Digital Image Processing

Arik Pamnani Meet Gandhi Shivdutt Sharma Ravi Shrimal

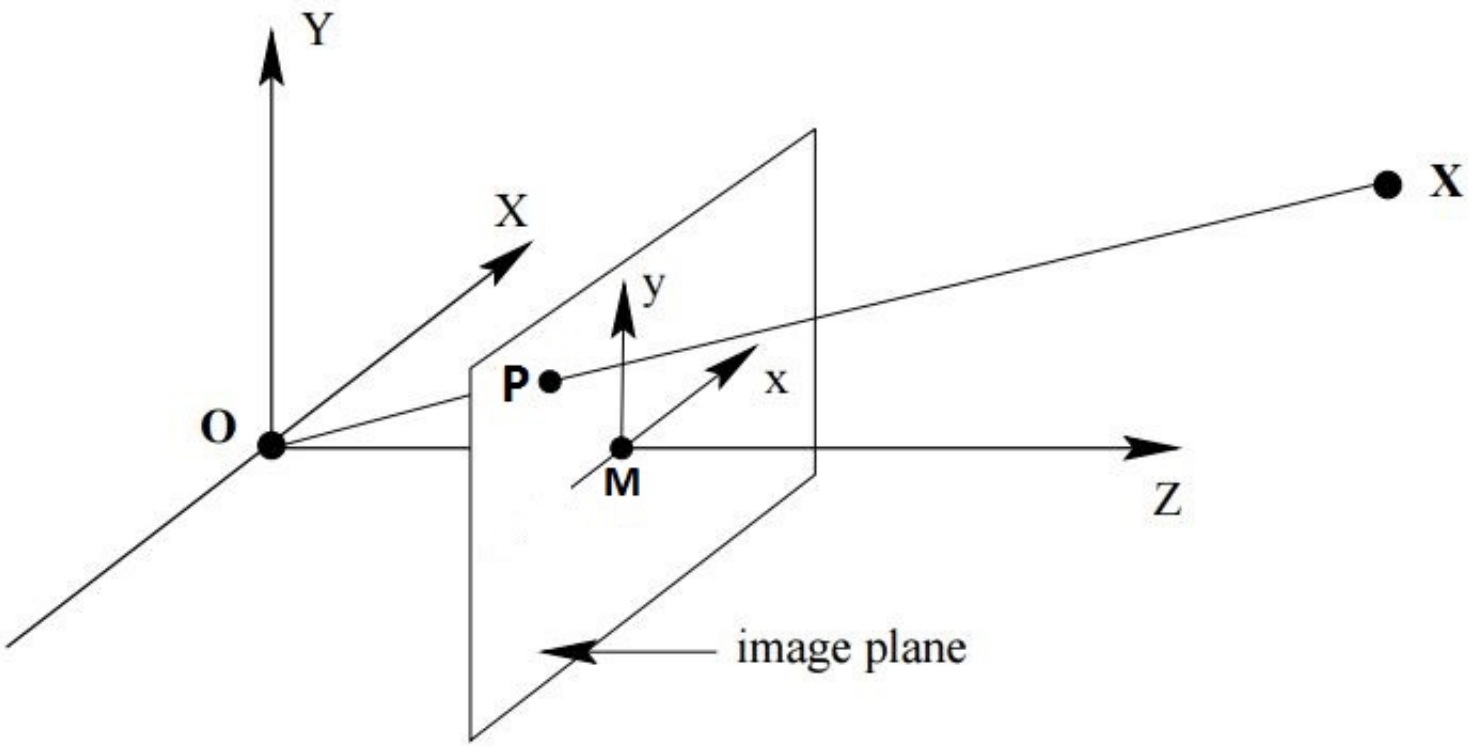
IIT Gandhinagar

17-4-2016

Camera Model

A camera matrix is a 3X4 matrix which maps 3D points in world to 2D points in an image.

Camera Model



Camera Model

From the previous image, the following can be concluded from the similarity of two triangles.

$$\frac{y_1}{x_1} = \frac{y_2}{x_2} = \frac{f}{x_3}$$

Here, (y_1, y_2) is a point in the image plane while (x_1, x_2, x_3) represents a point in 3D. f is the focal length of the camera.

Camera Model

A 3D point is mapped onto the image plane using the camera model given by the matrix equation -

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{f}{x_3} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Camera Model

We derive the camera matrix by rewriting the previous equation.

- In place of the 2D vector (y_1, y_2) we consider the 3D vector $(y_1, y_2, 1)$.

Camera Model

We derive the camera matrix by rewriting the previous equation.

- In place of the 2D vector (y_1, y_2) we consider the 3D vector $(y_1, y_2, 1)$.
- For the representation of the 3D point (x_1, x_2, x_3), we consider a 4-dimensional vector, $(x_1, x_2, x_3, 1)$.

Camera Model

Now, the equation becomes -

$$\begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = \frac{f}{x_3} \begin{bmatrix} x_1 \\ x_2 \\ \frac{x_3}{f} \end{bmatrix}$$

Camera Model

The 3D coordinates are expressed in a homogenous representation, the equation finally becomes -

$$\begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = \frac{f}{x_3} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

The camera matrix

The following 3X4 matrix is the camera matrix obtained from the previous equation -

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix}$$

Image representation

Mathematically, a 2-D image can be considered as a function of two variables, f(x, y), where x and y are coordinates in space. The value of the function at a pair of coordinates (x, y) gives the intensity of the image at that point.

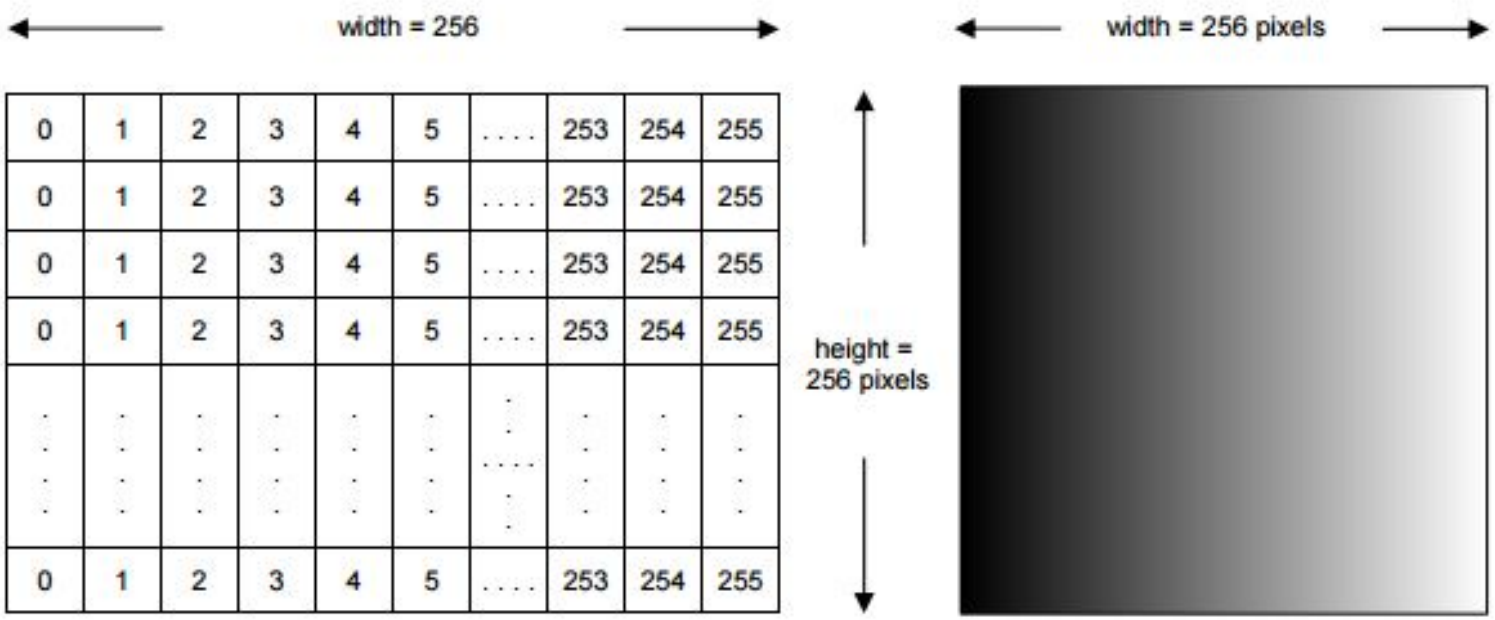
Image representation

Equivalently, an image can be represented in the form of a mxn matrix where each entry of the matrix represents a pixel of the image.

Image representation

The intensity at each point in the image is measured as the amount of RED, BLUE and GREEN colour present.

Image as a matrix



Blurring an image

Original

Blurred

Blurring an image

Blurring or smoothing of an image is analogous to the filtering of 1D signals with Low-pass filters or High-pass filters.

Convolution

We define what is called a convolution. Given a kernel and a part of an image (they must be of the same order), convolution is defined as multiplying entries which are located at the same position in the two matrices and adding.

Convolution

We take a 3X3 kernel and perform convolution on a 3X3 matrix.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} * \begin{bmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$$

$$= (1 \times 7) + (2 \times 8) + (3 \times 9) + (4 \times 4) + (5 \times 5) + (6 \times 6) + (7 \times 1) + (8 \times 2) + (9 \times 3)$$

Blurring an Image

The following types of filters maybe used for blurring an image -

- Averaging filter
- Weighted average filter
- Gaussian filter

Blurring an Image

The following types of filters maybe used for blurring an image -

- Averaging filter
- Weighted average filter
- Gaussian filter

Blurring an Image

The following types of filters maybe used for blurring an image -

- Averaging filter
- Weighted average filter
- Gaussian filter

Blurring an Image

We have used an averaging filter like the one below -

$$\frac{1}{49}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Blurring an Image

Convolving the previous kernel (averaging filter) with the image is similar to the following operation -

- Take a pixel on the image. Cut out a 7X7 matrix with the given pixel as the central entry.
- Take the average of all the pixels in the matrix.
- Replace the value of the central entry with the average value obtained.
- Do this operation for all the pixels in the image.

Blurring an Image

Convolving the previous kernel (averaging filter) with the image is similar to the following operation -

- Take a pixel on the image. Cut out a 7X7 matrix with the given pixel as the central entry.
- Take the average of all the pixels in the matrix.
- Replace the value of the central entry with the average value obtained.
- Do this operation for all the pixels in the image.

Blurring an Image

Convolving the previous kernel (averaging filter) with the image is similar to the following operation -

- Take a pixel on the image. Cut out a 7X7 matrix with the given pixel as the central entry.
- Take the average of all the pixels in the matrix.
- Replace the value of the central entry with the average value obtained.
- Do this operation for all the pixels in the image.

Blurring an Image

Convolving the previous kernel (averaging filter) with the image is similar to the following operation -

- Take a pixel on the image. Cut out a 7X7 matrix with the given pixel as the central entry.
- Take the average of all the pixels in the matrix.
- Replace the value of the central entry with the average value obtained.
- Do this operation for all the pixels in the image.

Rotation of an Image

Original

Rotated by 90°

Rotation of an Image

A 2D image is a function of two variables, f(x, y) where x and y are coordinates in space. Therefore, every pixel of the image can be picturized as lying on the euclidean plane.

The 2D rotation matrix

The following is the rotation matrix which rotates each point on the image in anti - clockwise direction through an angle θ about the origin.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Rotation of an Image

The openCV module provides a modified transformation matrix with adjustable centre of rotation. It also provides scaled rotation.

$$\begin{bmatrix} \alpha & \beta & (1-\alpha)\cdot \mathit{centre.x} - \beta \cdot \mathit{centre.y} \\ -\beta & \alpha & \beta \cdot \mathit{centre.x} + (1-\alpha)\cdot \mathit{centre.y} \end{bmatrix}$$

$$\alpha = \mathit{scale} \cdot \cos\theta$$
$$\beta = \mathit{scale} \cdot \sin\theta$$

Eigenfaces

Eigenfaces

•We will train M grayscale face images I_1, I_2, I_3… I_M, The more the no. of images, the better the system will be trained.

•Every image should have same size N*N.

•Every pixel has a value between 0 to 255.

•$I=[I_1 \ I_2 \ I_3 \ \dots \ I_M]$
So, I will be of size $N^2 \times M$

Eigenfaces

Image vector can be represented as:

$$I_k = \begin{bmatrix} p^k_{1,1} & p^k_{1,2} & \cdots & p^k_{1,N} \\ p^k_{2,1} & p^k_{2,2} & \cdots & p^k_{2,N} \\ \vdots & & & \\ p^k_{N,1} & p^k_{N,2} & \cdots & p^k_{N,N} \end{bmatrix}_{N \times N}$$

Each pixel value lies between 0 to 255.

Eigenfaces

To make calculations simple , face image I_k is transformed from a N×N matrix to a column vector .

$$\Gamma_k = \begin{bmatrix} p^k_{1,1} \\ p^k_{1,2} \\ \vdots \\ p^k_{1,N} \\ p^k_{2,1} \\ p^k_{2,2} \\ \vdots \\ p^k_{2,N} \\ \vdots \\ p^k_{N,1} \\ p^k_{N,2} \\ \vdots \\ p^k_{N,N} \end{bmatrix}_{N\times 1}$$

Eigenfaces

The common feature of all the faces can be found in a average face.
We will simply find the average of all the faces , i.e.

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

To find the unique features of these faces, we will subtract all the faces from average face.

$$\Phi_i = \Gamma_i - \Psi$$

Covariance Matrix

Covariance provides a measure of strength of the correlation between two or more sets of random variants. The covariance of two random variants X and Y, each with sample size N , is defined by the expectation value.

$$\begin{aligned} \operatorname{cov}(X, Y) &= \langle (X - \mu_X)(Y - \mu_Y) \rangle \\ &= \langle X\ Y \rangle - \mu_X\ \mu_y \end{aligned}$$

Covariance Matrix

$$\Sigma = \mathrm{E}\left[(\mathbf{X} - \mathrm{E}[\mathbf{X}])\ (\mathbf{X} - \mathrm{E}[\mathbf{X}])^{\mathrm{T}} \right]$$

Eigenfaces

Now, we have to find the set of orthonormal vectors which can describe our data.

Covariance Matrix

$$C = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \Phi_i^T = AA^T$$

$$A = [\Phi_1 \Phi_2 \dots \Phi_M]_{N^2 \times M}$$

Eigenfaces

Next step is to find the eigenvectors and eigen values of C. but it would be very difficult task, since the size of A is $N^2 \times M$, the size of C is $N^2 \times N^2$. For example if the size of image is 100 X 100, then the size of C will be 10000 X 10000.

To Tackle this problem, we define the transpose of matrix C by $L = A^T A$ (since $C = AA^T$) which will be of size MXM. (much smaller than NXN)

Eigen value Eqn.

$$Lv_i = \lambda_i v_i,$$

Where Vi is an eigenvector and λi is corresponding eigen value.

Eigenfaces

$$
\begin{aligned}
Lv_i &= \lambda_i v_i, \\
ALv_i &= \lambda_i Av_i \Rightarrow \\
AA^T Av_i &= \lambda_i Av_i \Rightarrow \\
CAv_i &= \lambda_i Av_i,
\end{aligned}
$$

- Let $u_i = Av_i$. Then u_i is an eigenvector of C and λ_i is the corresponding eigen value.
- The eigenvectors with the smallest eigen values are discarded, we will only take some R eigenvectors out of M. These eigenvectors are called **eigenfaces**.

$$U = [u_1 u_2 \dots u_R]_{N^2 \times R}$$

Various Methods and Algorithms used for Face Detection

Problem Definition & History

Year	Authors	Method
1973	Kanade	First automated system
1987	Sirovich & Kirby	Principal Component Analysis
1991	Turk & Pentland	Eigenface
1996	Etemad & Chellapa	Fisherface
2001	Viola & Jones	AdaBoost + Haar Cascade
2007	Naruniec & Skarbek	Gabor Jets

Major milestone that reinvigorated research.

Codes and Concepts

- How computers detect faces or separate other things from human faces?
- By Skin Colour (Colour Detection).
- Motion(Blinking of Eyes).
- Head shape and other unique features of the face.
- All of the above combined.

Codes and Concepts

- How computers detect faces or separate other things from human faces?
- By Skin Colour (Colour Detection).
- Motion(Blinking of Eyes).
- Head shape and other unique features of the face.
- All of the above combined.

Codes and Concepts

- How computers detect faces or separate other things from human faces?
- By Skin Colour (Colour Detection).
- Motion(Blinking of Eyes).
- Head shape and other unique features of the face.
- All of the above combined.

Codes and Concepts

- How computers detect faces or separate other things from human faces?
- By Skin Colour (Colour Detection).
- Motion(Blinking of Eyes).
- Head shape and other unique features of the face.
- All of the above combined.

Codes and Concepts

- How computers detect faces or separate other things from human faces?
- By Skin Colour (Colour Detection).
- Motion(Blinking of Eyes).
- Head shape and other unique features of the face.
- All of the above combined.

Codes and Concepts

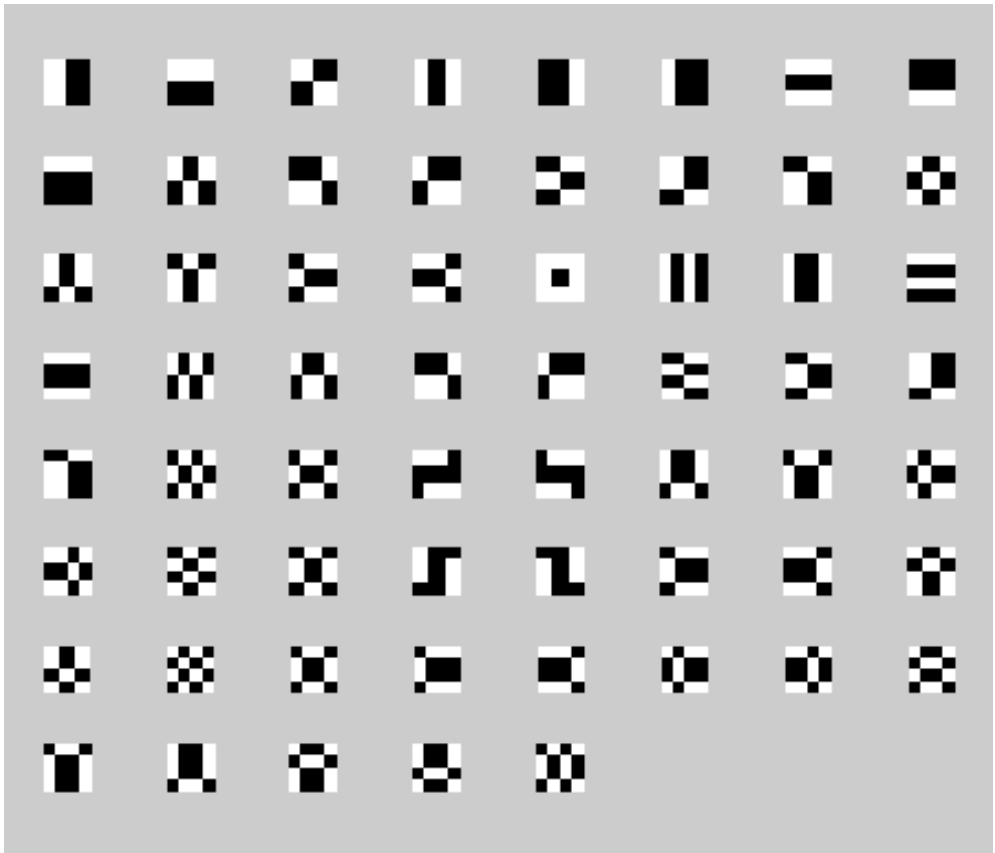
- Most of the modern algorithms for face detection are appearance based rather then learning based.
- Modern face detection algorithms are mostly based on Viola Jones object detection frame work which is based on **Haar Cascades.**

Codes and Concepts

- Most of the modern algorithms for face detection are appearance based rather then learning based.
- Modern face detection algorithms are mostly based on Viola Jones object detection frame work which is based on **Haar Cascades.**

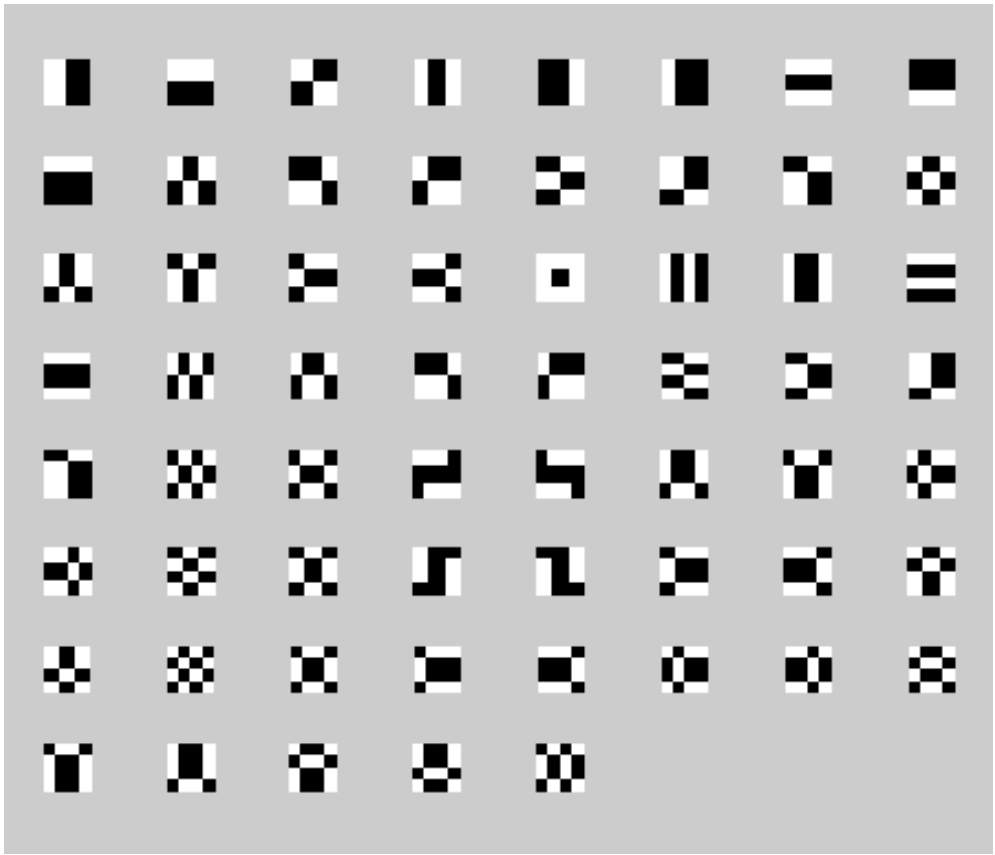
What is a **Haar Cascade ?**

- Haar Cascade is a collection of Haar like features included altogether to form a classifier.
- Haar like features:



What is a **Haar Cascade ?**

- Haar Cascade is a collection of Haar like features included altogether to form a classifier.
- Haar like features:



How does Haar like feature function?

- Fix a scale for Haar like feature (For example: 24 X 24 pixels).
- Starting from the topmost left, slide the Haar-like feature across the whole image.
- Calculate the average pixel values in white and black area of the Haar-cascade.
- If the difference of these values is greater than some threshold, the Haar-like feature matches with the portion of the image Haar-like feature is acted upon.

How does Haar like feature function?

- Fix a scale for Haar like feature (For example: 24 X 24 pixels).
- Starting from the topmost left, slide the Haar-like feature across the whole image.
- Calculate the average pixel values in white and black area of the Haar-cascade.
- If the difference of these values is greater than some threshold, the Haar-like feature matches with the portion of the image Haar-like feature is acted upon.

How does Haar like feature function?

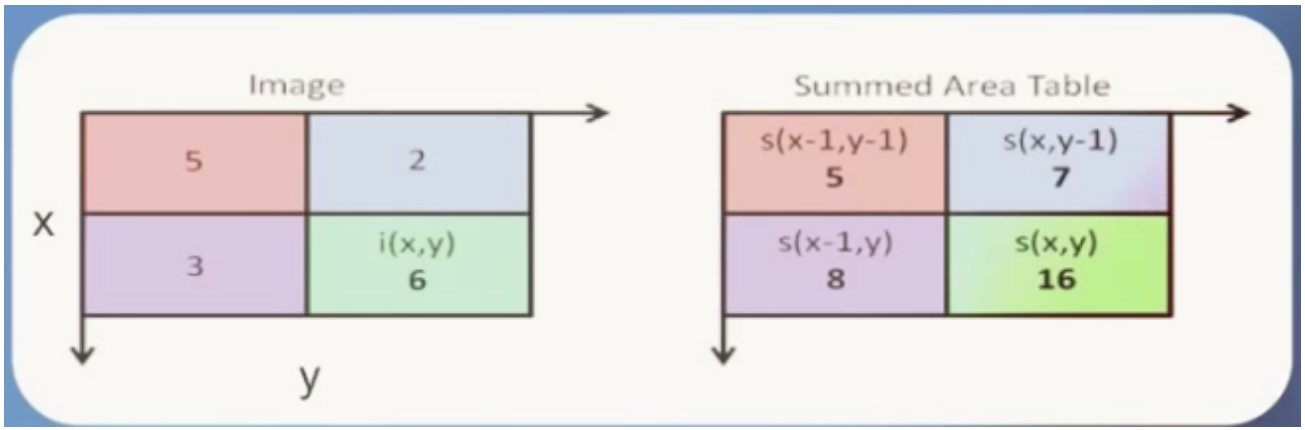
- Fix a scale for Haar like feature (For example: 24 X 24 pixels).
- Starting from the topmost left, slide the Haar-like feature across the whole image.
- Calculate the average pixel values in white and black area of the Haar-cascade.
- If the difference of these values is greater than some threshold, the Haar-like feature matches with the portion of the image Haar-like feature is acted upon.

How does Haar like feature function?

- Fix a scale for Haar like feature (For example: 24 X 24 pixels).
- Starting from the topmost left, slide the Haar-like feature across the whole image.
- Calculate the average pixel values in white and black area of the Haar-cascade.
- If the difference of these values is greater than some threshold, the Haar-like feature matches with the portion of the image Haar-like feature is acted upon.

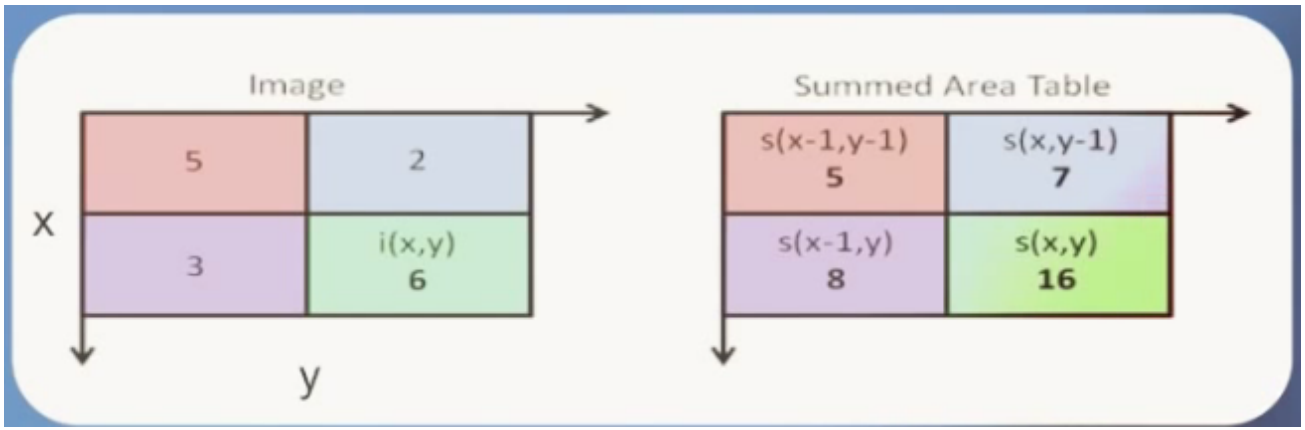
Integral Image- A technique to compute the sum of pixels in a given area

- Let the computed pixel values obtained by acting the convoluted kernel on an area of an input image be:
- Summed-Area Table can be computed by adding all the pixel values which are to the left and also up of the given pixel.



Integral Image- A technique to compute the sum of pixels in a given area

- Let the computed pixel values obtained by acting the convoluted kernel on an area of an input image be:
- Summed-Area Table can be computed by adding all the pixel values which are to the left and also up of the given pixel.



Integral Image- A technique to compute the sum of pixels in a given area

Sum of all pixels in

D = 1+4-(2+3)

= A+(A+B+C+D)-(A+C+A+B)

= D

- As a result, Integral Image computation allows to calculate the sum of pixels in a given rectangle by only knowing the pixel values at the corners of the rectangle.

Role of Adaboost

- Adaboost is a machine learning algorithm that eliminates the redundant features and finds only the best features which can describe the face among 160,000+ features.
- After these features are found, a linear combination of these features is used to decide whether a window or a photograph has a face or not.These features are known as **Weak classifiers.**
- Adaboost forms a **Strong classifier** which is a linear combination of weak classifiers.
- Negative value of weighted constant means that the image possesses the opposite feature to that of the Haar-like feature, to some extent which is decided by its magnitude.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier Weak classifier

Role of Adaboost

- Adaboost is a machine learning algorithm that eliminates the redundant features and finds only the best features which can describe the face among 160,000+ features.
- After these features are found, a linear combination of these features is used to decide whether a window or a photograph has a face or not.These features are known as **Weak classifiers.**
- Adaboost forms a **Strong classifier** which is a linear combination of weak classifiers.
- Negative value of weighted constant means that the image possesses the opposite feature to that of the Haar-like feature, to some extent which is decided by its magnitude.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier Weak classifier

Role of Adaboost

- Adaboost is a machine learning algorithm that eliminates the redundant features and finds only the best features which can describe the face among 160,000+ features.
- After these features are found, a linear combination of these features is used to decide whether a window or a photograph has a face or not.These features are known as **Weak classifiers.**
- Adaboost forms a **Strong classifier** which is a linear combination of weak classifiers.
- Negative value of weighted constant means that the image possesses the opposite feature to that of the Haar-like feature, to some extent which is decided by its magnitude.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier Weak classifier

Role of Adaboost

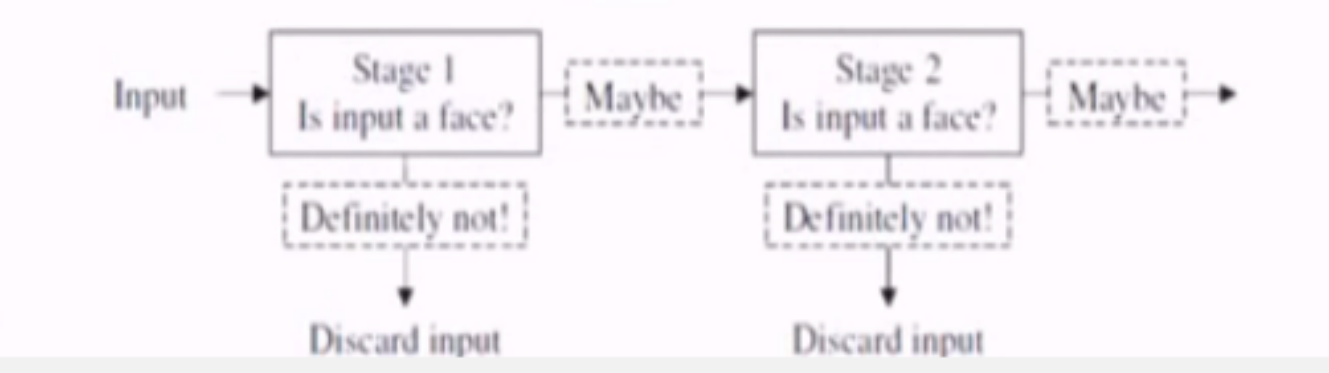
- Adaboost is a machine learning algorithm that eliminates the redundant features and finds only the best features which can describe the face among 160,000+ features.
- After these features are found, a linear combination of these features is used to decide whether a window or a photograph has a face or not.These features are known as **Weak classifiers.**
- Adaboost forms a **Strong classifier** which is a linear combination of weak classifiers.
- Negative value of weighted constant means that the image possesses the opposite feature to that of the Haar-like feature, to some extent which is decided by its magnitude.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier Weak classifier

Cascading

- Cascade classifier is made up of stages. Each stage consists of strong classifiers which is the collection of the most covariant features.



Applications of Face Detection and Recognition

- Attendance System
- Traffic Regulation at 4 Way crossing by counting the number of people through Face Detection and depending on it controlling the green signal
- Modern Digital and Smartphone cameras use Face Detection techniques for autofocus.
- Face Detection and Recognition is used in Biometrics, video surveillance and image database management.

Applications of Face Detection and Recognition

- Attendance System
- Traffic Regulation at 4 Way crossing by counting the number of people through Face Detection and depending on it controlling the green signal
- Modern Digital and Smartphone cameras use Face Detection techniques for autofocus.
- Face Detection and Recognition is used in Biometrics, video surveillance and image database management.

Applications of Face Detection and Recognition

- Attendance System
- Traffic Regulation at 4 Way crossing by counting the number of people through Face Detection and depending on it controlling the green signal
- Modern Digital and Smartphone cameras use Face Detection techniques for autofocus.
- Face Detection and Recognition is used in Biometrics, video surveillance and image database management.

Applications of Face Detection and Recognition

- Attendance System
- Traffic Regulation at 4 Way crossing by counting the number of people through Face Detection and depending on it controlling the green signal
- Modern Digital and Smartphone cameras use Face Detection techniques for autofocus.
- Face Detection and Recognition is used in Biometrics, video surveillance and image database management.

Resources

- https://en.wikipedia.org/wiki/Camera_matrix
- http://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html
- http://docs.opencv.org/3.1.0/da/d6e/tutorial_py_geometric_transformations.html#gsc.tab=0
- https://en.wikipedia.org/wiki/Eigenface
- http://blog.manfredas.com/eigenfaces-tutorial/