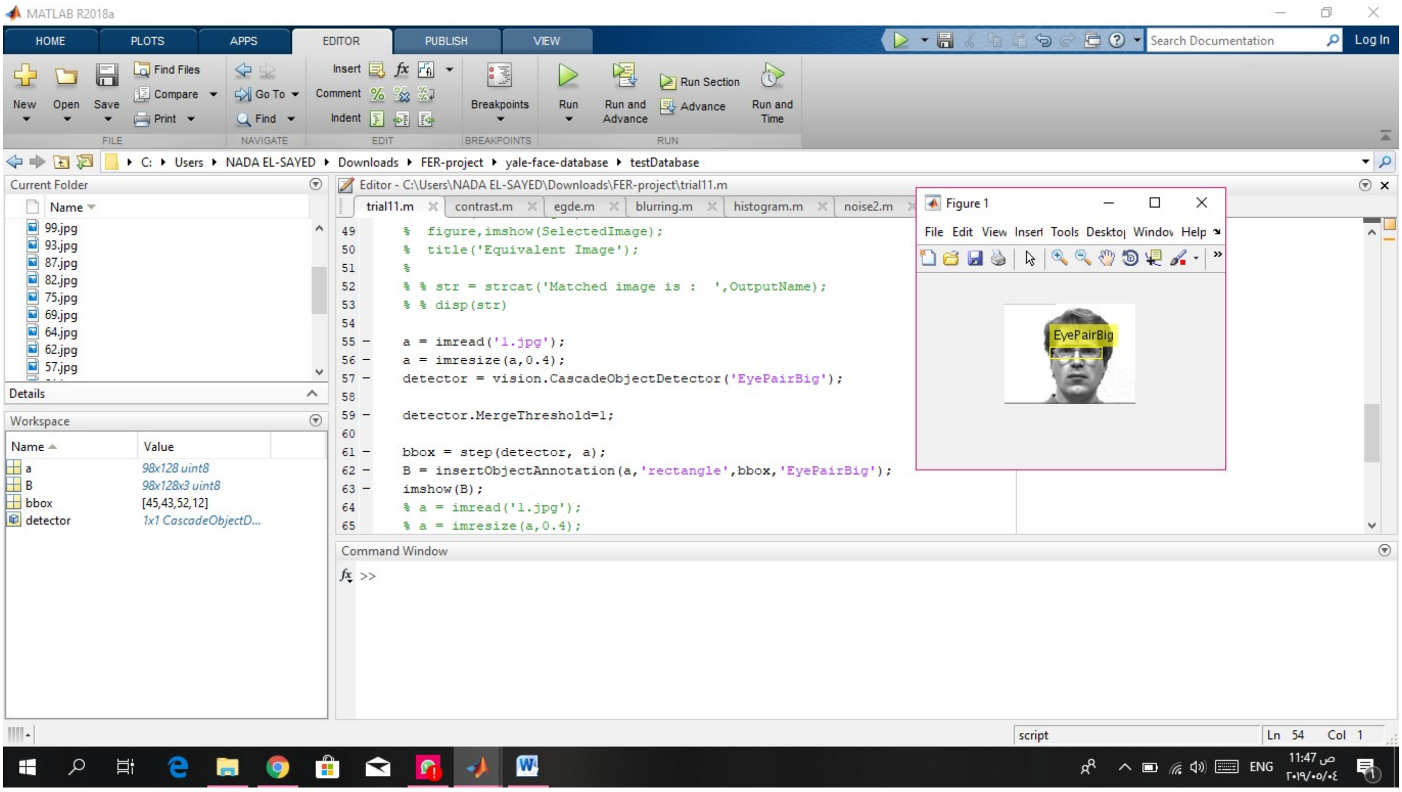
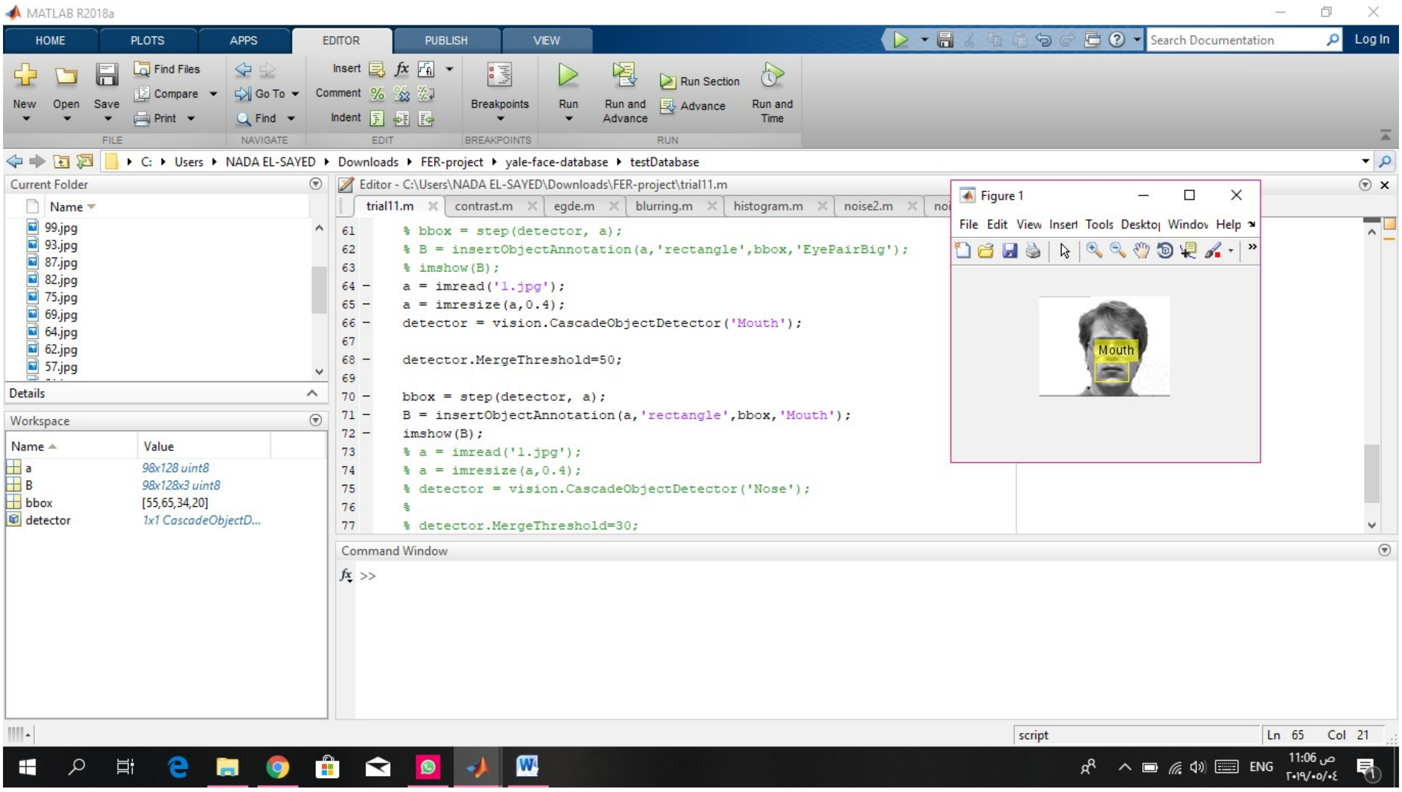


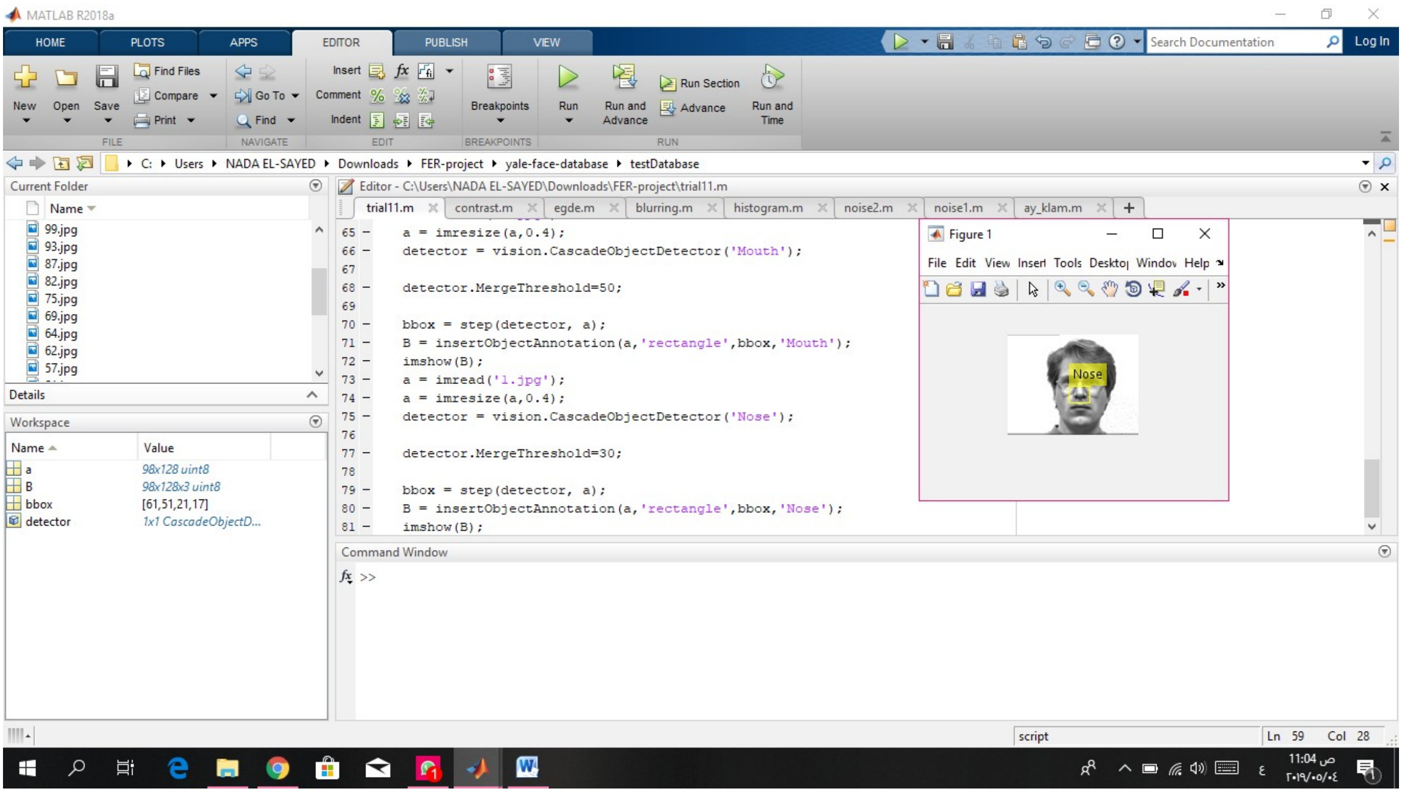
Figure 3
matches 8.jpg, score 0.000312
Error in imread (line 340)
fullname = get_full_filename(filename);
Error in triall1 (line 28)
input_image = imread('C:\Users\NADA EL-SAYED\Downloads\FER-project\yale-face-database\testDatabase\134*.jpg');

Face deduction:

We decided to take our project one step further by implementing a feature detection option. It can successfully detect the mouth, nose, and eyes on the input image.




MATLAB R2018a
Editor - C:\Users\NADA EL-SAYED\Downloads\FER-project\trial11.m
% bbox = step(detector, a);
% B = insertObjectAnnotation(a,'rectangle',bbox,'EyePairBig');
% imshow(B);
a = imread('1.jpg');
a = imresize(a,0.4);
detector = vision.CascadeObjectDetector('Mouth');
detector.MergeThreshold=50;
bbox = step(detector, a);
B = insertObjectAnnotation(a,'rectangle',bbox,'Mouth');
imshow(B);
% a = imread('1.jpg');
% a = imresize(a,0.4);
% detector = vision.CascadeObjectDetector('Nose');
%
% detector.MergeThreshold=30;
Figure 1
Mouth
Workspace
a 98x128 uint8
B 98x128x3 uint8
bbox [55,65,34,20]
detector 1x1 CascadeObjectD...
Command Window


MATLAB R2018a
Editor - C:\Users\NADA EL-SAYED\Downloads\FER-project\trial11.m
a = imresize(a,0.4);
detector = vision.CascadeObjectDetector('Mouth');
detector.MergeThreshold=50;
bbox = step(detector, a);
B = insertObjectAnnotation(a,'rectangle',bbox,'Mouth');
imshow(B);
a = imread('1.jpg');
a = imresize(a,0.4);
detector = vision.CascadeObjectDetector('Nose');
detector.MergeThreshold=30;
bbox = step(detector, a);
B = insertObjectAnnotation(a,'rectangle',bbox,'Nose');
imshow(B);
Figure 1
Nose
Workspace
a 98x128 uint8
B 98x128x3 uint8
bbox [61,51,21,17]
detector 1x1 CascadeObjectD...
Command Window

GUI:

We built a GUI for the program as GUIs have many advantages.

Advantages of graphical user interface:-

Easiness for non-technical people or for beginners good GUI's tends to make easiness in life. It allows them to get their work done efficiently and quickly without knowing the technical details of the program.

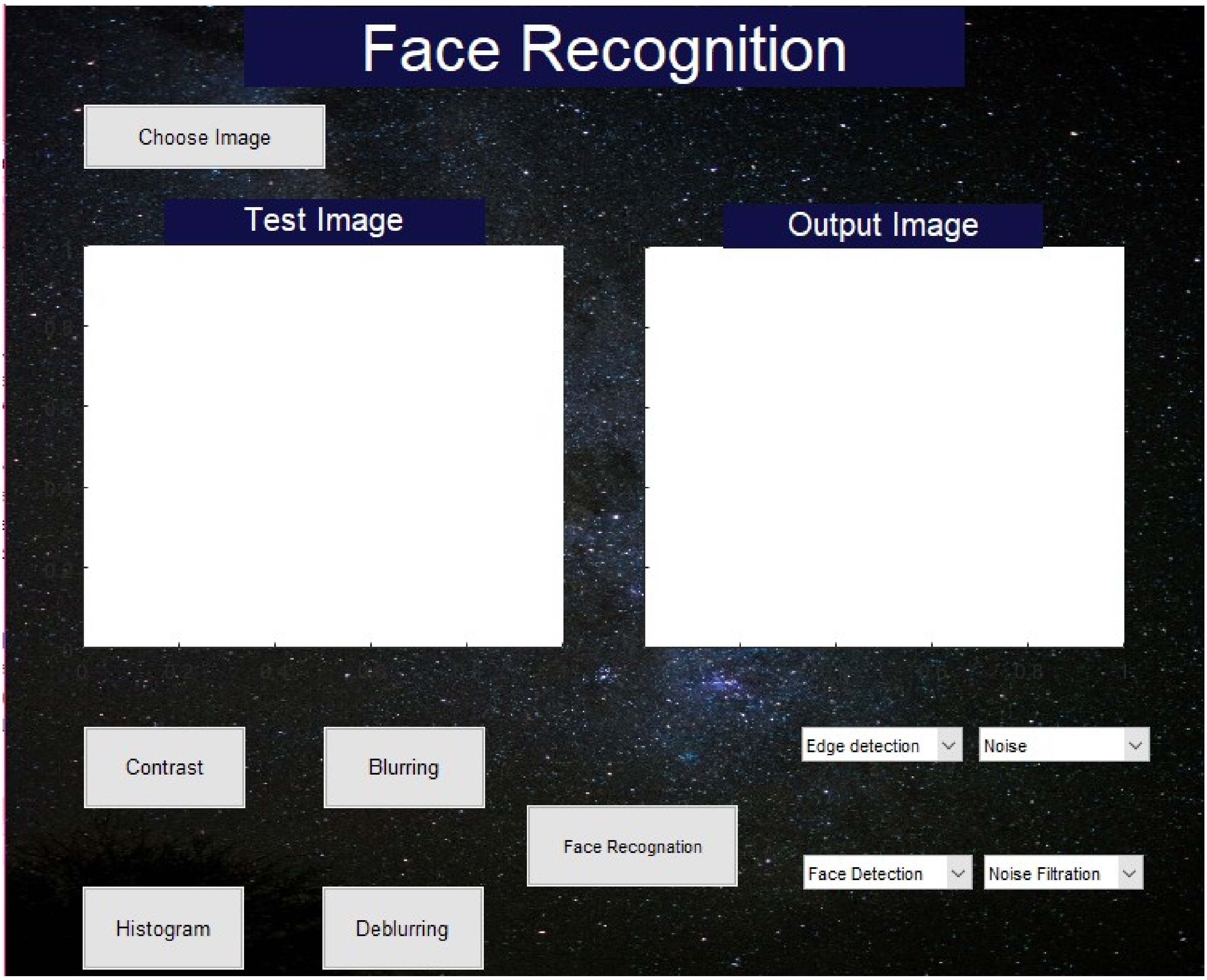


figure 16 our GUI

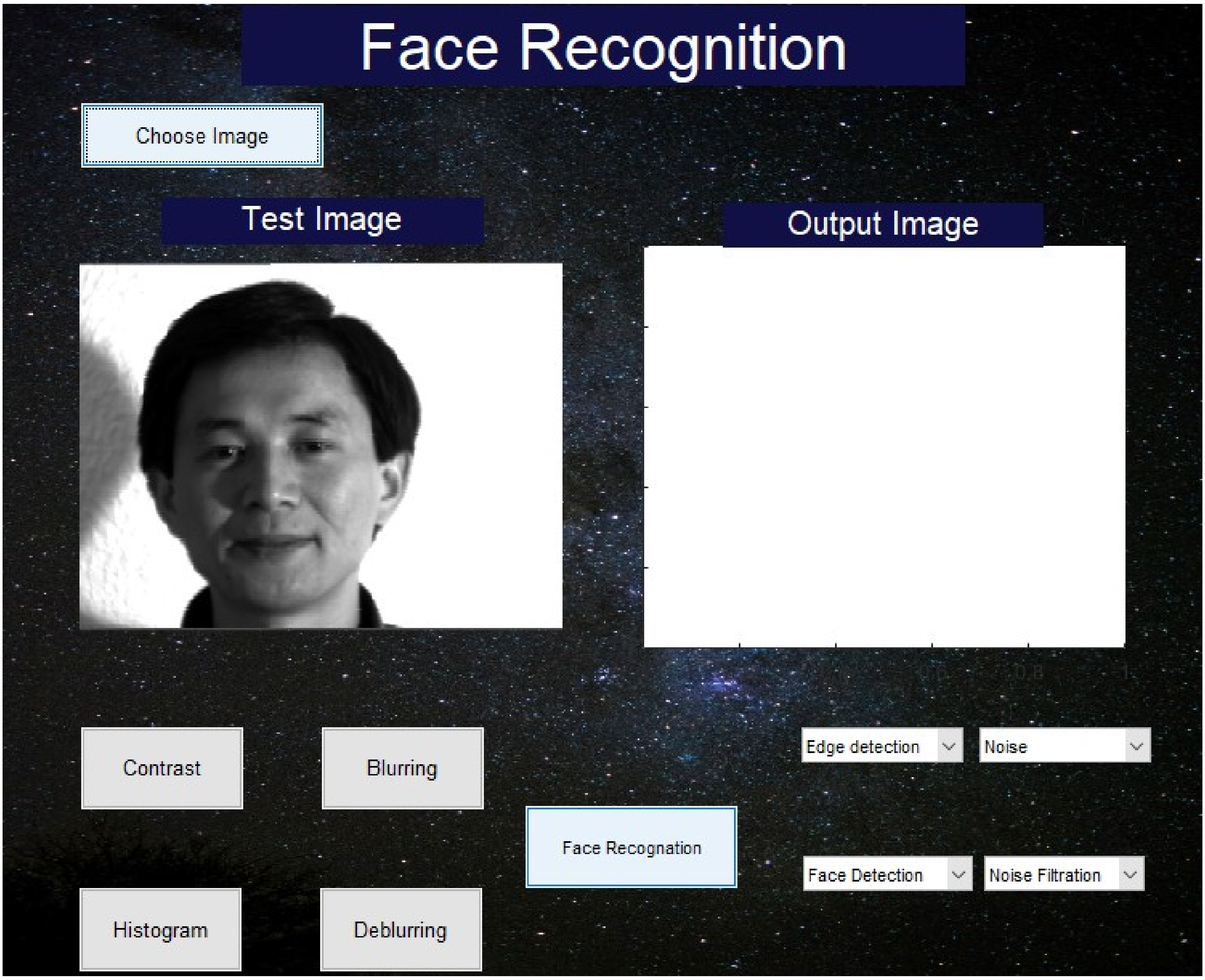


figure 17 'Choose Image' button

choose image: when the button is clicked it opens a window that allows the user to browse to the file containing the test images and select the input image.

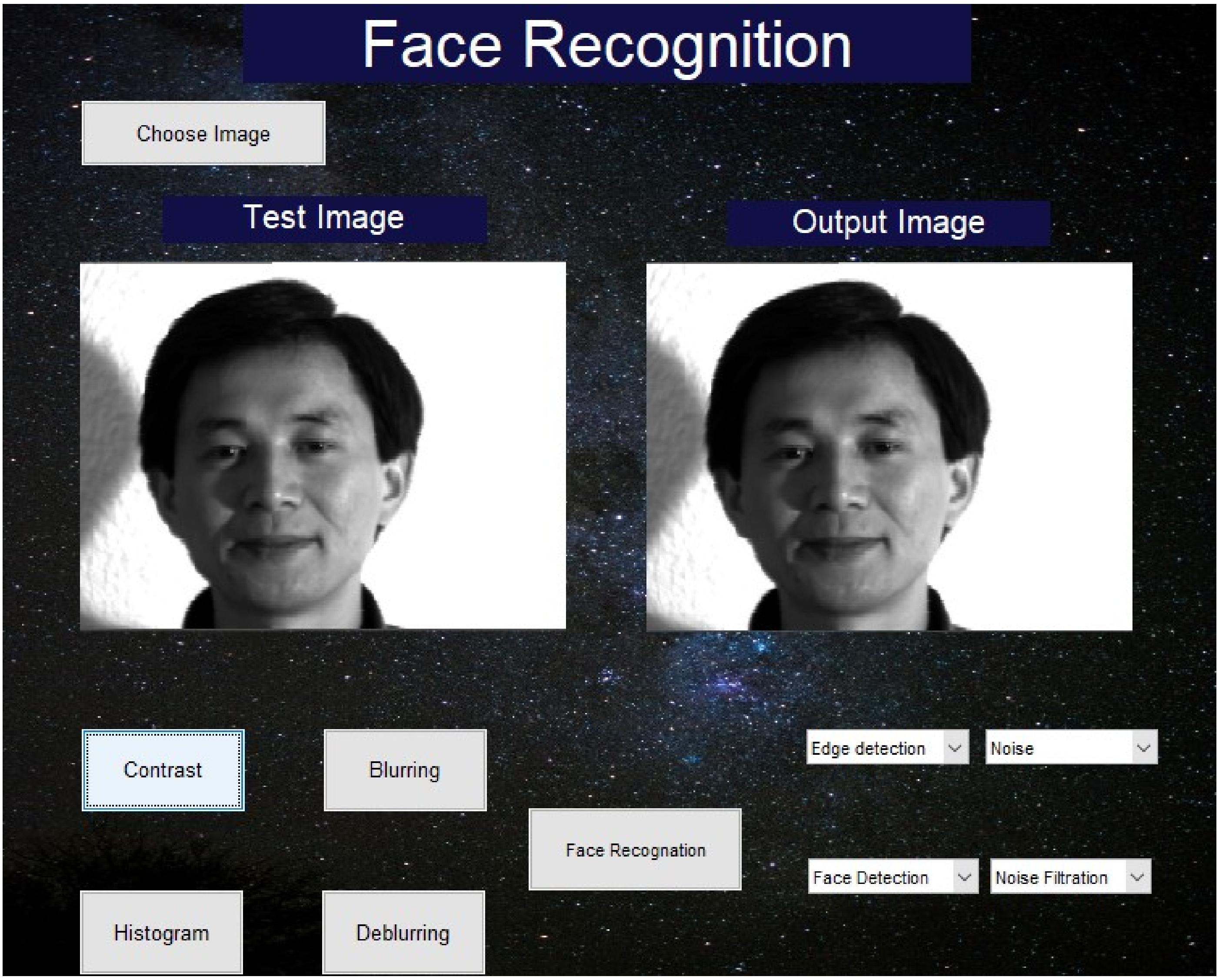


figure 18 'contrast' button

contrast button: enhances the contrast of the image by creating a grayscale colormap. The new colormap increases the contrast among pixels that are difficult to distinguish visually, but have slightly different brightness values.

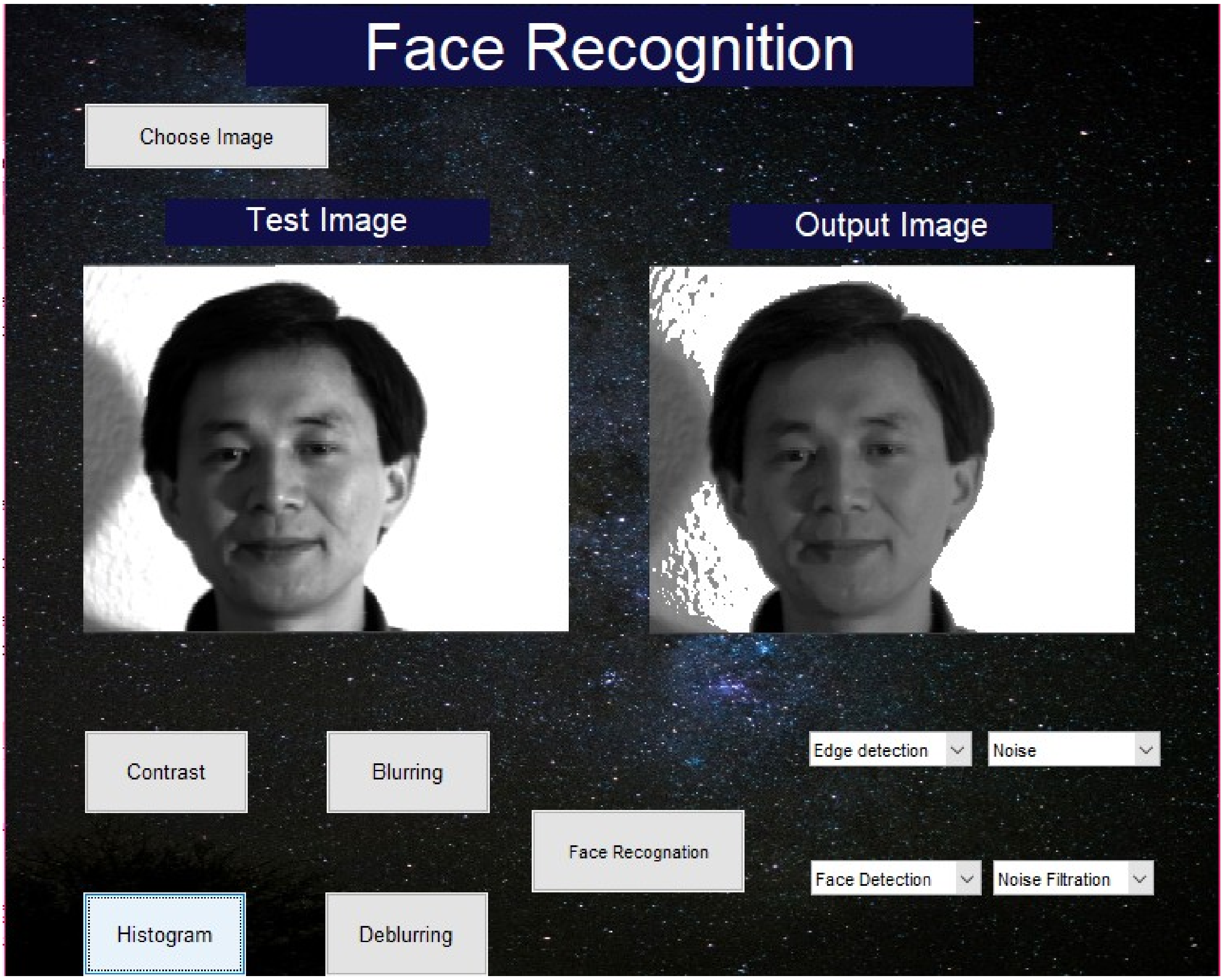


figure 19 'Histogram' button

Histogram Equalization button: this button uses the histeq( ) function to produce an output image with pixel values evenly distributed throughout the intensity range.

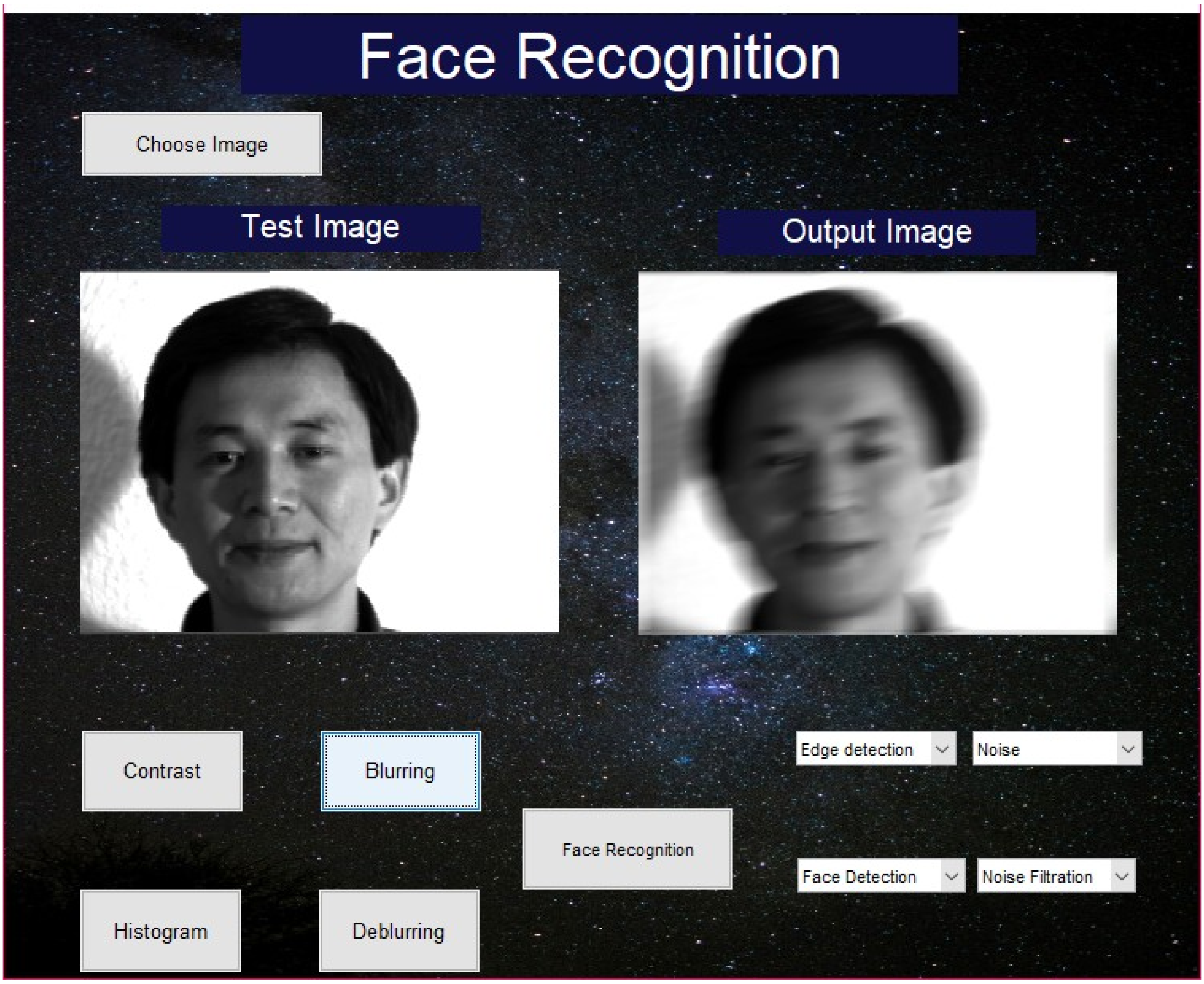


figure 20 'Blurring' button

Blurring button: blurs the image using the imfilter() function.

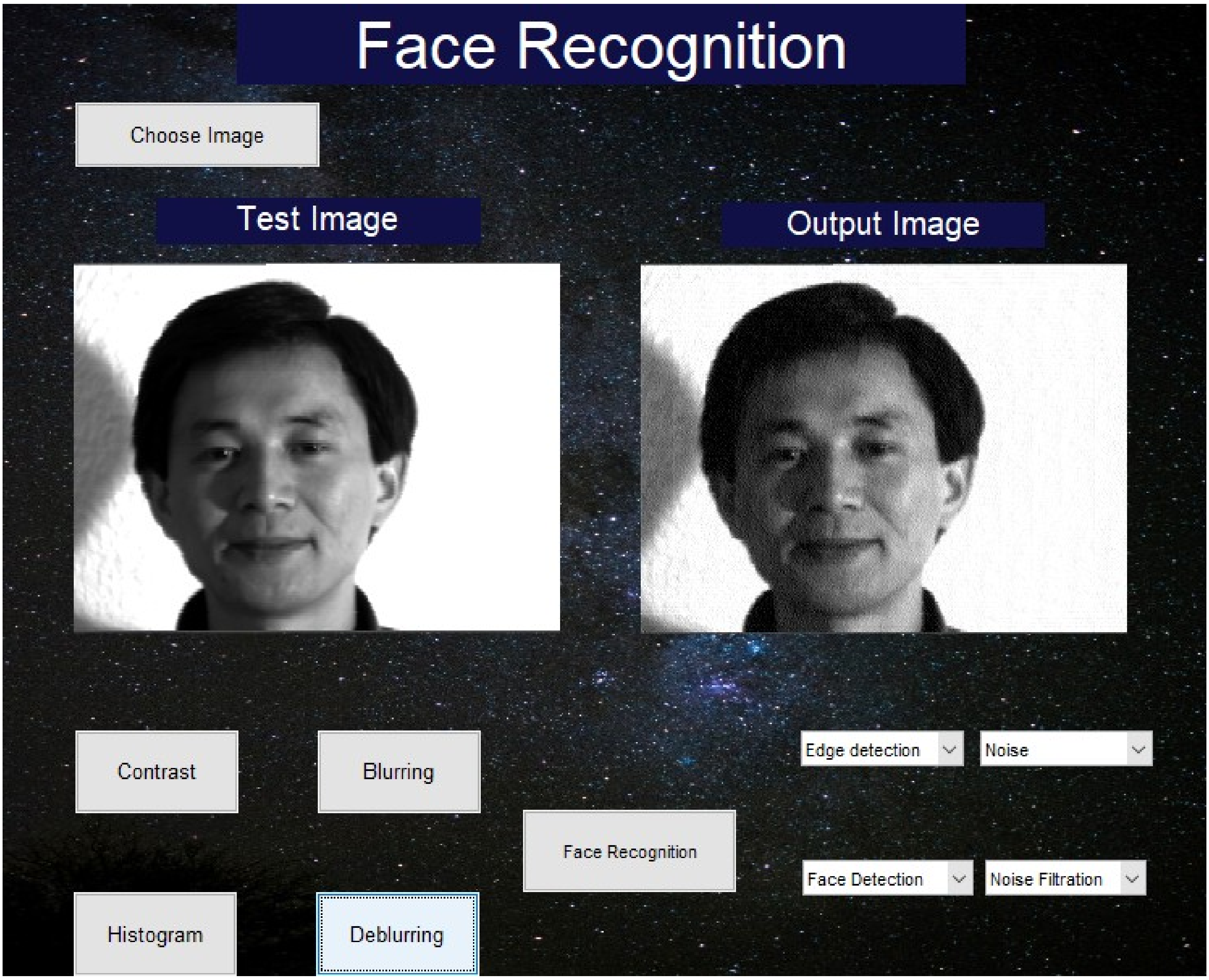


figure 21'Deblurring' button

Deblurring button: applies imfilter() to blur the image then the deconvwnr() function to deblur it.

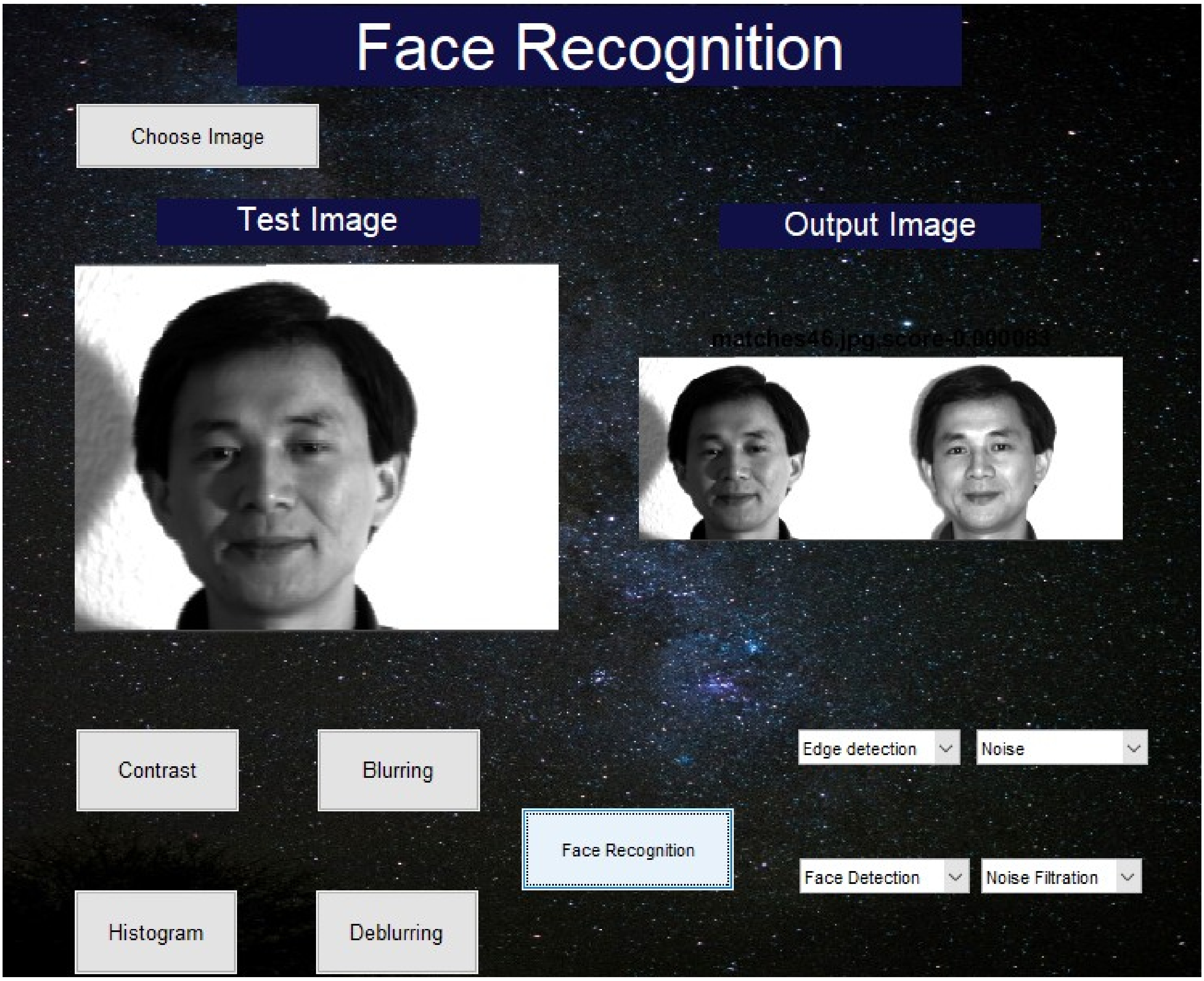
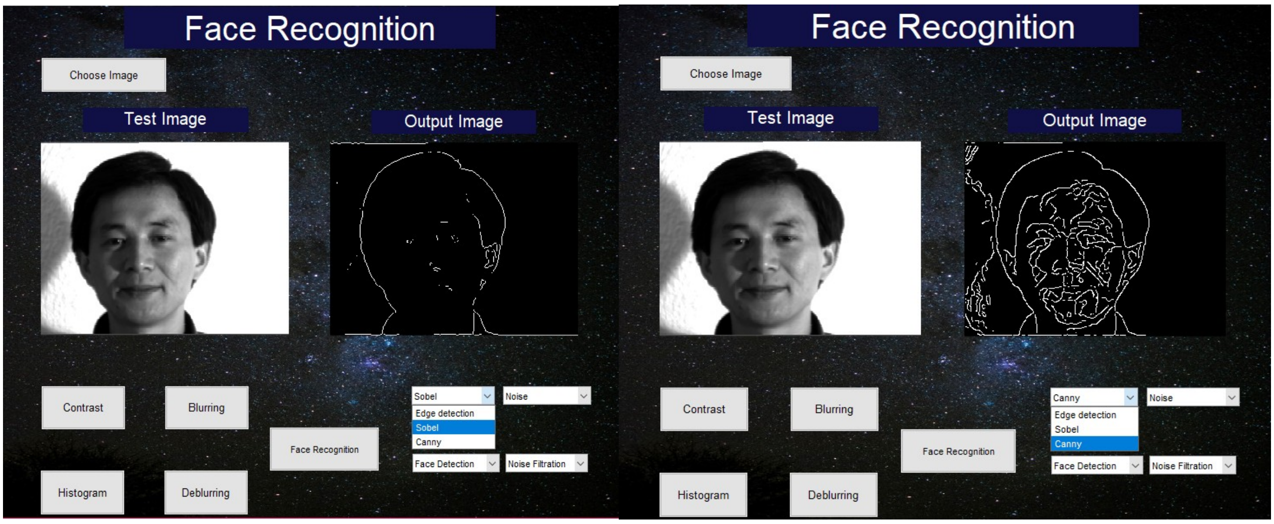


figure 22 'face recognition' button

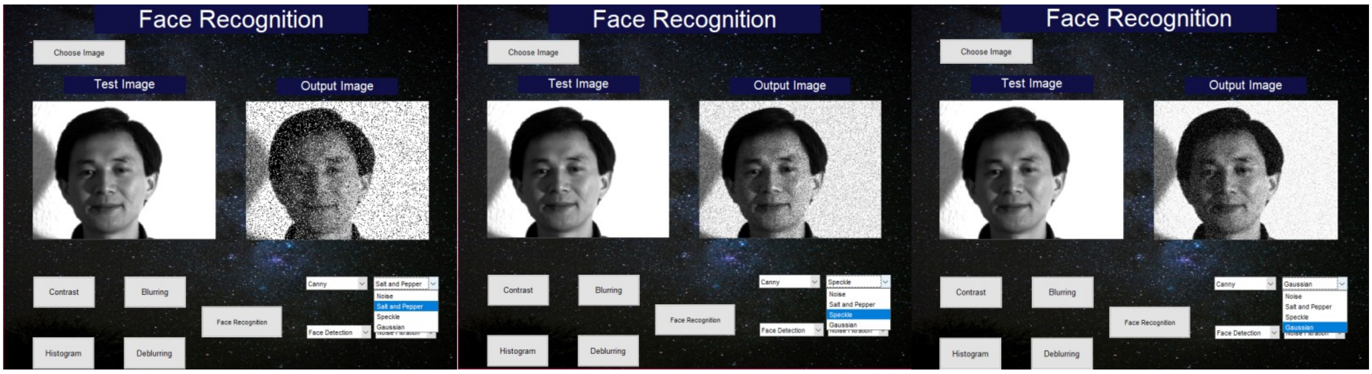
Face recognition button: activates the PCA model responsible for matching the input image with the suitable image from the training dataset.

edge detection list: contains two filters: 1- sobel filter. 2- canny filter.
The canny filter detects edges more accurately



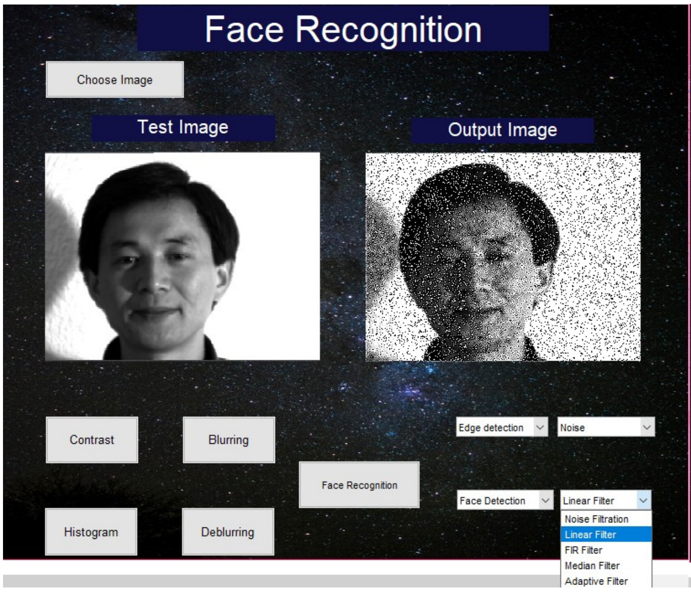
Noise list: a list that contains three types of noise applied by the imnoise() function:

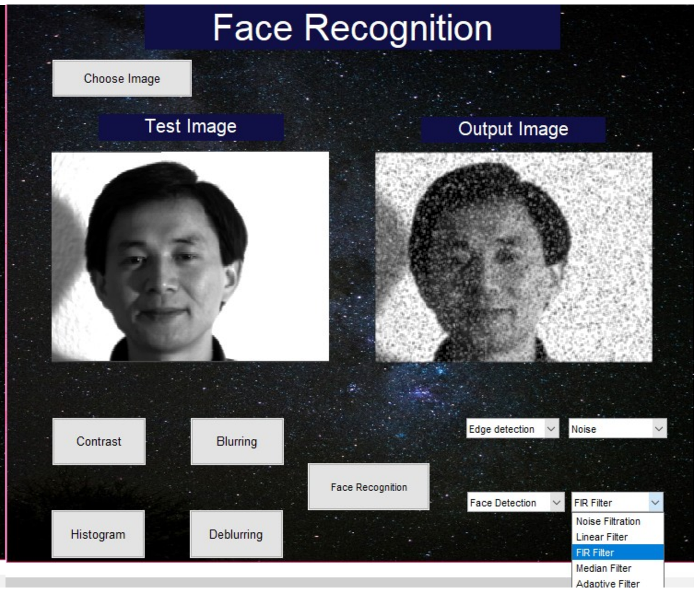
1. Gaussian noise.
2. Salt and pepper noise.
3. Speckle noise.

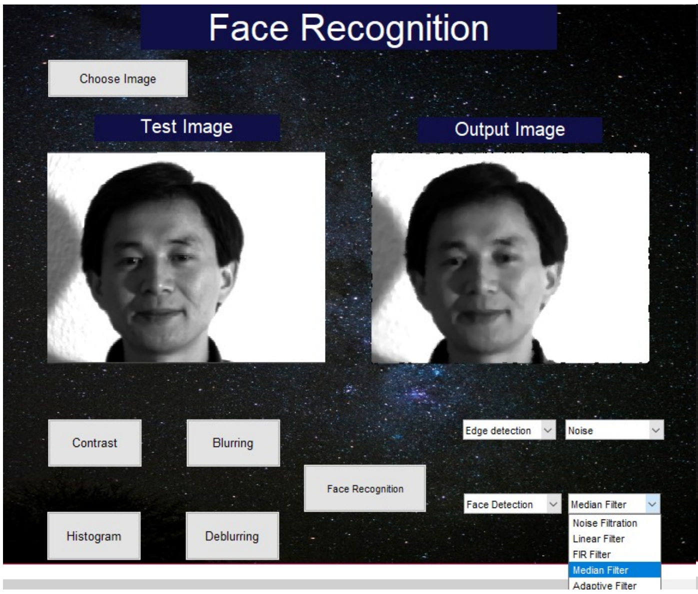


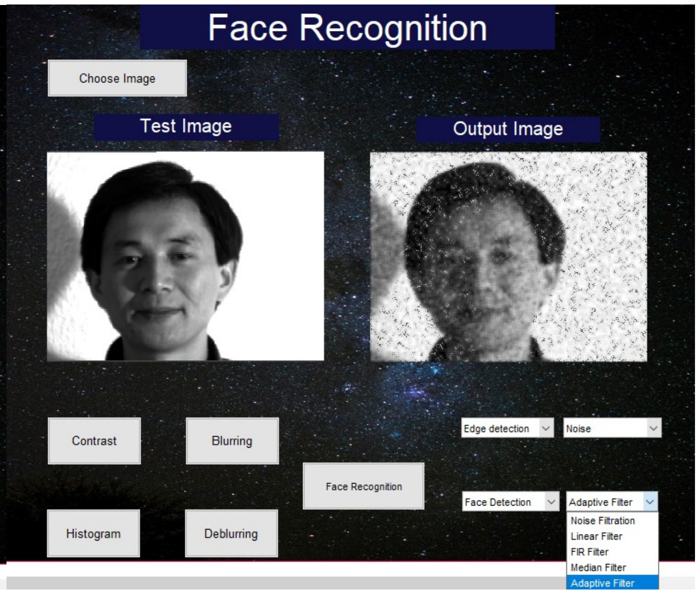
Noise Filtration list: a list that contains four types of noise filters:

1. Linear Filter.
2. FIR Filter.
3. Median Filter.
4. Adaptive Filter.





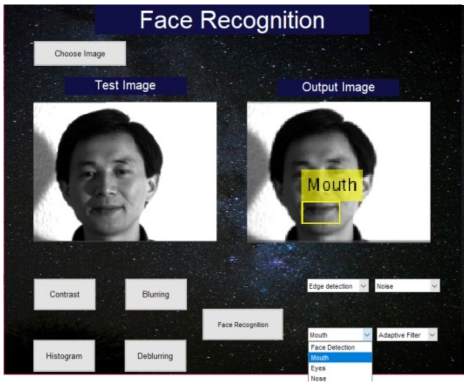


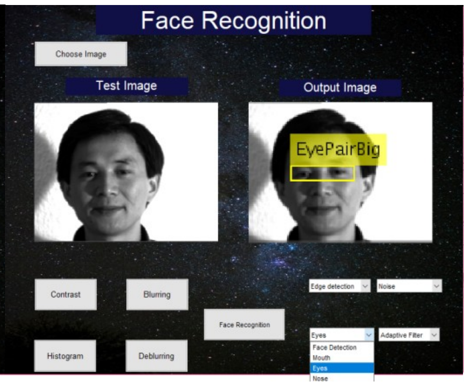


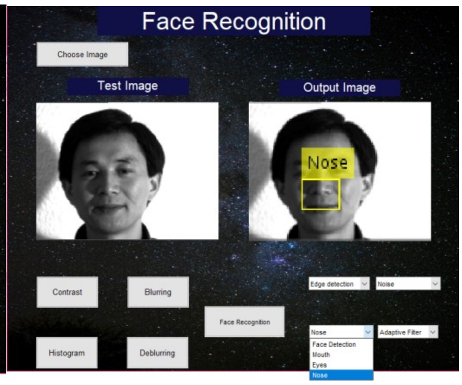
Face detection list: a list that contains three types of feature detection:

1. Mouth detection.
2. Nose Detection.
3. Eyes Detection.

We use the CascadeObjectDetector() function from the computer vision toolbox to determine the boundaries for each feature, then we step() function to apply the detection on the image, then we use the insertObjectAnnotation() to draw the boundary box around the detected feature.


Face Recognition
Choose Image
Test Image
Output Image
Mouth
Contrast
Blurring
Face Recognition
Histogram
Deblurring


Face Recognition
Choose Image
Test Image
Output Image
EyePairBig
Contrast
Blurring
Face Recognition
Histogram
Deblurring


Face Recognition
Choose Image
Test Image
Output Image
Nose
Contrast
Blurring
Face Recognition
Histogram
Deblurring

Reference:

- http://users.cecs.anu.edu.au/~wanglei/SPR_course/Direct_LDA.pdf
- http://www.rroij.com/open-access/an-overview-on-image-processing-techniques.php?aid=47175
- https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
- https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7
- https://www.youtube.com/watch?v=kE5QZ8G_78c&t=400s
- https://www.youtube.com/watch?v=xtOg44r6dsE&t=865s
- http://www.itrelease.com/2017/11/advantages-disadvantages-graphical-user-interface/