

# Multi Detection

**OpenCV**  
with  
**JavaFX**

▶ Aya Allah Ramzy

▶ Mohammed Salem

▶ Ahmed Alaa

▶ Mohammad Abd el-Haleem

▶ Islam Eissa

WORKING TEAM

# Face Detection

is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies.



# Applications Facial Recognition

- Biometrics
- Facial Recognition System
- Video Surveillance
- Human Computer Interface
- Image Database Management

# Applications Photography

- Digital Cameras



SO, HAVE TO TALK ABOUT OPENCV

Open Source Computer Vision Library :

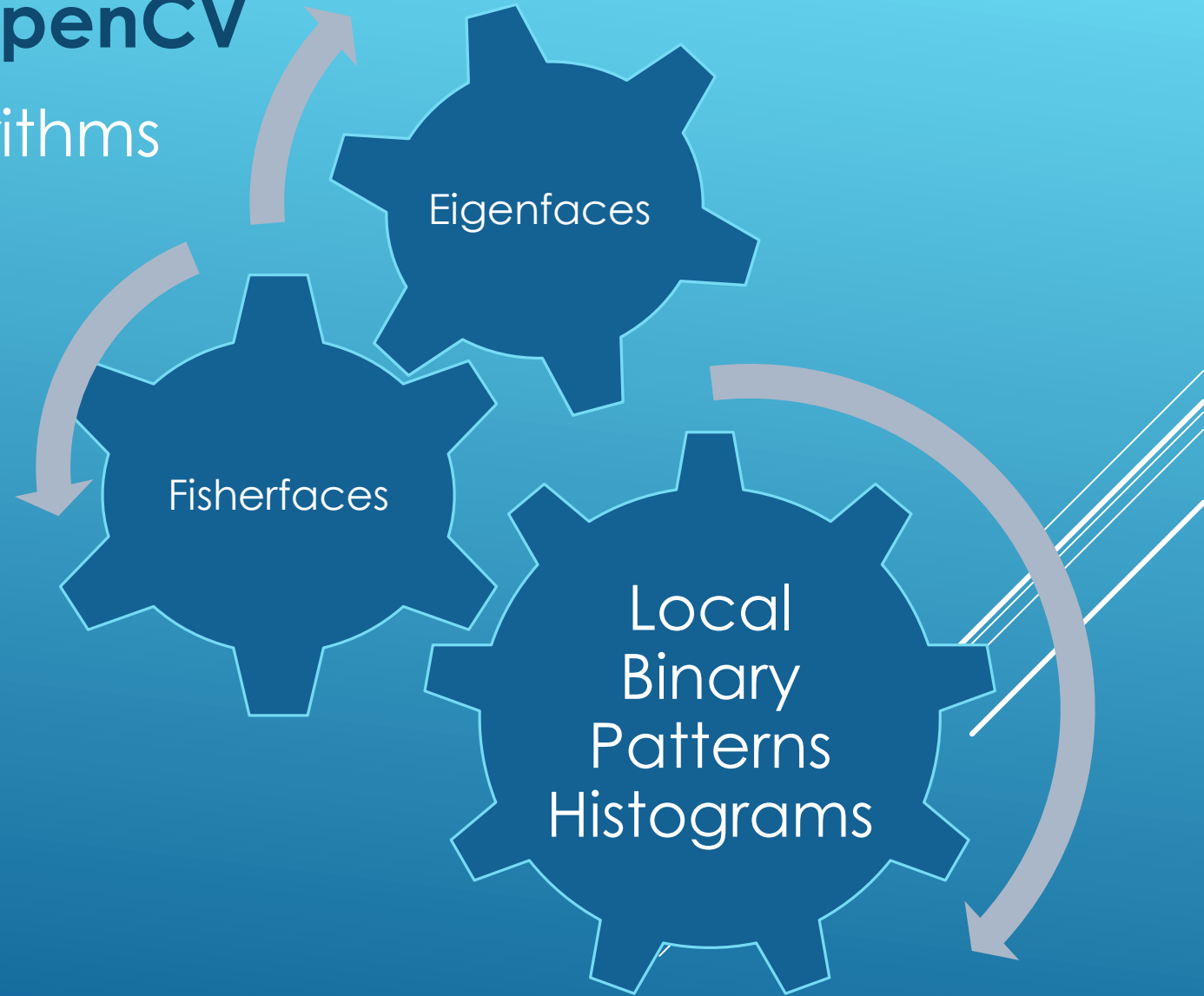
a library of programming functions mainly aimed at real-time computer vision, developed by Intel.

OpenCV is written in C++ and its primary interface is in C++, There are now full interfaces in Python, Java and MATLAB.

# Face Recognition with OpenCV

OpenCV uses three algorithms

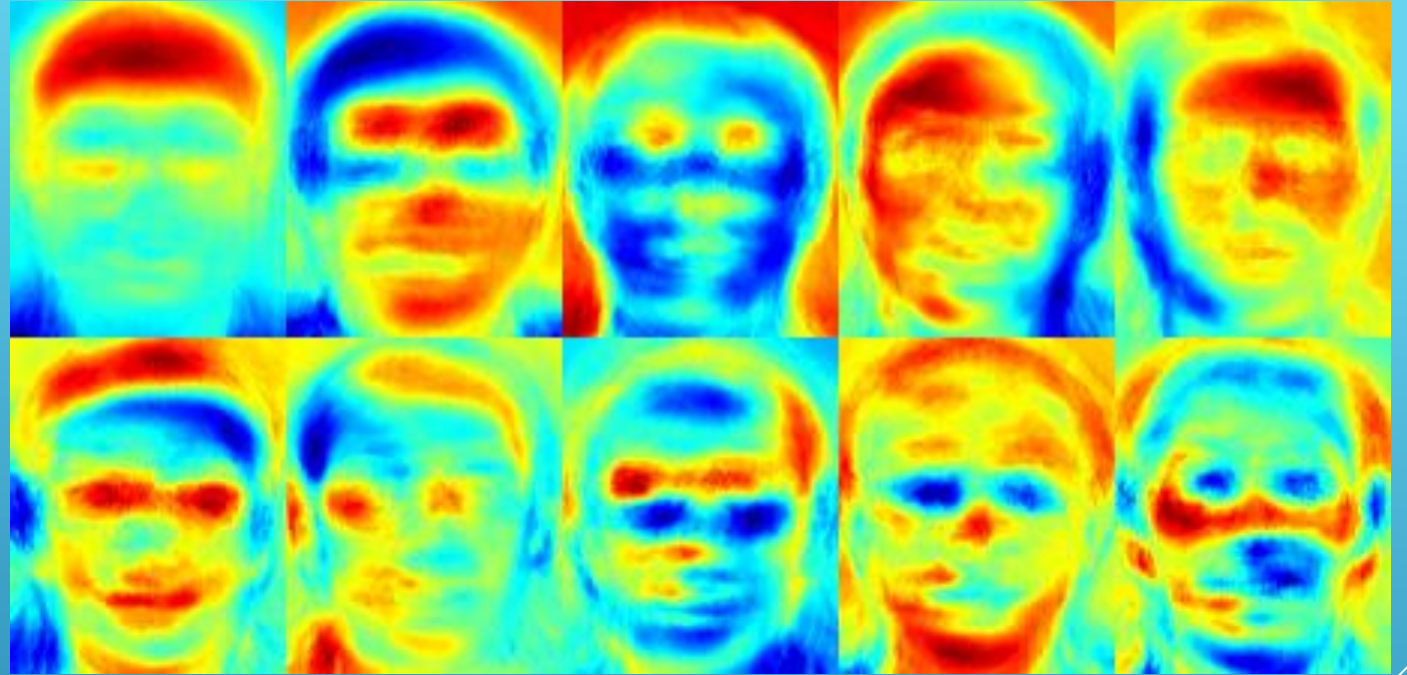
They are neural algorithms to train the program from a database of faces



# Eigenfaces

PCA (The Principal Component Analysis ), which is the core of the Eigenfaces method, finds a linear combination of features that maximizes the total variance in data.

The Principal Component Analysis (PCA) was independently proposed by **Karl Pearson (1901)** and **Harold Hotelling (1933)** to turn a set of possibly correlated variables into a smaller set of uncorrelated variables





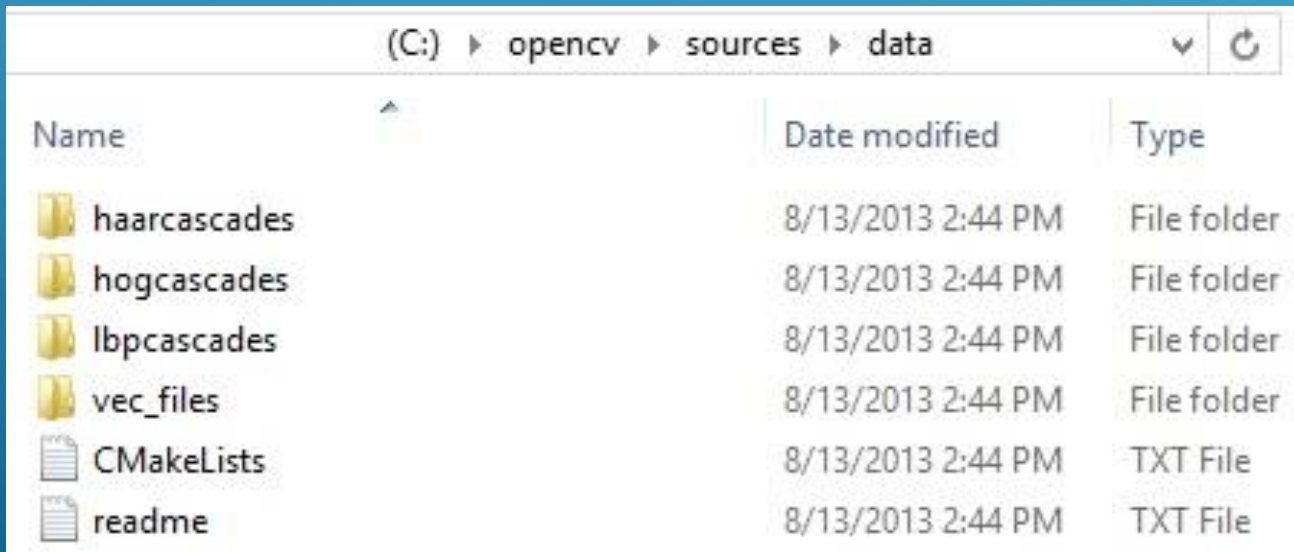
# Fisherfaces



The Fisherfaces method learns a class-specific transformation matrix, so they do not capture illumination as obviously as the Eigenfaces method.

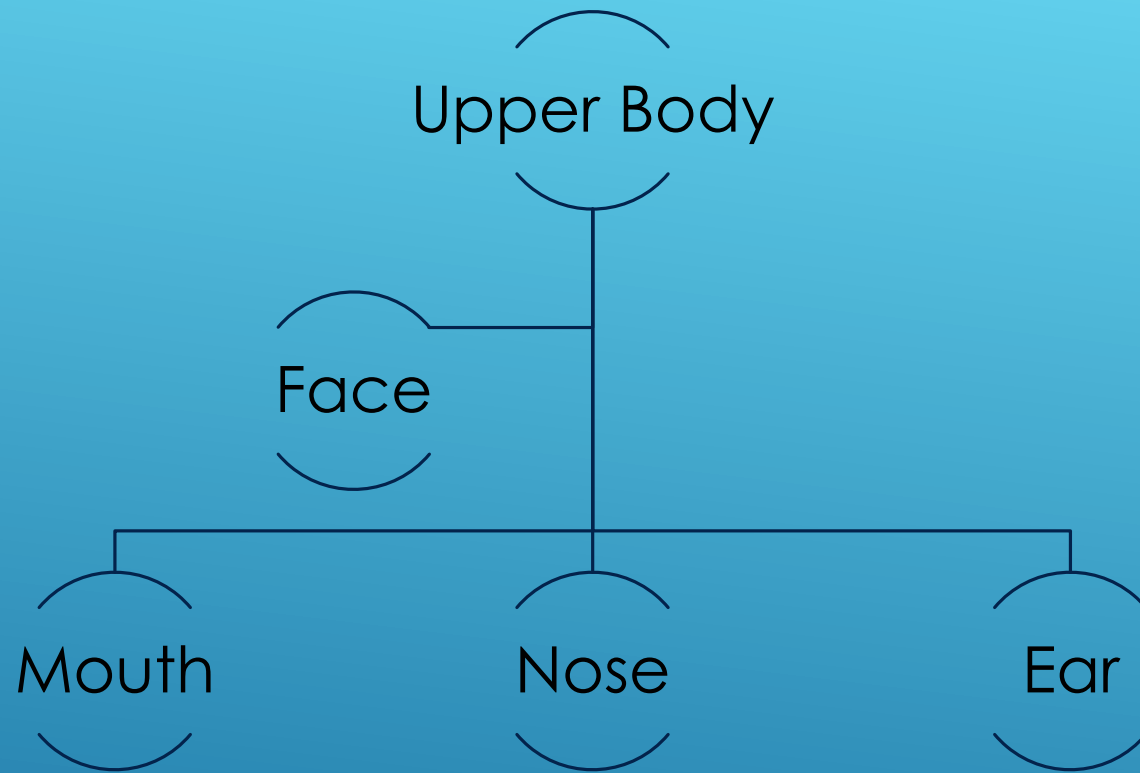
# OpenCV Database

OpenCV can be installed with folder contains various data that is used by cv libraries and/or demo applications.



The screenshot shows a Windows File Explorer window with the address bar set to (C:) > opencv > sources > data. The window displays a list of files and folders. The list has three columns: Name, Date modified, and Type. There are four folders (haarcascades, hogcascades, lbpcascades, vec\_files) and two text files (CMakeLists, readme). All items were last modified on 8/13/2013 at 2:44 PM.

Name	Date modified	Type
haarcascades	8/13/2013 2:44 PM	File folder
hogcascades	8/13/2013 2:44 PM	File folder
lbpcascades	8/13/2013 2:44 PM	File folder
vec_files	8/13/2013 2:44 PM	File folder
CMakeLists	8/13/2013 2:44 PM	TXT File
readme	8/13/2013 2:44 PM	TXT File



# OpenCV In Real Time Reduce Algorithms

# HOW IT WORKS

a) Load CascadeClassifier

```
CascadeClassifier classifier = new CascadeClassifier ("path/haarcascade_frontalface_alt.xml");
```

b) Start detect First from Upper Body

c) Draw a rectangle for each detected face

d) Decode AND Save Image in JPG Format

```

public class FaceDetecte extends CascadeLoder {

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.load("/usr/local/share/OpenCV/java/libopencv_java247.so");
        VideoCapture video = new VideoCapture(0); // open defalut cam
        Mat image = new Mat();
        for (int i = 0; i < 30; i++) { // read first grabed image from camera
            video.read(image);
            System.out.print(i%10);
            if(i%10 == 9) System.out.println();
        }

        // load the Classifier that will used in detect faces , given the path of input file
        CascadeClassifier classifier = new CascadeClassifier
            ("/usr/local/share/OpenCV/haarcascades/haarcascade_frontalface_alt.xml");

        // MatOfRect will use to store informations about detected faces
        MatOfRect rects = new MatOfRect();

        // start detect faces
        classifier.detectMultiScale(image, rects);
        // print number of detected faces
        System.out.println(String.format("\nDetected %s ", rects.toArray().length));

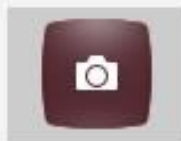
        // draw a rect foreach detected face
        for (Rect rect : rects.toArray()) {
            Core.rectangle(image, new Point(rect.x, rect.y), // source start point
                new Point(rect.x + rect.width, rect.y + rect.height), // end of diagonal point
                new Scalar(0,255,0)); // color of rect
        }

        // name of saved image
        String filename = "img/ouput.png";
        System.out.println(String.format("Writing %s", filename));
    }
}

```



# MuT Tracking



☒ Frontal Face Alt Tree

☐ Eye Glasses Tree

☐ Mouth

☐ Nose

☐ Left Ear

☐ Right Ear

☐ Upper Body



No.	Face: total		Face: right		Face: wrong		Eye: total		Eye: right		Eye: wrong		Mouth: total		Mouth: right		Mouth: wrong		Nose: total		Nose: right		Nose: wrong		Left Ear: total		Left Ear: right	
1	<div><div></div></div>	5	<div><div></div></div>	2	<div><div></div></div>	3	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	36	<div><div></div></div>	1	<div><div></div></div>	35	<div><div></div></div>	1	<div><div></div></div>	1		0	<div><div></div></div>	0	<div><div></div></div>	0
2	<div><div></div></div>	5	<div><div></div></div>	1	<div><div></div></div>	4	<div><div></div></div>	2	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	7	<div><div></div></div>	1	<div><div></div></div>	6	<div><div></div></div>	2	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	0
3	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	4	<div><div></div></div>	4		0	<div><div></div></div>	16	<div><div></div></div>	3	<div><div></div></div>	13	<div><div></div></div>	5	<div><div></div></div>	2	<div><div></div></div>	3	<div><div></div></div>	0	<div><div></div></div>	0
4	<div><div></div></div>	10	<div><div></div></div>	5	<div><div></div></div>	5		0		0		0	<div><div></div></div>	26	<div><div></div></div>	4	<div><div></div></div>	22	<div><div></div></div>	2	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	0	<div><div></div></div>	0
5	<div><div></div></div>	6	<div><div></div></div>	2	<div><div></div></div>	4	<div><div></div></div>	1		0	<div><div></div></div>	1	<div><div></div></div>	13	<div><div></div></div>	1	<div><div></div></div>	12	<div><div></div></div>	6	<div><div></div></div>	2	<div><div></div></div>	4	<div><div></div></div>	0	<div><div></div></div>	0
6	<div><div></div></div>	7	<div><div></div></div>	4	<div><div></div></div>	3	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	20	<div><div></div></div>	3	<div><div></div></div>	17	<div><div></div></div>	12	<div><div></div></div>	3	<div><div></div></div>	9	<div><div></div></div>	0	<div><div></div></div>	0
7	<div><div></div></div>	5	<div><div></div></div>	1	<div><div></div></div>	4		0		0		0	<div><div></div></div>	24	<div><div></div></div>	2	<div><div></div></div>	22	<div><div></div></div>	3	<div><div></div></div>	1	<div><div></div></div>	2	<div><div></div></div>	0	<div><div></div></div>	0
8	<div><div></div></div>	2	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	1		0	<div><div></div></div>	42	<div><div></div></div>	3	<div><div></div></div>	39	<div><div></div></div>	12	<div><div></div></div>	3	<div><div></div></div>	9	<div><div></div></div>	0	<div><div></div></div>	0
9	<div><div></div></div>	4	<div><div></div></div>	1	<div><div></div></div>	3		0		0		0	<div><div></div></div>	27	<div><div></div></div>	2	<div><div></div></div>	25	<div><div></div></div>	8	<div><div></div></div>	2	<div><div></div></div>	6	<div><div></div></div>	2	<div><div></div></div>	2
10	<div><div></div></div>	7	<div><div></div></div>	3	<div><div></div></div>	4		0		0		0	<div><div></div></div>	42	<div><div></div></div>	2	<div><div></div></div>	40	<div><div></div></div>	9	<div><div></div></div>	2	<div><div></div></div>	7	<div><div></div></div>	0	<div><div></div></div>	0
11	<div><div></div></div>	13	<div><div></div></div>	4	<div><div></div></div>	9		0		0		0	<div><div></div></div>	50	<div><div></div></div>	3	<div><div></div></div>	47	<div><div></div></div>	32	<div><div></div></div>	3	<div><div></div></div>	20	<div><div></div></div>	0	<div><div></div></div>	0
12	<div><div></div></div>	5	<div><div></div></div>	3	<div><div></div></div>	2	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	35	<div><div></div></div>	4	<div><div></div></div>	31	<div><div></div></div>	10	<div><div></div></div>	2	<div><div></div></div>	8	<div><div></div></div>	0	<div><div></div></div>	0
13	<div><div></div></div>	12	<div><div></div></div>	9	<div><div></div></div>	3		0		0		0	<div><div></div></div>	20	<div><div></div></div>	2	<div><div></div></div>	18	<div><div></div></div>	4	<div><div></div></div>	0	<div><div></div></div>	4	<div><div></div></div>	0	<div><div></div></div>	0
14	<div><div></div></div>	4	<div><div></div></div>	1	<div><div></div></div>	3		0		0		0	<div><div></div></div>	23	<div><div></div></div>	2	<div><div></div></div>	21	<div><div></div></div>	9	<div><div></div></div>	0	<div><div></div></div>	9	<div><div></div></div>	0	<div><div></div></div>	0
15	<div><div></div></div>	10	<div><div></div></div>	8	<div><div></div></div>	2	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	26	<div><div></div></div>	13	<div><div></div></div>	13	<div><div></div></div>	18	<div><div></div></div>	13	<div><div></div></div>	5	<div><div></div></div>	0	<div><div></div></div>	0
16	<div><div></div></div>	10	<div><div></div></div>	7	<div><div></div></div>	3	<div><div></div></div>	3	<div><div></div></div>	1	<div><div></div></div>	2	<div><div></div></div>	34	<div><div></div></div>	6	<div><div></div></div>	28	<div><div></div></div>	13	<div><div></div></div>	4	<div><div></div></div>	9	<div><div></div></div>	0	<div><div></div></div>	0
17	<div><div></div></div>	4	<div><div></div></div>	2	<div><div></div></div>	2	<div><div></div></div>	1	<div><div></div></div>	1		0	<div><div></div></div>	24	<div><div></div></div>	1	<div><div></div></div>	23	<div><div></div></div>	8	<div><div></div></div>	1	<div><div></div></div>	7	<div><div></div></div>	0	<div><div></div></div>	0
18	<div><div></div></div>	4	<div><div></div></div>	1	<div><div></div></div>	3	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	6	<div><div></div></div>	1	<div><div></div></div>	5	<div><div></div></div>	1	<div><div></div></div>	0	<div><div></div></div>	1	<div><div></div></div>	0	<div><div></div></div>	0
19	<div><div></div></div>	5	<div><div></div></div>	3	<div><div></div></div>	2		0		0		0	<div><div></div></div>	25	<div><div></div></div>	5	<div><div></div></div>	20	<div><div></div></div>	13		0	<div><div></div></div>	13	<div><div></div></div>	0	<div><div></div></div>	0
20	<div><div></div></div>	3	<div><div></div></div>	2	<div><div></div></div>	1	<div><div></div></div>	1	<div><div></div></div>	1		0	<div><div></div></div>	18	<div><div></div></div>	2	<div><div></div></div>	16	<div><div></div></div>	4	<div><div></div></div>	2	<div><div></div></div>	2	<div><div></div></div>	0	<div><div></div></div>	0
21	<div><div></div></div>	10	<div><div></div></div>	3	<div><div></div></div>	7	<div><div></div></div>	3	<div><div></div></div>	3		0	<div><div></div></div>	15	<div><div></div></div>	5	<div><div></div></div>	10	<div><div></div></div>	8	<div><div></div></div>	3	<div><div></div></div>	5	<div><div></div></div>	0	<div><div></div></div>	0
22	<div><div></div></div>	4	<div><div></div></div>	4		0	<div><div></div></div>	2	<div><div></div></div>	2		0	<div><div></div></div>	14	<div><div></div></div>	1	<div><div></div></div>	13	<div><div></div></div>	7	<div><div></div></div>	2	<div><div></div></div>	5	<div><div></div></div>	0	<div><div></div></div>	0
23	<div><div></div></div>	8	<div><div></div></div>	5	<div><div></div></div>	3	<div><div></div></div>	4	<div><div></div></div>	4		0	<div><div></div></div>	26	<div><div></div></div>	5	<div><div></div></div>	21	<div><div></div></div>	12	<div><div></div></div>	3	<div><div></div></div>	9	<div><div></div></div>	0	<div><div></div></div>	0

Any  
Question

A decorative graphic consisting of several thin, white, parallel lines that originate from the bottom right corner and extend diagonally towards the top right corner of the slide.