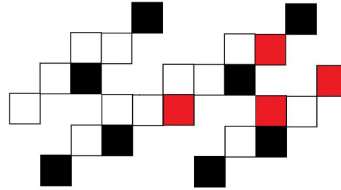


Computer Vision

Basics

Kakelino's Code School

Basics of Programming



Open CV demonstration

Traffic sign detection



Open CV demonstration

Traffic sign detection

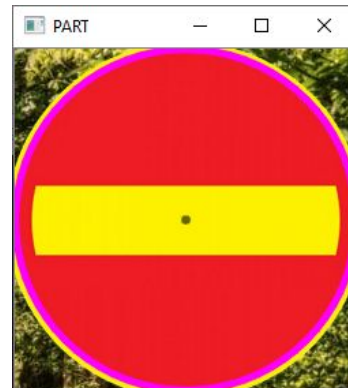
Can computer detect a circle shape from the picture



Open CV demonstration

Traffic sign detection

Can computer detect a circle shape from the picture



Now we wanted to detect only a circle from the left picture:

It was found.

Next step could be to detect colors...

Open CV demonstration

Traffic sign detection

Codes

We have a Visual Studio
console app and use C++

These header files were
added

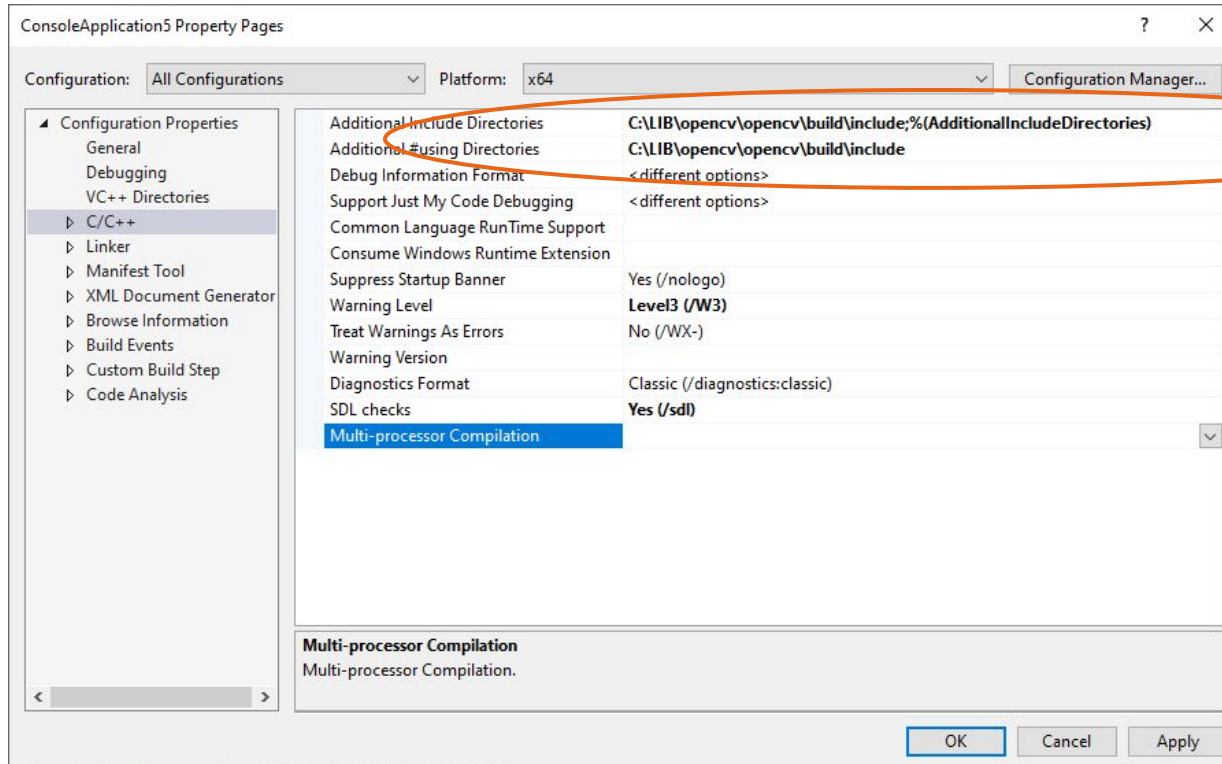
```
#include "pch.h"  
#include <iostream>  
#include <opencv2/core/core.hpp>  
#include <opencv2/highgui/highgui.hpp>  
#include <opencv2/imgproc/imgproc.hpp>  
#include <opencv2/highgui/highgui_c.h>  
#include "opencv2/objdetect/objdetect.hpp"  
#include "opencv2/features2d/features2d.hpp"  
#include <opencv2/ml/ml.hpp>  
#include "pre_img.h"
```

Open CV demonstration

Codes

Traffic sign detection

Take these libraries to
VS project



Open CV demonstration

Traffic sign detection

Codes

We have a Visual Studio console app and use C++

Main(): part 1

```
int main(int argc, char** argv)
{
    const char* filename = "c:\\kk\\nature3.png";
    // we load the image here to matrix src
    Mat src = imread(filename, IMREAD_COLOR);

    Mat gray;
    cvtColor(src, gray, COLOR_BGR2GRAY);
    medianBlur(gray, gray, 5);
    vector<Vec3f> circles;

    // here we give the measures: distances and radius
    HoughCircles(gray, circles, HOUGH_GRADIENT, 1,
        gray.rows / 3, 100, 70, 90, 130 );
    // read more info from OpenCV documentation

    Point center;
    Vec3i c;
    Mat copy = src.clone(); // make a copy
    for (size_t i = 0; i < circles.size(); i++)
    {
        c = circles[i];
        Point center = Point(c[0], c[1]);
        circle(src, center, 1, Scalar(0, 100, 100), 3, LINE_AA);

        int radius = c[2];
        circle(src, center, radius, Scalar(255, 0, 255), 3, LINE_AA);
    }
}
```

Open CV demonstration

Traffic sign detection

Codes

We have a Visual Studio console app and use C++

Main(): part 2

```
//print data for testing if needed
// cout << c[0] - c[2] << " " << c[1] - c[2] <<
//|" " << c[0] + c[2] << " " << c[1] + c[2] << endl;

Rect part(c[0] - c[2], c[1] - c[2], 2* c[2], 2 * c[2]);
Mat crop = src(part); // crop the circle if found

imshow("PART", crop);
imshow("Detected circles", src);
waitKey();
return 0;
}
```

Coding basics

Try it.

Make it better.

Detect colours ->
get more info
make decisions using that more
detailed info!

Coding basics

Give feedback, please!
Give ideas, proposals for new
guides!

C#

Java

JavaScript

C

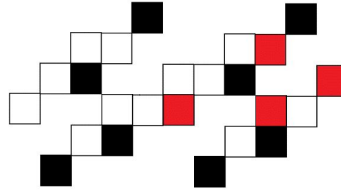
PHP

C++

Python

Kakelino's Code School

Basics of Programming



Open CV demonstration

Traffic sign detection, proto 2



Open CV demonstration

Traffic sign detection, proto 2

Official sign



Do we detect
It?



Open CV demonstration

Traffic sign detection, proto 2

Official sign



Codes

Source: opencv documentation

These are used in Visual Studio C++
Console project

```
#include "pch.h"  
#include <opencv2/highgui/highgui_c.h>  
#include "pre_img.h"
```

```
using namespace cv;  
using namespace std;
```

Open CV demonstration

Traffic sign detection, proto 2

Codes

Source: opencv documentation

Function that detects
squares

```
// find squares
static void findSquares(const Mat& image, vector<vector<Point> >& squares)
{
    squares.clear();
    Mat pyr, timg, gray0(image.size(), CV_8U), gray;
    pyrDown(image, pyr, Size(image.cols / 2, image.rows / 2));
    pyrUp(pyr, timg, image.size());
    vector<vector<Point> > contours;
    for (int c = 0; c < 3; c++)
    {
        int ch[] = { c, 0 };
        mixChannels(&timg, 1, &gray0, 1, ch, 1);
        for (int l = 0; l < N; l++)
        {
            if (l == 0)
            {
                Canny(gray0, gray, 0, thresh, 5);
                dilate(gray, gray, Mat(), Point(-1, -1));
            }
            else
            {
                gray = gray0 >= (l + 1) * 255 / N;
            }
        }
        // contours
        findContours(gray, contours, RETR_LIST, CHAIN_APPROX_SIMPLE);
        vector<Point> approx;

        for (size_t i = 0; i < contours.size(); i++)
        {
            approxPolyDP(contours[i], approx, arcLength(contours[i], true)*0.02, true);
            if (approx.size() == 4 &&
                fabs(contourArea(approx)) > 1000 &&
                isContourConvex(approx))
            {
                double maxCosine = 0;
                for (int j = 2; j < 5; j++)
                {
                    // find the maximum cosine of the angle between joint edges
                    double cosine = fabs(angle(approx[j % 4], approx[j - 2], approx[j - 1]));
                    maxCosine = MAX(maxCosine, cosine);
                }
                if (maxCosine < 0.3)
                    squares.push_back(approx);
            }
        }
    }
}
```

Open CV demonstration

Traffic sign detection, proto 2

Codes

Source: opencv documentation

Crop and draw

```
// we have only one square now, so it is drawn
static double drawSquares(Mat& image, const vector<vector<Point> >& squares)
{
    for (size_t i = 0; i < 1; i++)
    {
        const Point* p = &squares[i][0];
        int n = (int)squares[i].size();
        polylines(image, &p, &n, 1, true, Scalar(0, 255, 0), 1, LINE_AA);
    }

    // we get now the coordinates of the square to crop it - (Rect(...))
    int x1, x2, y1, y2;
    x1 = squares[0][0].x; y1 = squares[0][0].y;
    x2 = squares[0][2].x; y2 = squares[0][2].y;

    // take the sign from the bigger image
    Mat crop(image, Rect(x1,y1,x2-x1,y2-y1));

    imshow(newWindow, image);
    imshow("Part", crop); // show the sign
    imwrite("c:/kk/osanen.png", crop); // save it to a image file
    Mat orig = imread("c:/kk/merkkil.png"); // original, official traffic sign
    // for testing
    cout << "Similarity is " << getSimilarity(crop, orig);
    return getSimilarity(crop, orig);
}
```

Open CV demonstration

Traffic sign detection, proto 2

Codes

Source: opencv documentation

Are images similar

```
// If we find the traffic sign, we compare it to the official sign
double getSimilarity(const Mat A, const Mat B) {
    if (A.rows > 0 && A.rows == B.rows && A.cols > 0 && A.cols == B.cols) {
        double errorL2 = norm(A, B, CV_L2);
        double similarity = errorL2 / (double)(A.rows * A.cols);
        return similarity;
    }
    else {
        return -0.001;
    }
}
```

Open CV demonstration

Traffic sign detection, proto 2

Codes

Source: opencv documentation

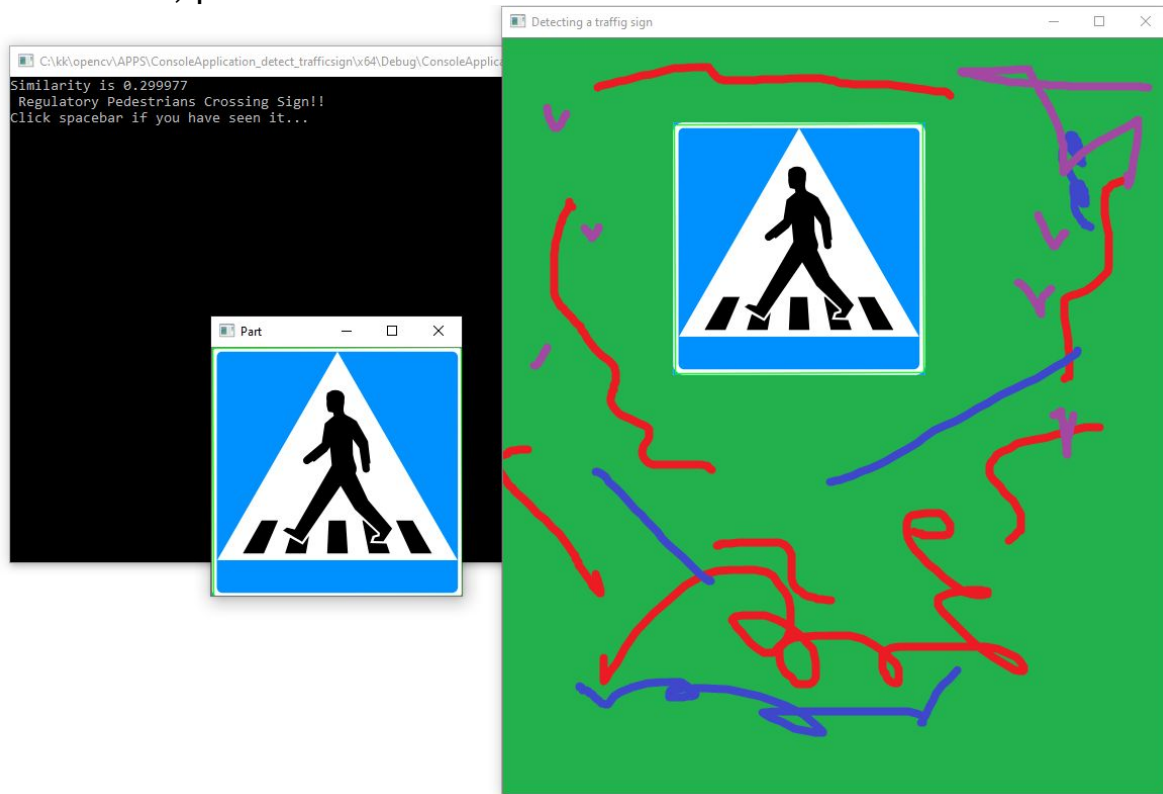
Function main()

```
int main(int argc, char** argv)
{
    static const char* name = "c:/kk/merkki2.png"; //the big image having also a traffic sign
    vector<vector<Point> > squares;
    for (int i = 0; i < 1; i++)
    {
        string filename = name;
        Mat image = imread(filename, IMREAD_COLOR);
        if (image.empty())
        {
            cout << "Couldn't load " << filename << endl;
            continue;
        }
        findSquares(image, squares);
        double result = drawSquares(image, squares);
        if (result > 0) {
            cout << "\007"; // beeps until noticed
            cout << "\007";
            cout << "\n Regulatory Pedestrians Crossing Sign!! \n";
            cout << "Click spacebar if you have seen it...\n";
            int c = waitKey();
            if (c == 32)
                break;
        }
    }
    return 0;
}
```

Open CV demonstration

Test run

Traffic sign detection, proto 2



Coding basics

Try it.

Make it better.

Detect colours ->
get more info
make decisions using that more
detailed info!

Coding basics

Give feedback, please!
Give ideas, proposals for new guides!

C#

Java

JavaScript

C

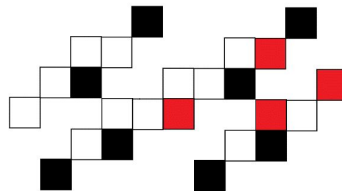
PHP

C++

Python

Kakelino's Code School

Basics of Programming



Open CV demonstration

Detect colors

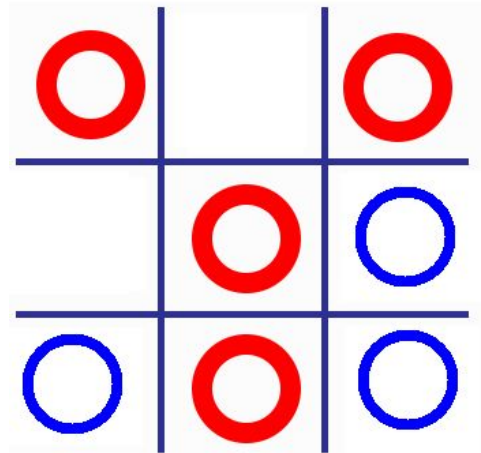
Source:

OpenCV documentation

Open CV demonstration

Detect colors

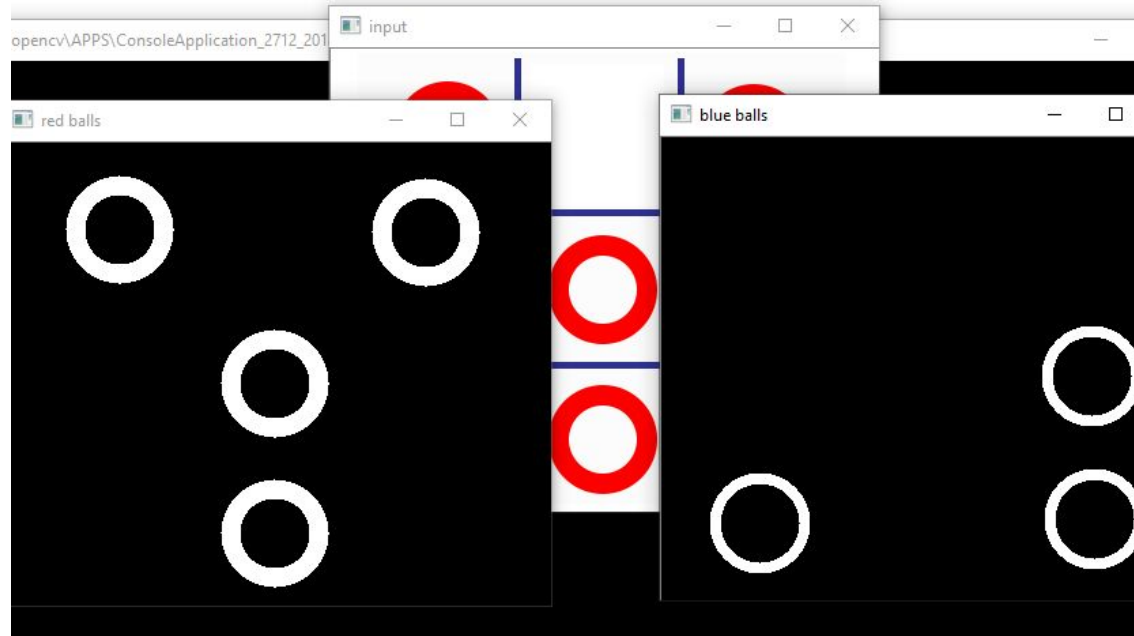
Tictactoe: situation



Open CV demonstration

Detect colors

Tictactoe: situation

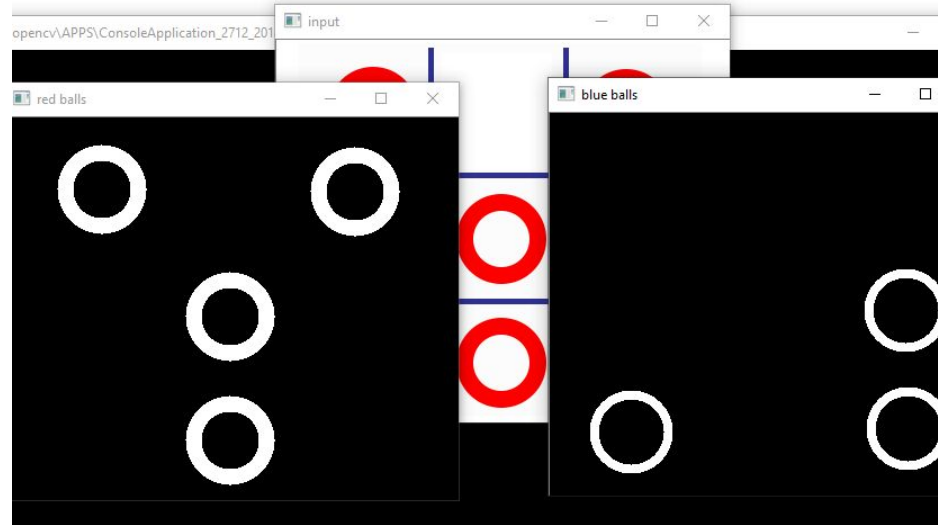


Open CV demonstration

Detect colors

OpenCV functions
can detect red
and blue balls

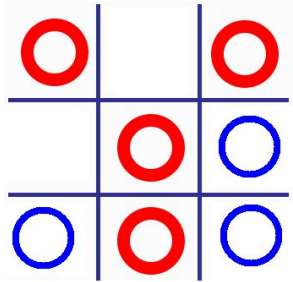
Tictactoe: situation



Open CV demonstration

Detect colors

Tictactoe: situation



Codes

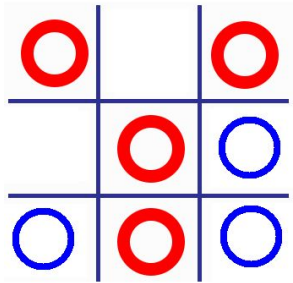
These headerfiles are added to Visual Studio console project

```
#include "pch.h"  
#include "opencv2/highgui.hpp"  
#include <iostream>  
  
using namespace cv;  
using namespace std;
```

Open CV demonstration

Detect colors

Tictactoe: situation



Codes

Here are the important functions that detect balls

```
Mat findReds(const Mat& src)
{
    Mat reds;
    inRange(src, Scalar(0, 0, 0), Scalar(0, 0, 255), reds);

    return reds;
}
```

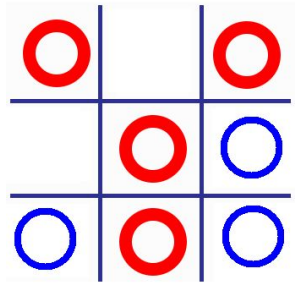
```
Mat findBlues(const Mat& src)
{
    Mat blues;
    inRange(src, Scalar(0, 0, 0), Scalar(255, 0, 0), blues);

    return blues;
}
```

Open CV demonstration

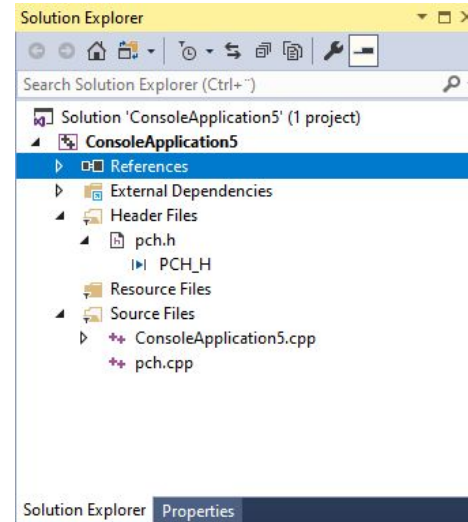
Detect colors

Tictactoe: situation



Codes

Included pch files



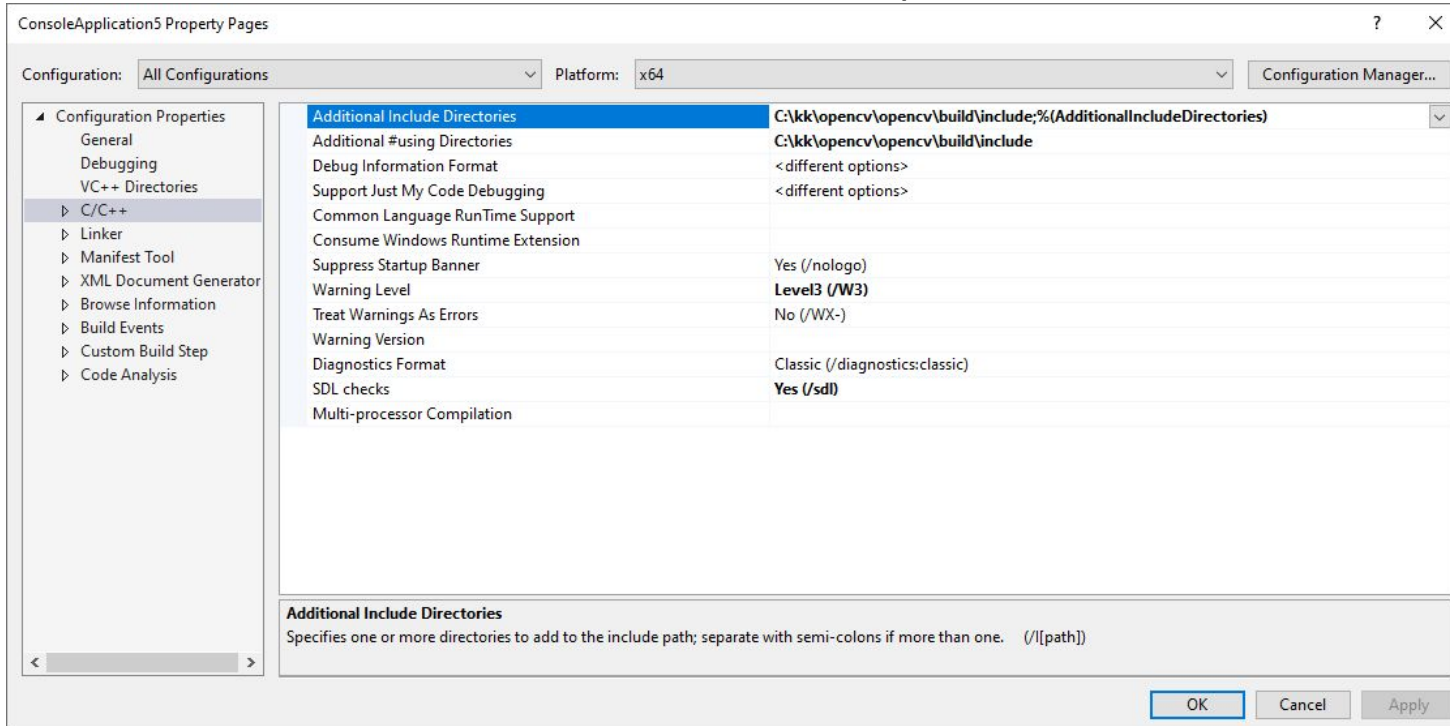
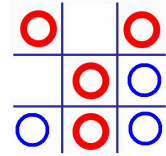
Open CV demonstration

Detect colors

Tictactoe: situation

Codes

Additional directories needed
for opencv:

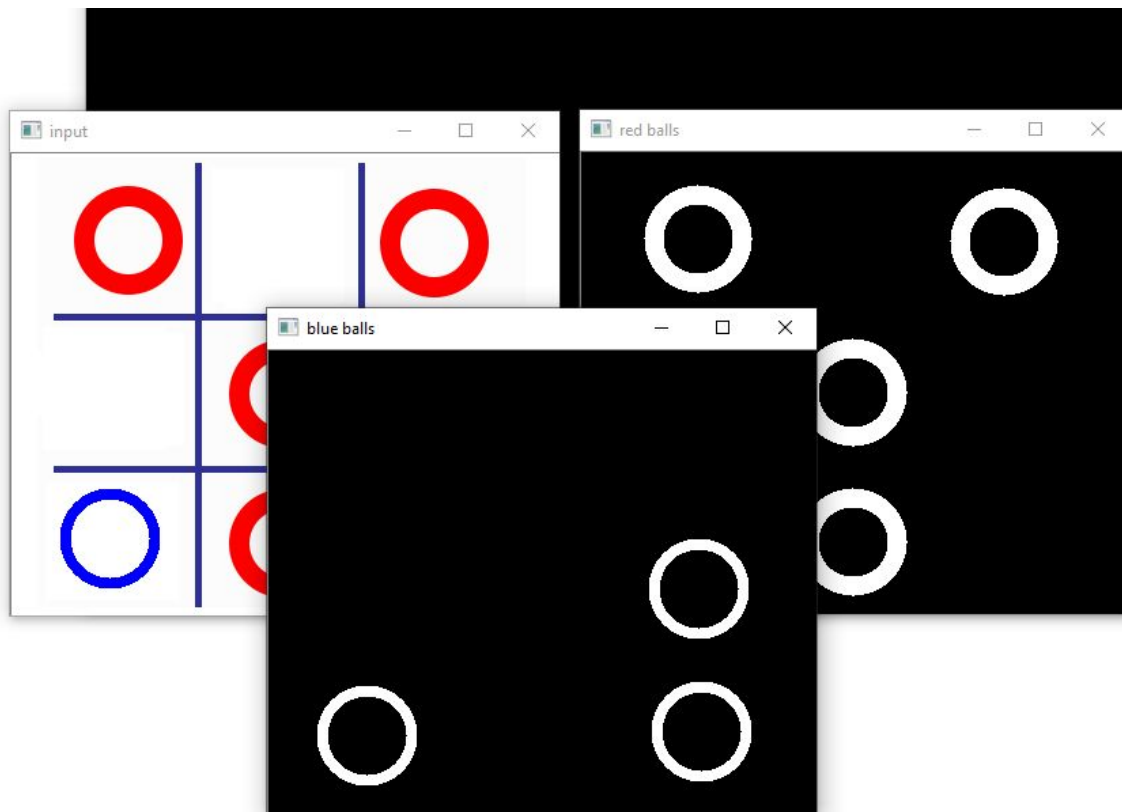
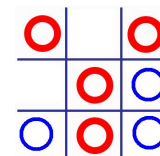


Open CV demonstration

Detect colors

Tictactoe: situation

Test run



Coding basics

Try it.

Make it better.

Detect more colours ->
get more info
make decisions using that more
detailed info!

Coding basics

Give feedback, please!
Give ideas, proposals for new guides!

C#

Java

JavaScript

C

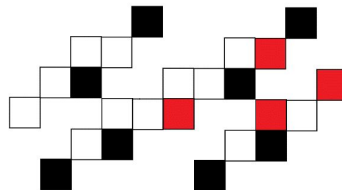
PHP

C++

Python

Kakelino's Code School

Basics of Programming



Open CV demonstration

Detect faces and expressions



Sources:
OpenCV documentation

The goal is to test
Karan's code...

<https://www.paulekman.com/about/paul-ekman/>

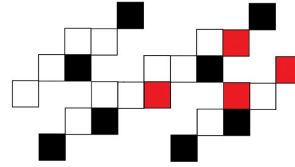
https://www.rcciit.org/students_projects/projects/it/2018/GR8.pdf

Main code that is tested in this presentation comes from

<https://medium.com/swlh/emotion-detection-using-opencv-and-keras-771260bbd7f7>

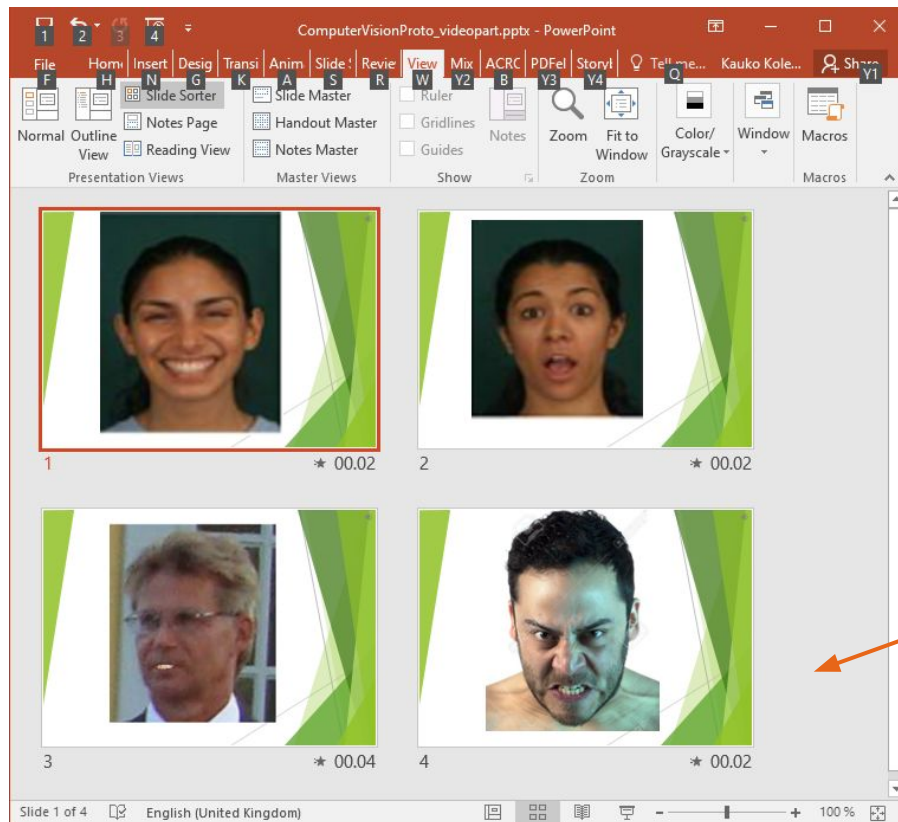
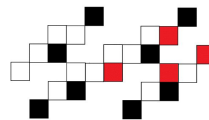
Thanks to Karan Sethi

Kakelino's Code School



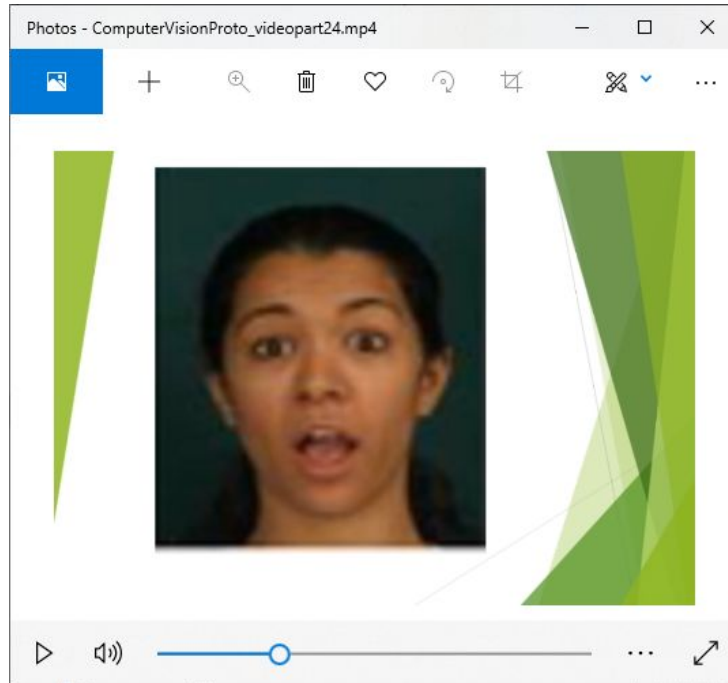
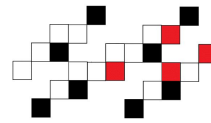
Here is the powerpoint document that is used as an image collection: it is exported to a video instead of using a live video (laptop cam)

Kakelino's Code School



Unfortunately source not know,
hopefully person is happier if
we meet later...

Kakelino's Code School

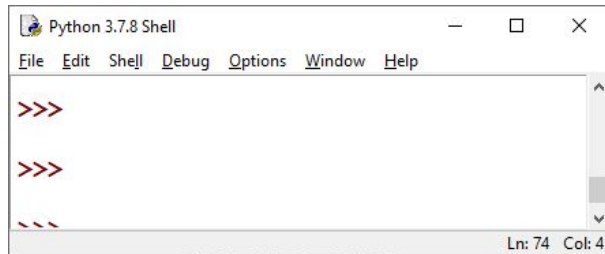


Open CV demonstration

Detect faces and expressions

Now, main codes...

We use python and now the version was
Python 3.7.8...



Open CV demonstration

Detect faces and expressions

Now, main codes...

To python environment you have to install
tensorflow, keras and cv2.
Also pillow is needed...

Sometimes it is better to upgrade instead
of trying to install modules

Tool/command pip is used for installing...

Open CV demonstration

Detect faces and expressions

Now, main codes...

Some adjustments had to be done because of my newer mdoule versions...

```
from tensorflow.python.keras.backend import set_session  
sess = tf.compat.v1.Session()  
graph = tf.compat.v1.get_default_graph()  
set_session(sess)
```

Open CV demonstration

Now, main codes...

Detect faces and expressions

```
faces2020.py - Notepad
File Edit Format View Help
from keras.models import load_model
from time import sleep
from keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np

face_classifier=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
classifier = load_model('EmotionDetectionModel.h5')

class_labels=['Angry','Happy','Neutral','Sad','Surprise']
cap = cv2.VideoCapture("ComputerVisionProto_videopart24.mp4")

while True:
    ret,frame=cap.read()
    labels=[]
    # if ret == True:
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces=face_classifier.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray=gray[y:y+h,x:x+w]
        roi_gray=cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)

        if np.sum([roi_gray])!=0:
            roi=roi_gray.astype('float')/255.0
            roi=img_to_array(roi)
            roi=np.expand_dims(roi,axis=0)

            preds=classifier.predict(roi)[0]
            label=class_labels[preds.argmax()]
            label_position=(x,y)
            cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
        else:
            cv2.putText(frame,'No Face Found',(20,20),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)

    cv2.imshow('Emotion Detector',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

Open CV demonstration

Detect faces and expressions

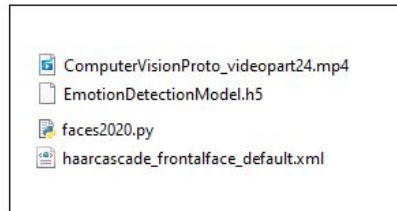
Now, main codes...

Code from
Karan Sethi

Thank you!

- code is
published in
github

All needed files are now in
the same folder



Training and validation data is in static
subfolder



Open CV demonstration

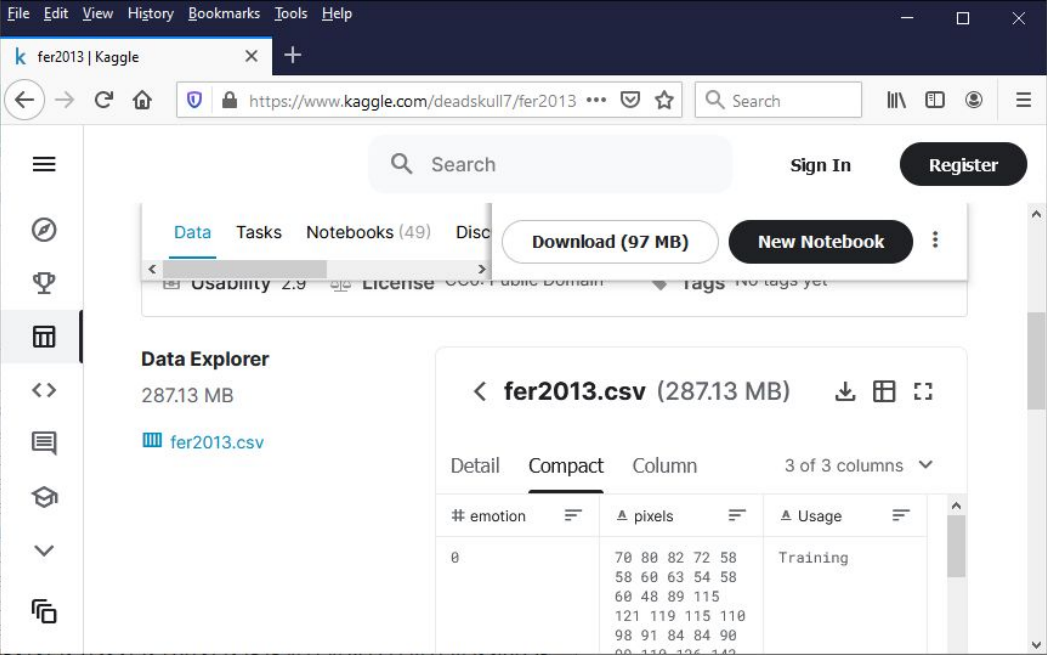
Detect faces and expressions

Now, main codes...

Expression database is fer2012.csv

Code from
Karan Sethi
Thank you!

- code is published in github



The screenshot shows the Kaggle website interface. The browser address bar displays the URL <https://www.kaggle.com/deadskull7/fer2013>. The page features a navigation bar with 'Data', 'Tasks', 'Notebooks (49)', and 'Discussions'. Below this, a 'Data Explorer' section highlights the 'fer2013.csv' dataset, which is 287.13 MB in size. To the right, a detailed view of the dataset is shown, including a table with columns for '# emotion', 'pixels', and 'Usage'. The 'pixels' column contains a 4x4 grid of numbers representing face coordinates. The 'Usage' column indicates the dataset is used for 'Training'.

# emotion	pixels	Usage
0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 80 110 106 110	Training

Open CV demonstration

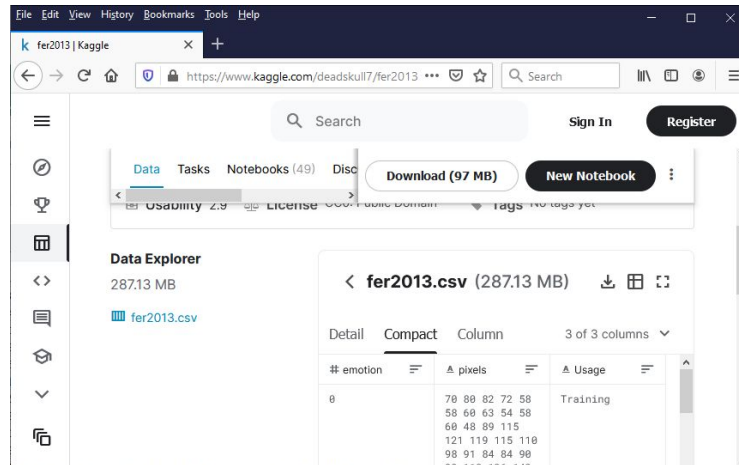
Detect faces and expressions

Now, main codes...

Code from
Karan Sethi
Thank you!

- code is published in github

Expression database is fer2012.csv



The screenshot shows the Kaggle interface for the 'fer2013' dataset. The 'Data Explorer' section displays the dataset 'fer2013.csv' with a size of 287.13 MB. The 'Compact' view shows a table with three columns: '# emotion', 'pixels', and 'Usage'. The 'pixels' column contains a 10x5 grid of numbers representing face coordinates. The 'Usage' column shows the dataset is used for 'Training'.

# emotion	pixels	Usage
0	79 80 82 72 58 58 69 63 54 58 68 48 89 115 121 119 115 118 98 91 84 84 98 20 110 106 143	Training

It is used to get training and validation data.

Open CV demonstration

Detect faces and expressions

Now, main codes...

Code from
Karan Sethi
Thank you!

- code is published in github

Expression database is fer2012.csv

It is used to get training and validation data.

Sample in Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	emotion,pixels,Usage											
2	0,70	80	82	72	58	58	60	63	54	58	60	48
3	0,15	150	147	155	148	133	111	140	170	174	182	154
4	2,23	212	156	164	174	138	161	173	182	200	106	38
5	4,24	32	36	30	32	23	19	20	30	41	21	22
6	6,4	0	0	0	0	0	0	0	0	3	15	23
7	2,55	55	55	55	54	60	68	54	85	151	163	170
8	4,20	17	19	21	25	38	42	46	54	56	62	63
9	3,77	78	79	78	75	60	55	47	48	58	73	77
10	3,85	84	30	121	101	102	133	153	153	163	177	183
11	2,25	254	255	254	254	173	122	107	35	124	143	150
12	0,30	24	21	23	25	25	43	67	84	103	120	125
13	6,32	75	78	58	50	45	43	48	103	156	81	45
14	6,219	213	206	202	209	217	216	215	219	216	223	230
15	6,148	144	130	123	113	122	123	131	133	153	140	128
16	3,4	2	13	41	56	62	67	67	65	70	80	107

Open CV demonstration

Detect faces and expressions

Now, main codes...

Code from
Karan Sethi

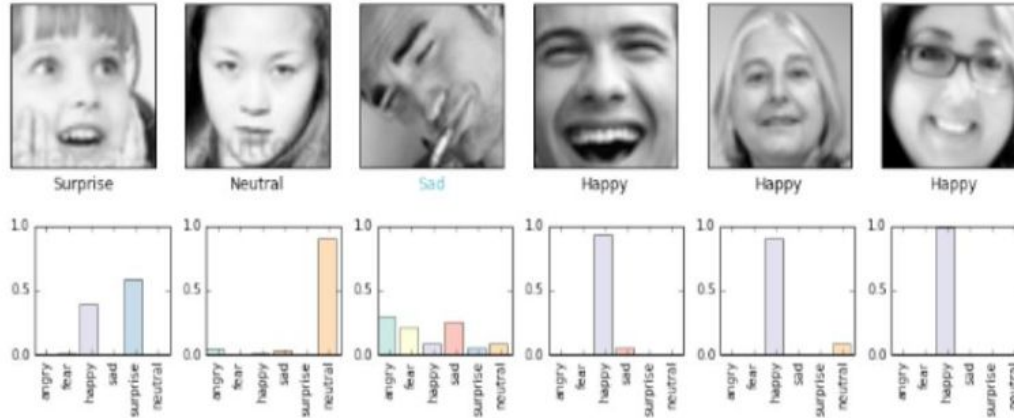
Thank you!

- code is published in github

Expression database is fer2012.csv

It is used to get training and validation data.

Example of model validation



Open CV demonstration

Detect faces and expressions

There are several steps to go through when setting up face and expression detecting system....

Good instructions can be found from Internet, OpenCV documents and published codes..

Good sources were given also at the beginning of this presentation

Open CV demonstration

Detect faces and expressions

There are several steps to go through when setting up face and expression detecting system....

Good instructions can be found from Internet, OpenCV documents and published codes..

Good sources were given also at the beginning of this presentation

Now the main goal.
We wanted to test if our code can detect expressions from our own video!

Open CV demonstration

Detect faces and expressions

Now the main goal.
We wanted to test if our
code can detect
expressions from our
own video!

Here is a life presentation



Small joke: these are too
difficult to detect or separate
(source John Liggings)

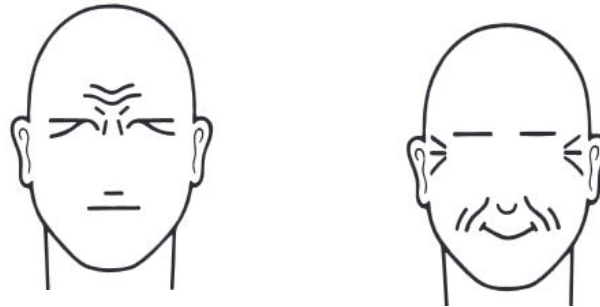
Open CV demonstration

Detect faces and expressions

Now the main goal.
We wanted to test if our
code can detect
expressions from our
own video!

Here is a life presentation

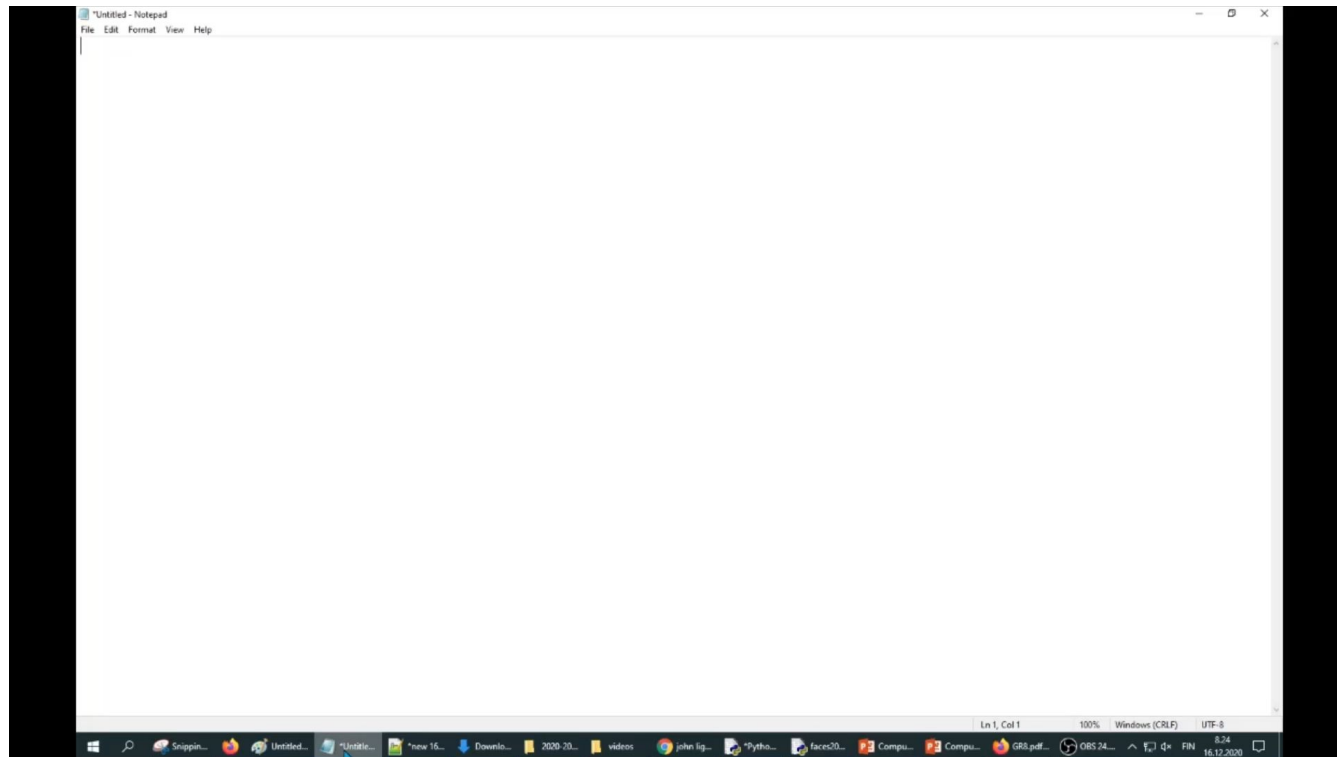
What about these?



Open CV demonstration

Detect faces and expressions

Here is a life presentation



Coding basics

Try it.

Make it better.

Detect more colours ->
get more info
make decisions using that more
detailed info!

Coding basics

Give feedback, please!
Give ideas, proposals for new guides!

C#

Java

JavaScript

C

PHP

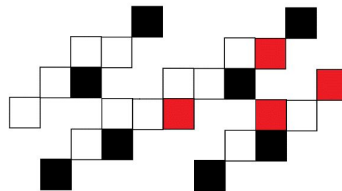
C++

Python



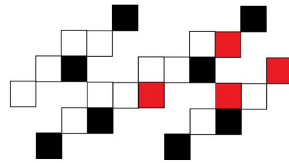
Kakelino's Code School

Basics of Programming





Kakelino's Code School

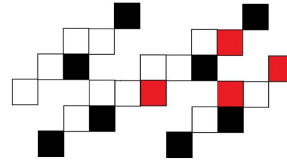


Open CV demonstration

Detecting chords

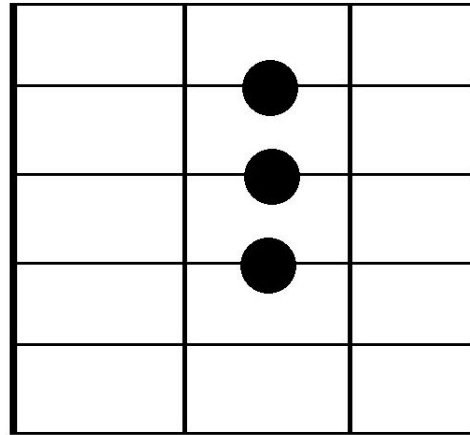


Kakelino's Code School



Open CV demonstration

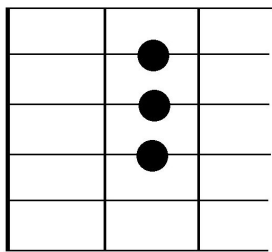
Detecting chords



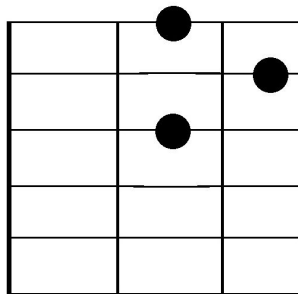
2nd fret

Open CV demonstration

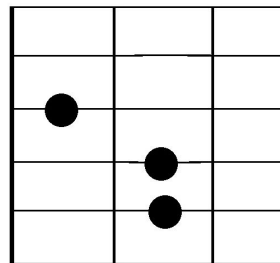
Detecting chords



2nd fret



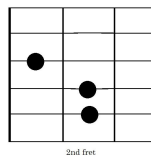
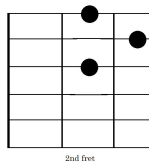
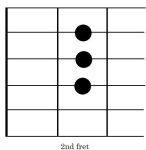
2nd fret



2nd fret

Open CV demonstration

Detecting chords

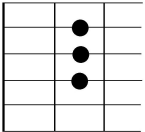


Open CV demonstration

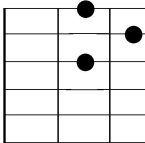
Detecting chords

We have only 3 chords here,
they are played with a guitar

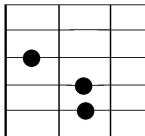
...



2nd fret



2nd fret



2nd fret

Codes

Source is opencv

We have a Visual Studio
C++ Console project.

These header files are
added first

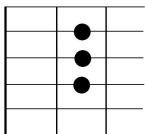
```
#include "pch.h"  
#include <opencv2/highgui/highgui_c.h>  
#include "opencv2/objdetect/objdetect.hpp"  
#include "opencv2/features2d/features2d.hpp"  
#include "pre_img.h"  
  
using namespace cv;  
using namespace std;
```

Open CV demonstration

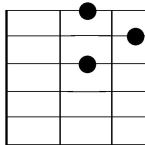
Detecting chords

We have only 3 chords here,
they are played with a guitar

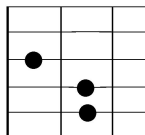
...



2nd fret



2nd fret



2nd fret

Codes

Source is opencv

Function main()

A)

```
int main()
{
    // https://docs.opencv.org
    Mat src1, src2, src3, gray1, gray2, gray3;
    src1 = imread("A_major.png", 1);
    resize(src1, src1, Size(640, 480));
    src2 = imread("D_major.png", 1);
    resize(src2, src2, Size(640, 480));
    src3 = imread("E_major.png", 1);
    resize(src3, src3, Size(640, 480));

    cvtColor(src1, gray1, CV_BGR2GRAY);
    GaussianBlur(gray1, gray1, Size(9, 9), 2, 2);
    cvtColor(src2, gray2, CV_BGR2GRAY);
    GaussianBlur(gray2, gray2, Size(9, 9), 2, 2);
    cvtColor(src3, gray3, CV_BGR2GRAY);
    GaussianBlur(gray3, gray3, Size(9, 9), 2, 2);

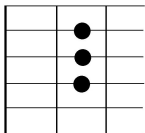
    vector<Vec3f> circles;
    HoughCircles(gray1, circles, CV_HOUGH_GRADIENT, 1, 10, 200, 50, 0, 50);
```

Open CV demonstration

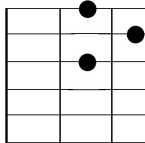
Detecting chords

We have only 3 chords here,
they are played with a guitar

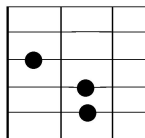
...



2nd fret



2nd fret



2nd fret

Codes

Source is opencv

Function main()
B)

```
int x1 = 0; int x2 = 0; int x3 = 0; int x4 = 0; int x5 = 0; int x6 = 0;
for (size_t i = 0; i < circles.size(); i++)
{
    Point center(cvRound(circles[i][0]), cvRound(circles[i][1]));
    int radius = cvRound(circles[i][2]);
    circle(src1, center, 3, Scalar(0, 255, 0), -1, 8, 0); // circle center
    circle(src1, center, radius, Scalar(0, 0, 255), 3, 8, 0); // circle outline
    cout << "center : " << center.x << ", " << center.y << "\nradius : " << radius << endl;
}

play(circles);

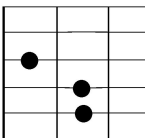
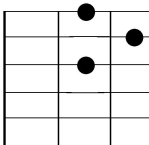
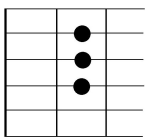
namedWindow("Hough Circle Transform Demo", CV_WINDOW_AUTOSIZE);
imshow("Demonstration - applying to guitar chords", src1);

waitKey(0);
return 0;
```

Open CV demonstration

Detecting chords

We have only 3 chords here,
they are played with a guitar ...



Codes

Source is opencv

Function play()

```
string chords[3];
int s = 0;

void play(vector<Vec3f> circles)
{
    // just testing chord A, suppose fret is 2
    int fret1 = 0; int line1 = 0; int fret2 = 0; int line2 = 0;
    int fret3 = 0; int line3 = 0;
    if (circles[0][0] > 300 && circles[0][0] < 490) // fret is 2
        if (circles[0][1] > 50 && circles[0][1] < 150) // line is 2
        {
            line1 = 2; fret1 = 2;
        }
    if (circles[1][0] > 300 && circles[1][0] < 490) // fret is 2
        if (circles[1][1] > 150 && circles[1][1] < 250) // line is 2
        {
            line2 = 3; fret2 = 2;
        }
    if (circles[2][0] > 300 && circles[2][0] < 490) // fret is 2
        if (circles[2][1] > 250 && circles[2][1] < 350) // line is 2
        {
            line3 = 4; fret3 = 2;
        }
    if (fret1 == 2 && fret2 == 2 && fret3 == 2)
        if (line1 == 2 && line2 == 3 && line3 == 4)
        {
            chords[s] = "A"; s++;
        }
    // using same brute force logic, we get from images that present major chords D and E
    chords[s] = "D"; s++;
    chords[s] = "E"; s++;
    for (int i = 0; i < s; i++)
        cout << chords[i] << " ";

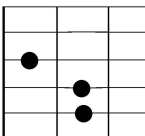
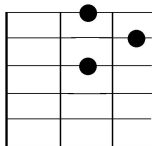
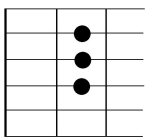
    cout << endl << "let's listen..."; // for testing
    system("c:\\kk\\mm.bat");
}
```

Open CV demonstration

Codes
Source is opencv

Detecting chords

We have only 3 guitar chords here



In this proto, chords are played with Windows Media Player

We use just a batch-file to open that player 😊

```
system("c:\\codes\\mm.bat");
```

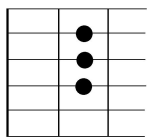
```
mm.bat - Notepad
File Edit Format View Help
cd "C:\\Program Files (x86)\\Windows Media Player"
wmplayer.exe c:\\codes\\ADE.mp3
Ln 2, Col 23 90% Windows (CRLF) UTF-8
```

Open CV demonstration

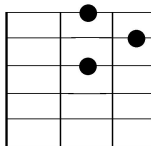
Codes
Source is opencv

Detecting chords

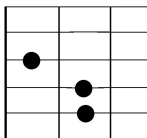
Test run



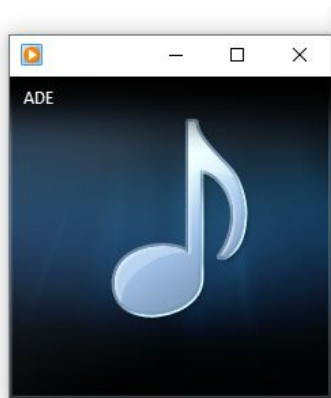
2nd fret



2nd fret



2nd fret



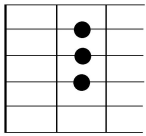
```
radius : 24  
center : 330, 190  
radius : 24  
center : 326, 266  
radius : 24  
A D E  
let's listen...  
C:\Program Files (x86)\Windows Media Player>wmplayer.exe c:  
mp3
```

Open CV demonstration

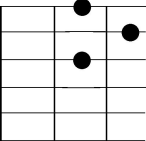
Codes
Source is opencv

Detecting chords

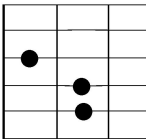
Test run



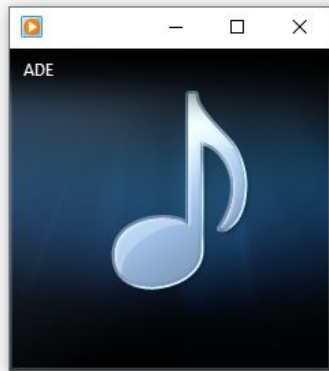
2nd fret



2nd fret



2nd fret



```
radius : 24  
center : 330, 190  
radius : 24  
center : 326, 266  
radius : 24  
A D E  
let's listen...  
C:\Program Files (x86)\Windows Media Player>wmplayer.exe c:\  
mp3
```

Listen:



To easily get guitar chords were some tools used

The screenshot shows the Chordbook.com website interface. The browser tab is titled "Guitar Chords with Chordbook". The address bar shows the URL "chordbook.com/guitar-chords/". The website header includes the "chordbook.com" logo and navigation links for "BLOG", "SHOP", "LOG IN", and "SIGNUP". Below the header is a navigation bar with links for "guitar chords", "guitar scales", "guitar tuner", and "learn guitar", along with a Facebook icon and a "guitar settings" link.

The main content area displays the "E major" chord page. On the left, there is a fretboard diagram showing the E major chord shape with fingerings: 1st fret on the 1st string, 2nd fret on the 2nd string, and 3rd fret on the 3rd string. To the right of the fretboard, the text "E major" is displayed, followed by a "more >" link. Below this, there are two sections: "SELECT ROOT" and "SELECT TYPE".

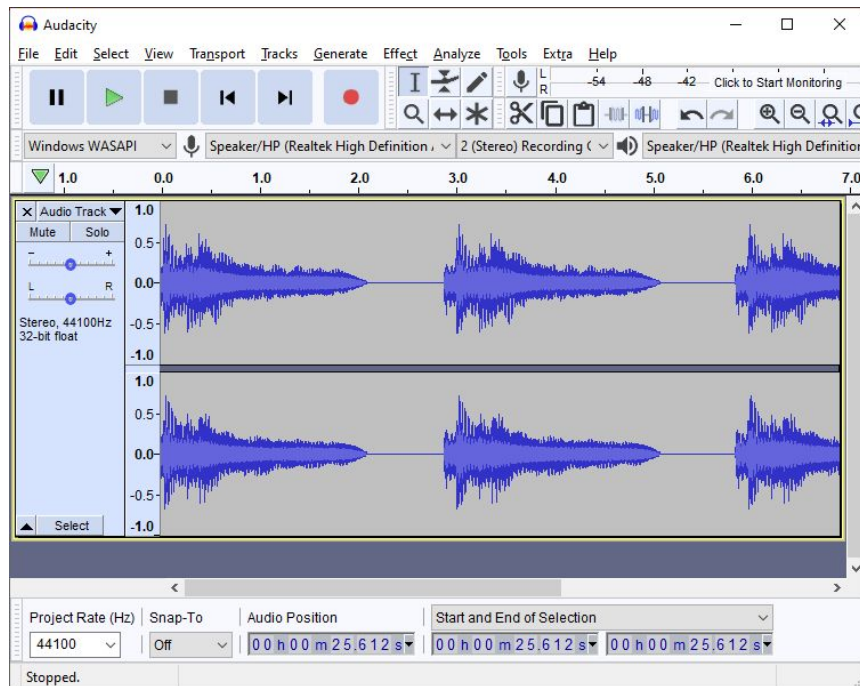
SELECT ROOT

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
	D#		E#		G#		A#		B#		

SELECT TYPE

maj	5	6	7	maj7
9	maj9	11	13	maj13
min	m6	m7	m9	m11
m13	m(maj7)	sus2	sus4	dim
aug	6/9	7sus4	7b5	7b9
9sus4	add9	aug9		

To easily get guitar chords were some tools used



To easily get guitar chords were some tools used...

Sample here: A D and E chords were used:

Second Hand News

A D A
I know there's nothing to say
 D A
Someone has taken my place
 E A E
When times go bad, when times go rough,
 A D
won't you lay me down in the tall grass,
 E
and let me do my stuff?

Listen a sample



Coding basics

Try it.

Make it more universal...

Develop the logic...

Add video based chord detection...

Coding basics

Give feedback, please!
Give ideas, proposals for new
guides!

C#

Java

JavaScript

C

PHP

C++

Python