Technological Institute of the Philippines	Quezon City - Computer Engineering
Course Code:	CPE 018
Code Title:	Emerging Technologies in CpE 1 - Fundamentals of Computer Vision
1st Semester	AY 2023-2024
ACTIVITY NO.	**TITLE**
Name	Pisalbon, Ery Jay P.
Section	CPE32S3
Date Performed:	02/21/2025
Date Submitted:	02/21/2025
Instructor:	Dr. Jonathan V. Taylar / Engr. Verlyn V. Nojor / Engr. Roman M. Richard

1. Objectives

This activity aims to enable students to perform data preparation and face recognition on their own generated dataset.

2. Intended Learning Outcomes (ILOs)

After this activity, the students should be able to:

- Utilize data preparation techniques for images.
- Perform Face Recognition using multiple algorithms.
- Evaluate the performance of different algorithms.

3. Procedures and Outputs

Preparing the training data

Now that we have our data, we need to load these sample pictures into our face recognition algorithms. All face recognition algorithms take two parameters in their `train()` method: an array of images and an array of labels. What do these labels represent? They are the IDs of a certain individual/face so that when face recognition is performed, we not only know the person was recognized but also who—among the many people available in our database—the person is.

To do that, we need to create a comma-separated value (CSV) file, which will contain the path to a sample picture followed by the ID of that person.

Include a Screenshot of Your Dataset Here

Loading the data and recognizing faces

Next up, we need to load these two resources (the array of images and CSV file) into the face recognition algorithm, so it can be trained to recognize our face. To do this, we build a function that reads the CSV file and—for each line of the file—loads the image at the corresponding path into the images array and the ID into the labels array.

In [3]:
```
import numpy as np
import os
import errno
import sys
import cv2

def read_images(path, sz=None):
  c = 0
  X, y = [], []

  for dirname, dirnames, filenames in os.walk(path):
    for subdirname in dirnames:
      subject_path = os.path.join(dirname, subdirname)
      for filename in os.listdir(subject_path):
        try:
          if(filename == ".directory"):
            continue
          filepath = os.path.join(subject_path, filename)
          im = cv2.imread(os.path.join(subject_path, filename), cv2.IMREAD_GRAYSCAL

          # Resize the images to the prescribed size
          if (sz is not None):
            im = cv2.resize(im, (200,200))

          X.append(np.asarray(im, dtype=np.uint8))
          y.append(c)

        except IOError as e:
          print(f"I/O Error({e.errno}): {e.strerror}")
        except:
          print("Unexpected error:", sys.exc_info()[0])
          raise
      c = c+1

  return [X, y]
```

In [2]:
```
import numpy as np
import csv
```

```
def save_images_and_labels_to_csv(images, labels, csv_filename="dataset.csv"):
    # Ensure images and labels have the same length
    assert len(images) == len(labels), "Mismatch between number of images and label

    with open(csv_filename, mode='w', newline='') as file:
        writer = csv.writer(file)

        # Write header (optional)
        num_pixels = images[0].size  # Get the number of pixels in an image
        header = ["label"] + [f"pixel{i}" for i in range(num_pixels)]
        writer.writerow(header)

        # Write each image and its corresponding label
        for img, label in zip(images, labels):
            img_flattened = img.flatten()  # Convert image to 1D array
            row = np.insert(img_flattened, 0, label)  # Insert label at the beginni
            writer.writerow(row)

    print(f"Images and labels saved to {csv_filename}")

# Example usage
# Assuming `images` is a list of NumPy arrays (grayscale images)
# And `labels` is a list of corresponding labels

# Example images and labels (assuming grayscale 200x200 images)
```

In [3]:
```
images, labels = read_images("Act 7",1)
save_images_and_labels_to_csv(images, labels, "act_7_dataset.csv")
```

```
Images and labels saved to act_7_dataset.csv
```

In [4]:
```
print(images)
```

```
[array([[ 7, 10,  9, ..., 52, 50, 47],
       [11, 12, 17, ..., 64, 61, 63],
       [15, 19, 21, ..., 70, 72, 71],
       ...,
       [80, 86, 91, ..., 45, 46, 39],
       [83, 91, 93, ..., 46, 43, 43],
       [ 2, 81, 94, ..., 45, 42, 39]], dtype=uint8), array([[51, 49, 50, ..., 25, 3
2, 29],
       [49, 46, 49, ..., 24, 29, 26],
       [49, 45, 49, ..., 28, 29, 25],
       ...,
       [16, 16, 13, ..., 23, 22, 22],
       [13, 10, 10, ..., 23, 20, 20],
       [17, 13, 13, ..., 22, 20, 19]], dtype=uint8), array([[ 7,  3,  2, ...,  0,
0,  0],
       [ 3,  4,  0, ...,  0,  0,  0],
       [ 2,  0,  0, ...,  0,  0,  0],
       ...,
       [53, 48, 47, ..., 47, 41, 45],
       [51, 54, 51, ..., 76, 61, 55],
       [52, 59, 64, ..., 65, 74, 56]], dtype=uint8), array([[128, 195, 199, ..., 18
5, 188, 126],
       [134, 201, 195, ..., 184, 187, 131],
       [139, 204, 197, ..., 189, 195, 133],
       ...,
       [ 74, 108, 101, ..., 183, 183, 124],
       [ 72, 106, 103, ..., 177, 182, 125],
       [ 69, 104, 103, ..., 175, 178, 119]], dtype=uint8), array([[44, 50, 52, ...,
34, 37, 48],
       [38, 44, 48, ..., 39, 44, 49],
       [34, 40, 48, ..., 38, 41, 47],
       ...,
       [50, 55, 53, ..., 36, 33, 33],
       [51, 54, 55, ..., 35, 35, 32],
       [51, 54, 55, ..., 33, 33, 35]], dtype=uint8), array([[ 39,  38,  36, ..., 12
0, 122, 121],
       [ 37,  36,  34, ..., 121, 123, 127],
       [ 36,  35,  33, ..., 127, 132, 129],
       ...,
       [ 33,  31,  28, ..., 121, 121, 120],
       [ 30,  28,  25, ..., 122, 124, 122],
       [ 28,  26,  24, ..., 124, 126, 124]], dtype=uint8), array([[ 48,  46,  43,
..., 118, 114, 111],
       [ 53,  51,  46, ..., 122, 121, 116],
       [ 52,  51,  47, ..., 128, 130, 123],
       ...,
       [138, 153, 162, ...,  89,  89,  91],
       [140, 157, 163, ...,  81,  83,  87],
       [146, 161, 165, ...,  74,  80,  82]], dtype=uint8), array([[19, 17, 16, ...,
12, 11,  7],
       [17, 16, 16, ..., 11, 11,  8],
       [16, 16, 16, ..., 11, 10, 10],
       ...,
       [83, 82, 78, ..., 33, 35, 38],
       [83, 82, 78, ..., 33, 35, 39],
       [82, 82, 79, ..., 31, 36, 39]], dtype=uint8), array([[ 27,  25,  28, ...,  3
```

```
0,  29,  28],
       [ 27,  26,  29, ...,  30,  32,  30],
       [ 27,  25,  28, ...,  30,  33,  31],
       ...,
       [130, 101,  85, ...,  93, 102, 104],
       [129,  96,  81, ...,  93, 102, 101],
       [128,  98,  77, ...,  96,  94, 103]], dtype=uint8), array([[91, 89, 86, ...,
28, 24, 20],
       [93, 86, 81, ..., 28, 24, 20],
       [91, 87, 83, ..., 27, 25, 21],
       ...,
       [43, 45, 42, ..., 44, 47, 46],
       [43, 43, 41, ..., 45, 44, 43],
       [45, 45, 41, ..., 42, 39, 40]], dtype=uint8), array([[ 42,  37,  30, ..., 11
4, 118, 122],
       [ 46,  39,  31, ..., 107, 113, 120],
       [ 50,  42,  35, ...,  98, 105, 116],
       ...,
       [ 24,  21,  23, ..., 198, 173, 128],
       [ 32,  29,  29, ..., 202, 180, 137],
       [ 36,  33,  31, ..., 207, 185, 144]], dtype=uint8), array([[160, 152, 151,
...,  12,  14,  17],
       [171, 168, 167, ...,  14,  16,  18],
       [180, 175, 175, ...,  16,  18,  19],
       ...,
       [ 63,  77,  70, ..., 185, 187, 185],
       [ 69,  72,  72, ..., 190, 202, 179],
       [ 60,  68,  72, ..., 190, 200, 183]], dtype=uint8), array([[142, 142, 142,
..., 129, 126, 121],
       [142, 142, 142, ..., 129, 126, 121],
       [142, 142, 142, ..., 130, 128, 123],
       ...,
       [135, 135, 137, ..., 107, 104,  99],
       [136, 136, 136, ..., 106, 104,  98],
       [135, 135, 136, ..., 105, 103,  97]], dtype=uint8), array([[144, 135, 127,
...,  35,  35,  34],
       [150, 144, 137, ...,  35,  35,  35],
       [159, 156, 148, ...,  35,  35,  35],
       ...,
       [ 41,  39,  37, ...,  72,  66,  62],
       [ 41,  39,  37, ...,  65,  66,  62],
       [ 37,  33,  36, ...,  67,  69,  64]], dtype=uint8), array([[ 0,  0,  0, ...,
0,  0,  0],
       [ 0,  0,  0, ...,  0,  0,  0],
       [ 0,  0,  0, ...,  0,  0,  0],
       ...,
       [10, 13, 15, ..., 47, 49, 54],
       [11, 13, 15, ..., 48, 48, 57],
       [11, 12, 16, ..., 62, 52, 54]], dtype=uint8), array([[157, 158, 155, ...,  1
1,   7,   4],
       [159, 158, 159, ...,  11,   7,   3],
       [160, 160, 162, ...,  11,   7,   3],
       ...,
       [131, 121, 107, ...,  33,  33,  32],
       [127, 119, 105, ...,  33,  30,  30],
       [123, 115, 101, ...,  31,  31,  30]], dtype=uint8), array([[ 17,  18,  18,
```

```
...,  32,  33,  32],
       [ 21,  22,  22, ...,  32,  32,  32],
       [ 27,  27,  27, ...,  32,  33,  33],
       ...,
       [ 77,  78,  78, ...,  97,  96, 105],
       [ 77,  77,  77, ...,  99,  97, 108],
       [ 77,  77,  76, ..., 101,  99, 111]], dtype=uint8), array([[24, 20, 15, ...,
8,  8, 11],
       [24, 20, 15, ...,  8,  8, 10],
       [24, 20, 15, ...,  8,  8,  9],
       ...,
       [24, 29, 34, ..., 12, 12, 13],
       [25, 29, 34, ..., 12, 12, 13],
       [24, 29, 34, ..., 12, 12, 13]], dtype=uint8), array([[ 5,  6,  7, ..., 14, 1
5, 17],
       [ 5,  6,  7, ..., 14, 15, 17],
       [ 6,  6,  6, ..., 13, 16, 18],
       ...,
       [30, 30, 30, ..., 10, 11, 13],
       [27, 28, 28, ..., 10, 11, 13],
       [27, 28, 28, ..., 10, 13, 15]], dtype=uint8), array([[ 4,  4,  4, ...,  9,
8,  6],
       [ 4,  4,  4, ...,  7,  7,  5],
       [ 4,  4,  4, ...,  5,  5,  4],
       ...,
       [49, 35, 27, ..., 19, 12,  9],
       [54, 36, 22, ..., 26, 17, 10],
       [47, 37, 21, ..., 33, 23, 13]], dtype=uint8)]
```

In [5]:
```
print(labels)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Question: Run the function above on your generated dataset. Provide an analysis and note all the challenges you have encountered running this code.

Performing Face Recognition Algorithms

Here is a sample script for testing the Face Recognition Algorithm. In this section, we're going to follow the same process but with different algorithms for face recognitions, namely:

- Eigenface Recognition
- Fisherface Recognition
- Local Binary Pattern Histograms (LBPH) Recognition

In [6]:
```
pip install --user opencv-contrib-python==4.5.5.64
```

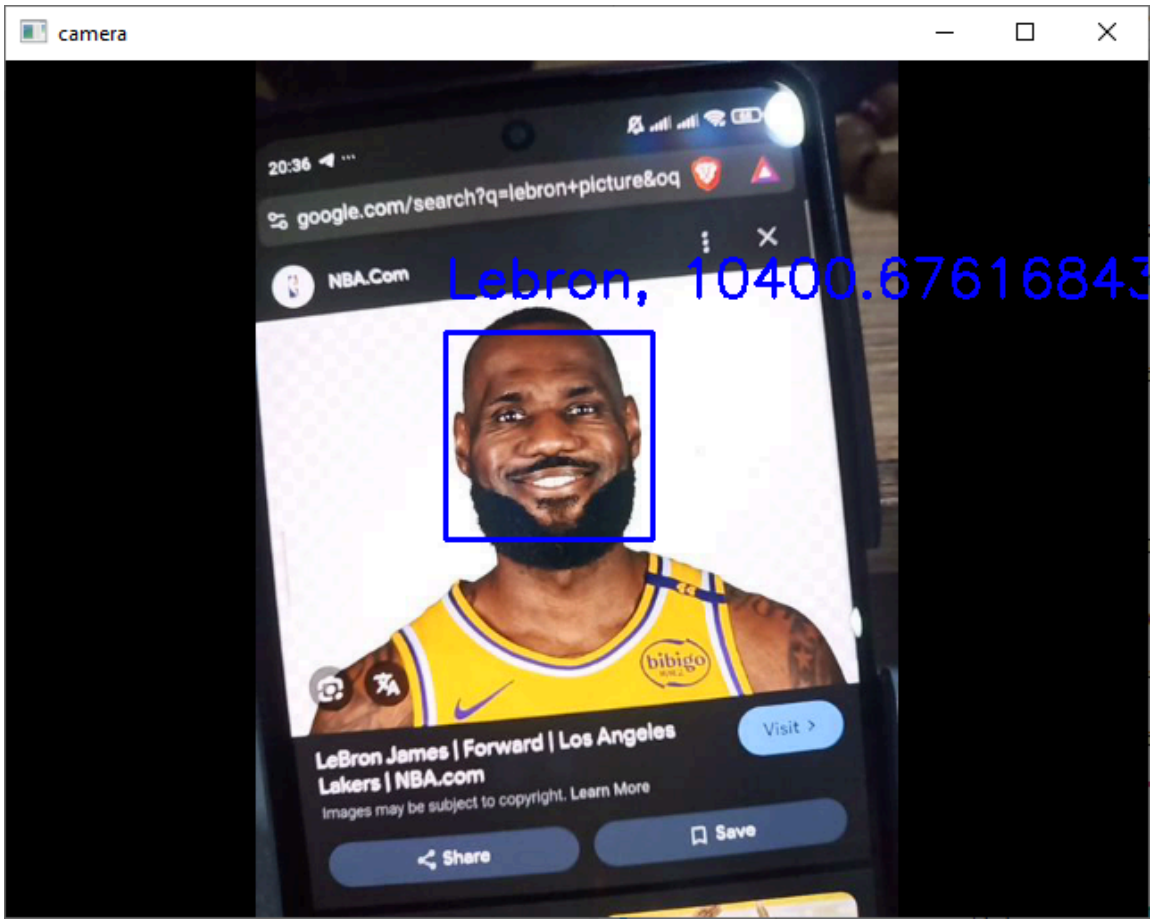
```
Collecting opencv-contrib-python==4.5.5.64
  Using cached opencv_contrib_python-4.5.5.64-cp36-abi3-win_amd64.whl.metadata (18 k
B)
Requirement already satisfied: numpy>=1.21.2 in e:\anacondanavigator\lib\site-packag
es (from opencv-contrib-python==4.5.5.64) (1.26.4)
Using cached opencv_contrib_python-4.5.5.64-cp36-abi3-win_amd64.whl (42.2 MB)
Installing collected packages: opencv-contrib-python
  Attempting uninstall: opencv-contrib-python
    Found existing installation: opencv-contrib-python 4.11.0.86
    Uninstalling opencv-contrib-python-4.11.0.86:
      Successfully uninstalled opencv-contrib-python-4.11.0.86
Successfully installed opencv-contrib-python-4.5.5.64
Note: you may need to restart the kernel to use updated packages.
  WARNING: Failed to remove contents in a temporary directory 'C:\Users\Ery\AppData
\Roaming\Python\Python311\site-packages\~v2'.
  You can safely remove it manually.
```

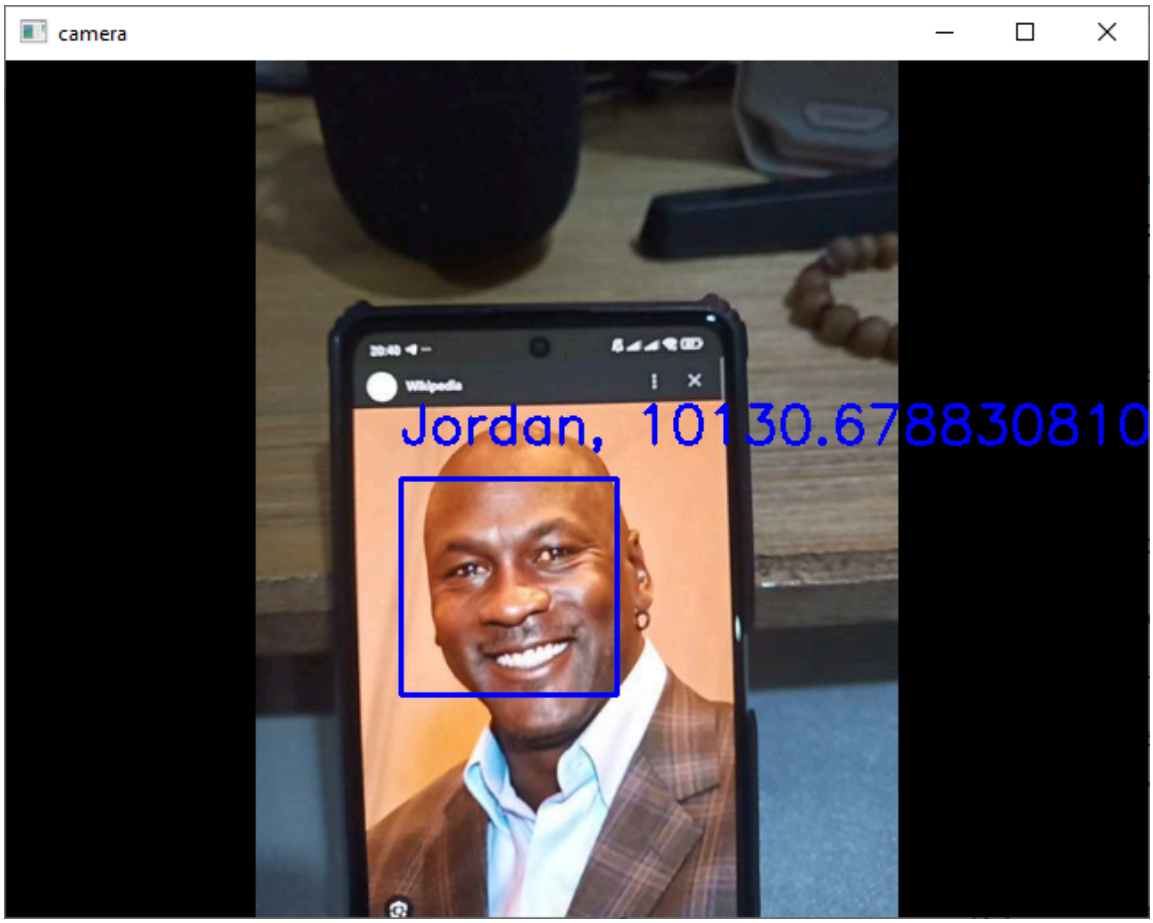
In [1]:
```python
import cv2
print(cv2.__version__)
```

```
4.5.5
```

In [5]:
```python
def face_rec():
  names = ['Lebron', 'Jordan'] # Put your names here for faces to recognize
  [X, y] = read_images("Act 7",1)
  y = np.asarray(y, dtype=np.int32)

  model = cv2.face.EigenFaceRecognizer_create()
  model.train(X, y)

  camera = cv2.VideoCapture(0)
  face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

  while True:
    ret, img = camera.read()
    if not ret:
      break

    faces = face_cascade.detectMultiScale(img, 1.3, 5)

    for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
      gray = cv2.cvtColor(img[y:y + h, x:x + w], cv2.COLOR_BGR2GRAY)
      roi = cv2.resize(gray, (200, 200), interpolation=cv2.INTER_LINEAR)

      try:
        params = model.predict(roi)
        label = names[params[0]]
        cv2.putText(img, label + ", " + str(params[1]), (x, y - 20), cv2.FONT_HERSH
      except:
        continue

    cv2.imshow("camera", img)
    if cv2.waitKey(1) & 0xFF == ord("q"):
      break
```

```
  camera.release()
  cv2.destroyAllWindows()

if __name__ == "__main__":
  face_rec()
```





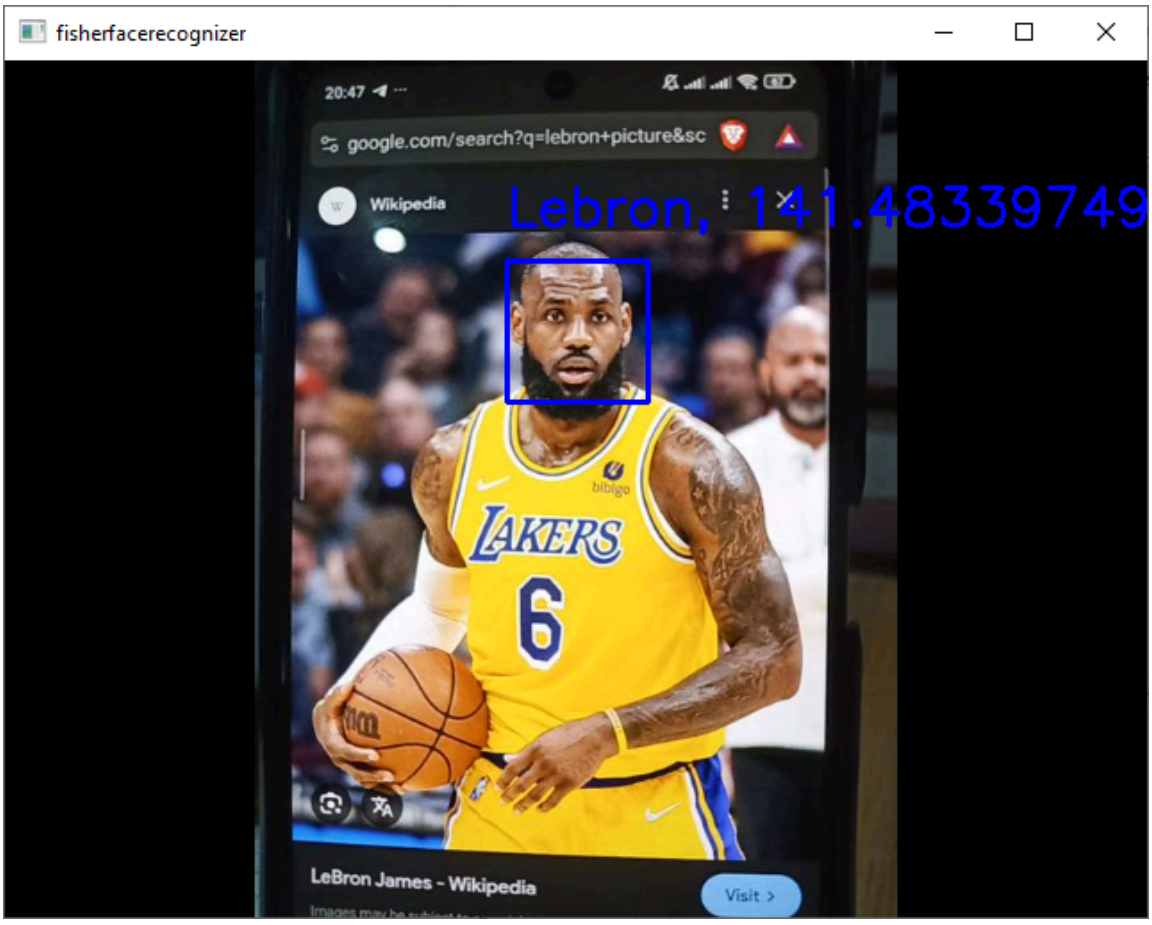
Question: Provide an analysis of the sample script for the process using the Eigenface Model. What is the sample code doing? Are you able to troubleshoot any problems encountered?

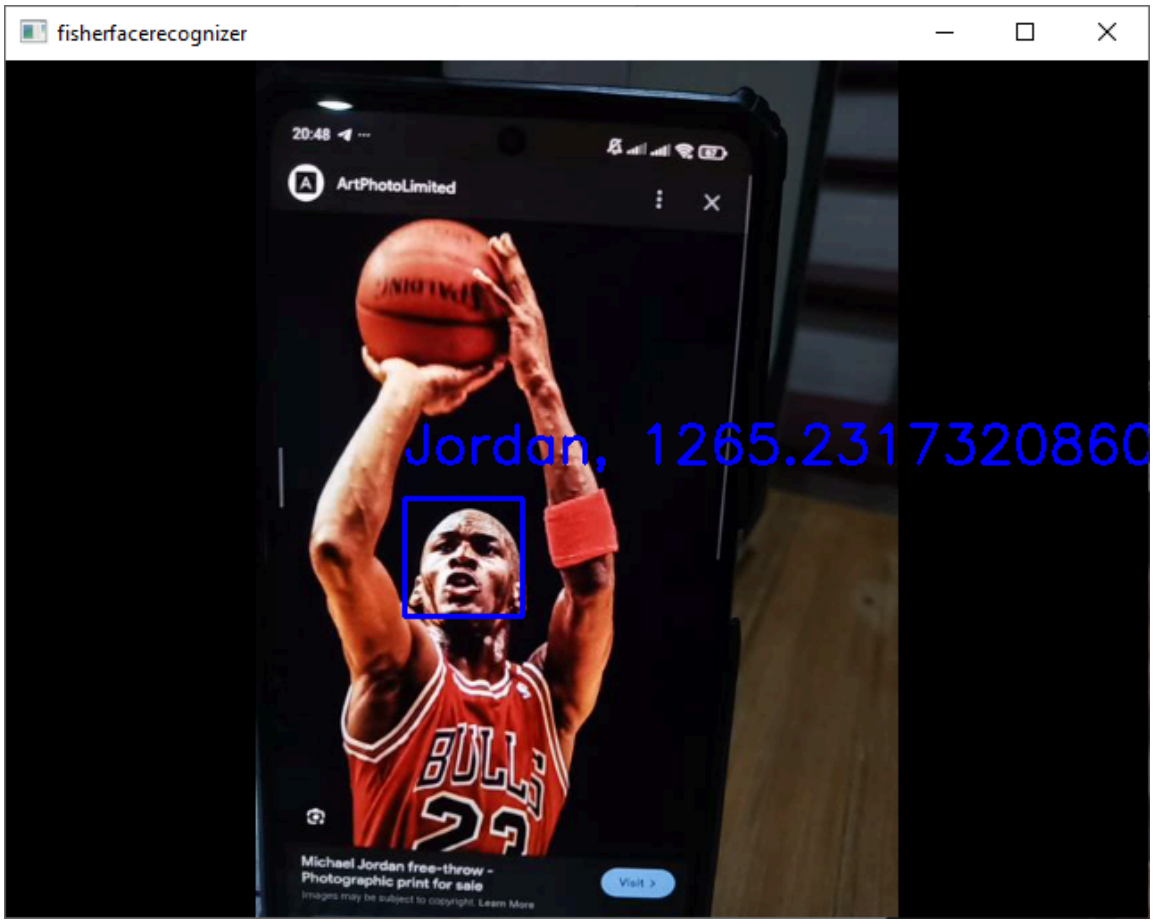
Perform the remaining face recognition techniques by using the same (or modified) process from the sample code:

- `model = cv2.face.createFisherFaceRecognizer()`
- `model = cv2.face.createLBPHFaceRecognizer()`

In [8]:
```
# fisher Face recognizer
def face_rec():
  names = ['Lebron', 'Jordan'] # Put your names here for faces to recognize
  [X, y] = read_images("Act 7",1)
  y = np.asarray(y, dtype=np.int32)

  model = cv2.face.FisherFaceRecognizer_create()
  model.train(X, y)

  camera = cv2.VideoCapture(0)
  face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

  while True:
```

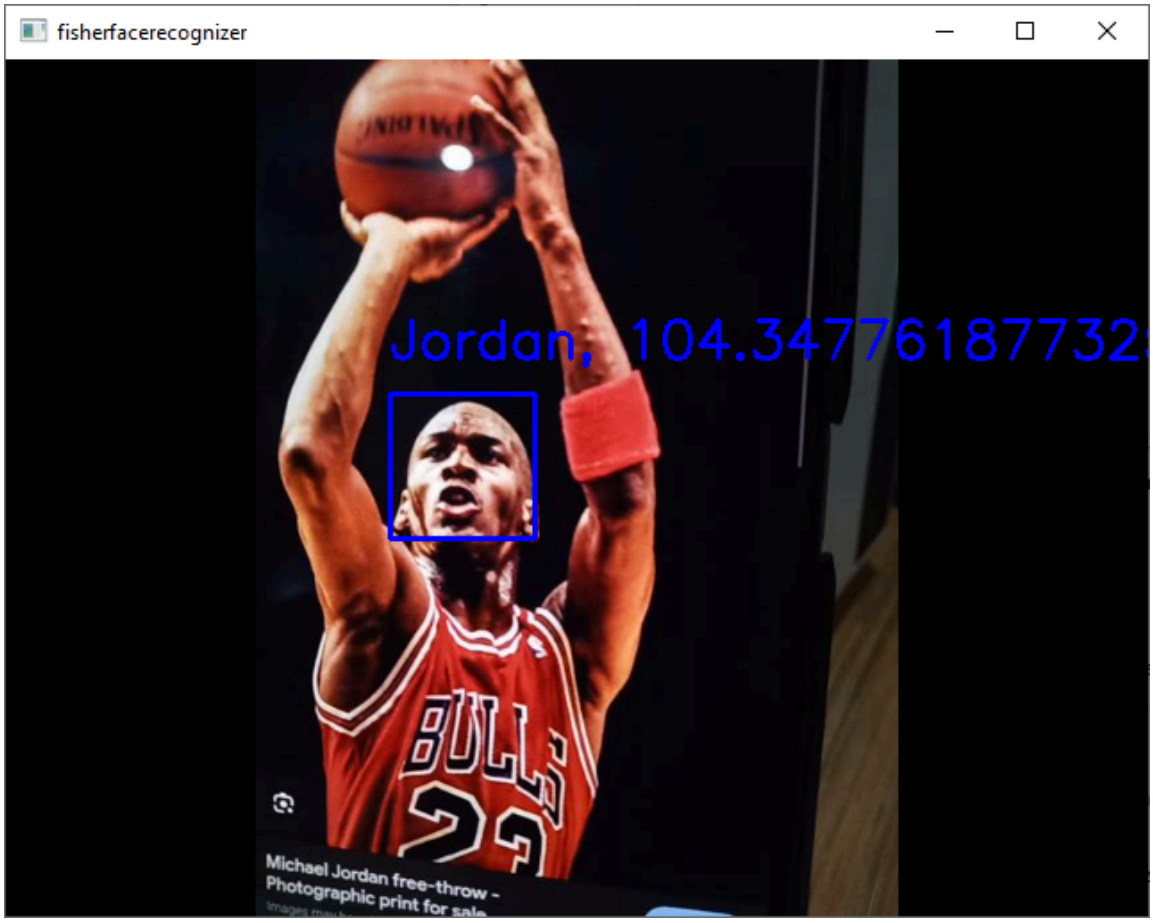
```
    ret, img = camera.read()
    if not ret:
      break

    faces = face_cascade.detectMultiScale(img, 1.3, 5)

    for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
      gray = cv2.cvtColor(img[y:y + h, x:x + w], cv2.COLOR_BGR2GRAY)
      roi = cv2.resize(gray, (200, 200), interpolation=cv2.INTER_LINEAR)

      try:
        params = model.predict(roi)
        label = names[params[0]]
        cv2.putText(img, label + ", " + str(params[1]), (x, y - 20), cv2.FONT_HERSH
      except:
        continue

    cv2.imshow("fisherfacerecognizer", img)
    if cv2.waitKey(1) & 0xFF == ord("q"):
      break

  camera.release()
  cv2.destroyAllWindows()

if __name__ == "__main__":
  face_rec()
```





In [11]:
```
# LBPH Face Recognizer
def face_rec():
  names = ['Lebron', 'Jordan'] # Put your names here for faces to recognize
  [X, y] = read_images("Act 7",1)
  y = np.asarray(y, dtype=np.int32)

  model = cv2.face.LBPHFaceRecognizer_create()
  model.train(X, y)

  camera = cv2.VideoCapture(0)
  face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

  while True:
    ret, img = camera.read()
    if not ret:
      break

    faces = face_cascade.detectMultiScale(img, 1.3, 5)

    for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
      gray = cv2.cvtColor(img[y:y + h, x:x + w], cv2.COLOR_BGR2GRAY)
      roi = cv2.resize(gray, (200, 200), interpolation=cv2.INTER_LINEAR)

      try:
        params = model.predict(roi)
        label = names[params[0]]
```

```
        cv2.putText(img, label + ", " + str(params[1]), (x, y - 20), cv2.FONT_HERSH
      except:
        continue

    cv2.imshow("LBPHrecognizer", img)
    if cv2.waitKey(1) & 0xFF == ord("q"):
      break

  camera.release()
  cv2.destroyAllWindows()

if __name__ == "__main__":
  face_rec()
```



Question: The `predict()` method returns a two-element array. Provide your analysis of the two returned values and their important ince this application.

4. Supplementary Activity

Your accomplisment of the tasks below contribute to the achievement of ILO1, ILO2, and ILO3 for this module.

Tasks:

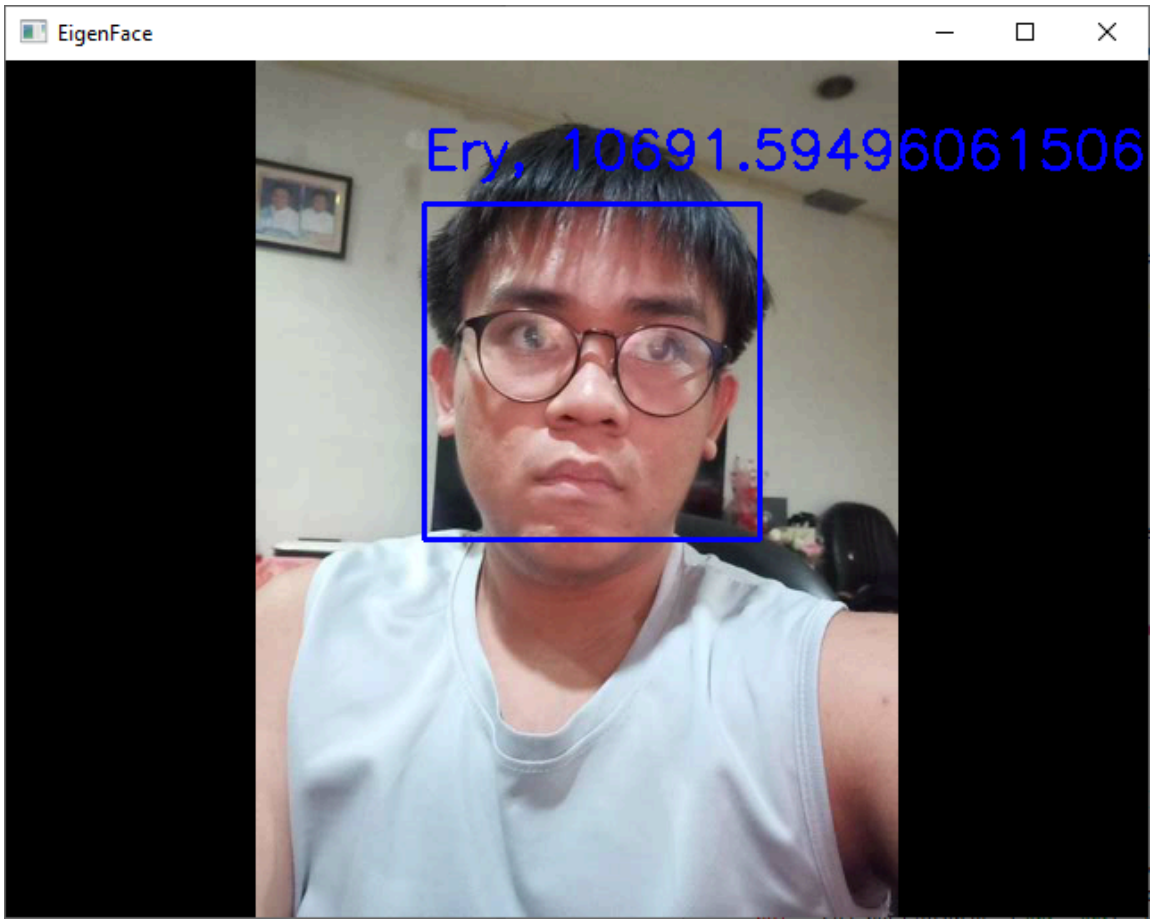
1. Create a new dataset for testing, this dataset must include the following:

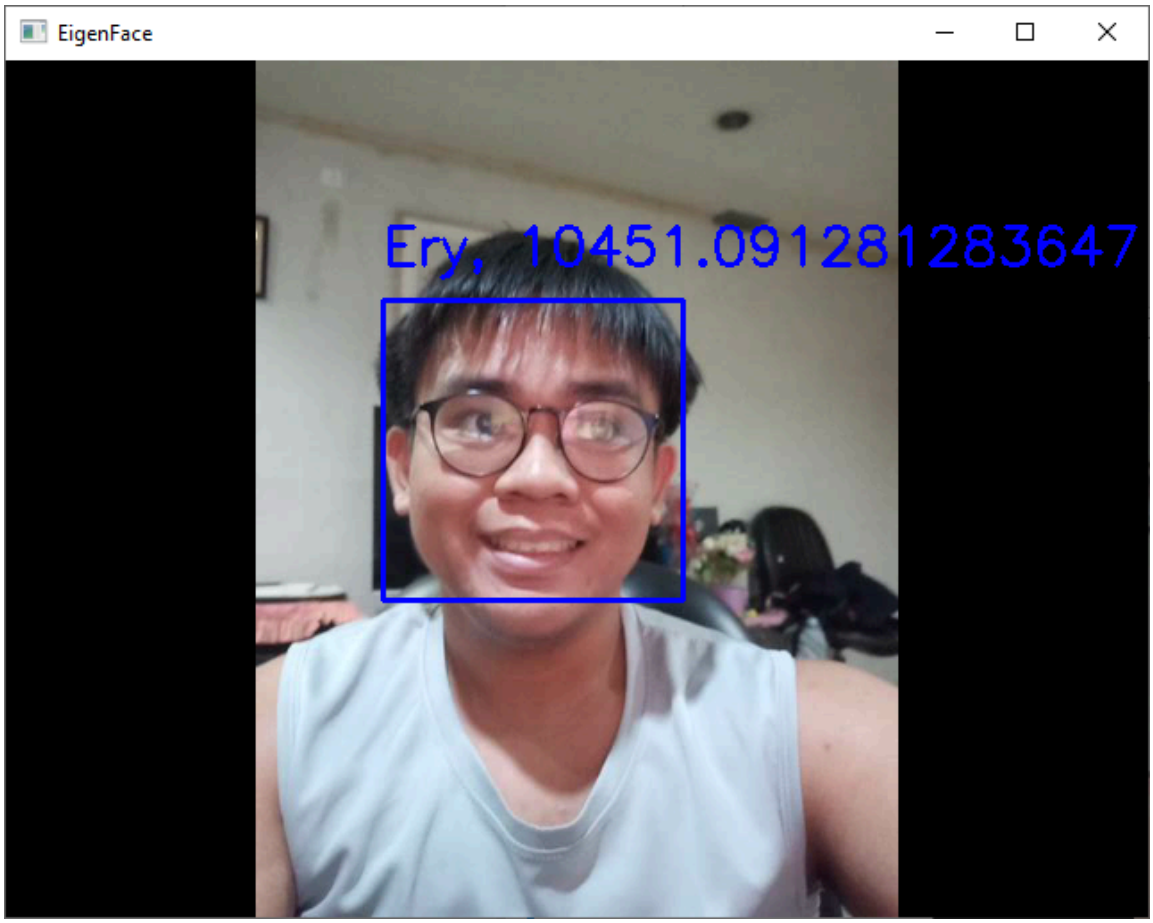
- The same person/s that the model has to recognize.
- Different person/s that the model should not recognize.

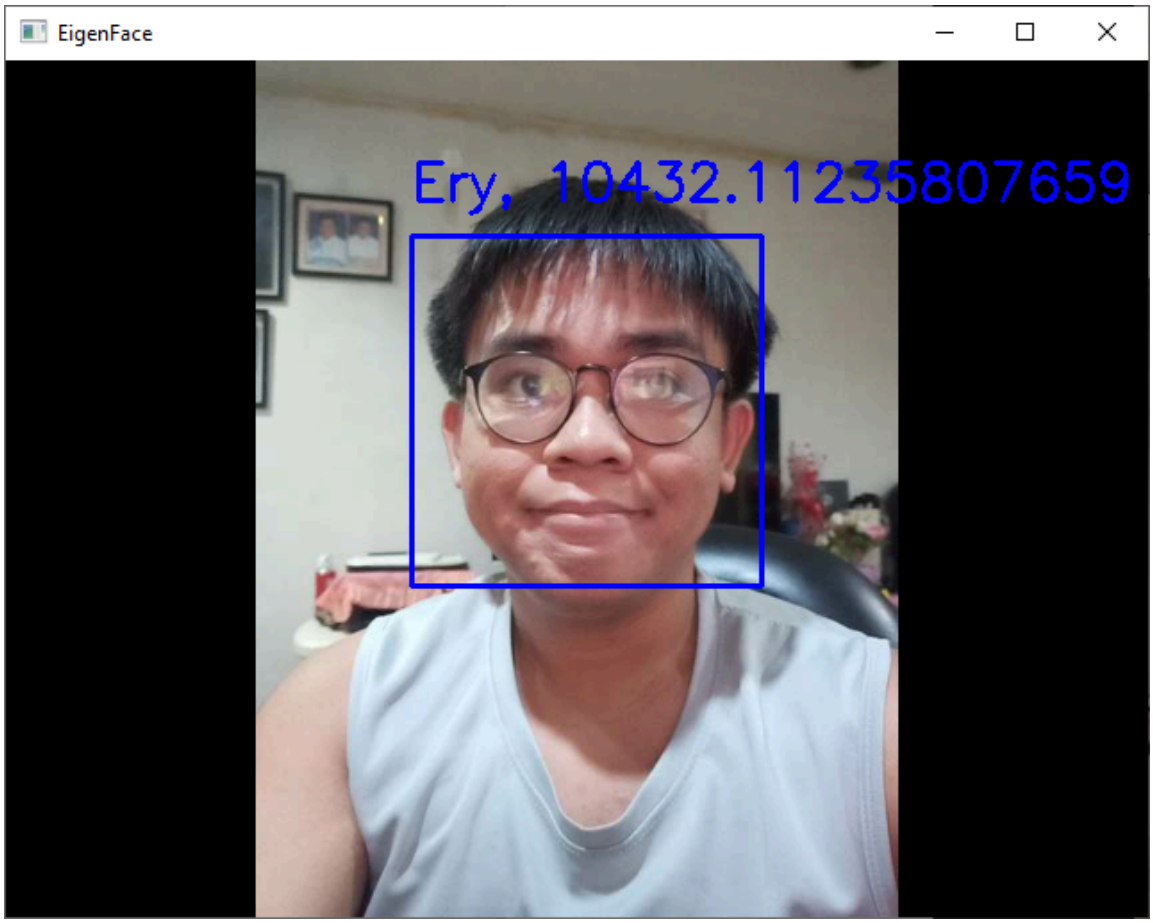
2. For each model, perform 20 tests. Document the testing performed and provide observations.
3. Conclude on the performed tests by providing your evaluation of the performance of the models.

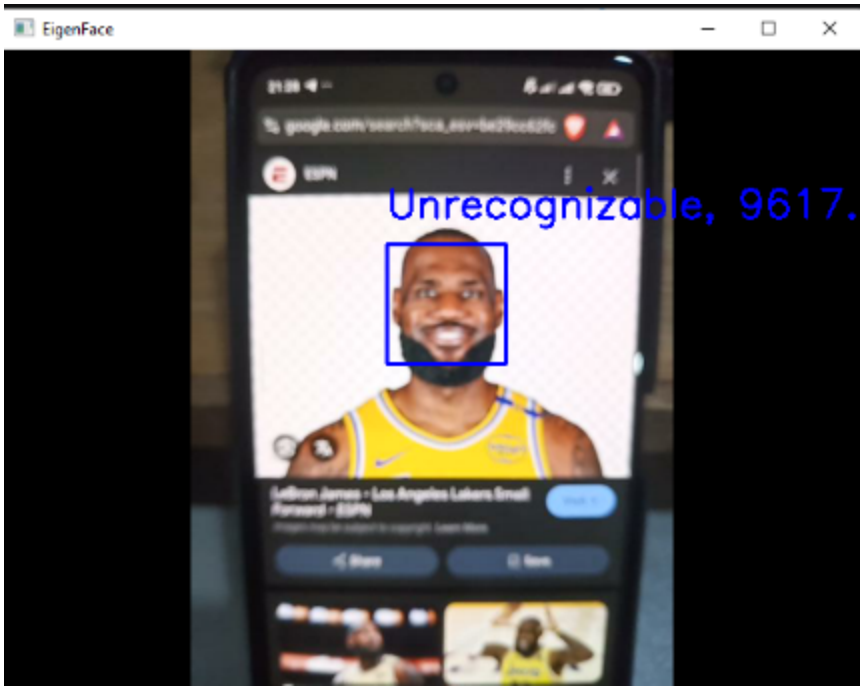
In [24]:
```
def face_rec():
  names = ['Ery','Unrecognizable'] # Put your names here for faces to recognize
  [X, y] = read_images("Act7_suppl",1)
  print(y)
  y = np.asarray(y, dtype=np.int32)

  model = cv2.face.EigenFaceRecognizer_create()
  model.train(X, y)

  camera = cv2.VideoCapture(0)
  face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

  while True:
    ret, img = camera.read()
    if not ret:
      break

    faces = face_cascade.detectMultiScale(img, 1.3, 5)

    for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
      gray = cv2.cvtColor(img[y:y + h, x:x + w], cv2.COLOR_BGR2GRAY)
      roi = cv2.resize(gray, (200, 200), interpolation=cv2.INTER_LINEAR)

      try:
        params = model.predict(roi)
        label = names[params[0]]
        cv2.putText(img, label + ", " + str(params[1]), (x, y - 20), cv2.FONT_HERSH
      except:
        continue

    cv2.imshow("EigenFace", img)
    if cv2.waitKey(1) & 0xFF == ord("q"):
      break

  camera.release()
  cv2.destroyAllWindows()

if __name__ == "__main__":
  face_rec()
```

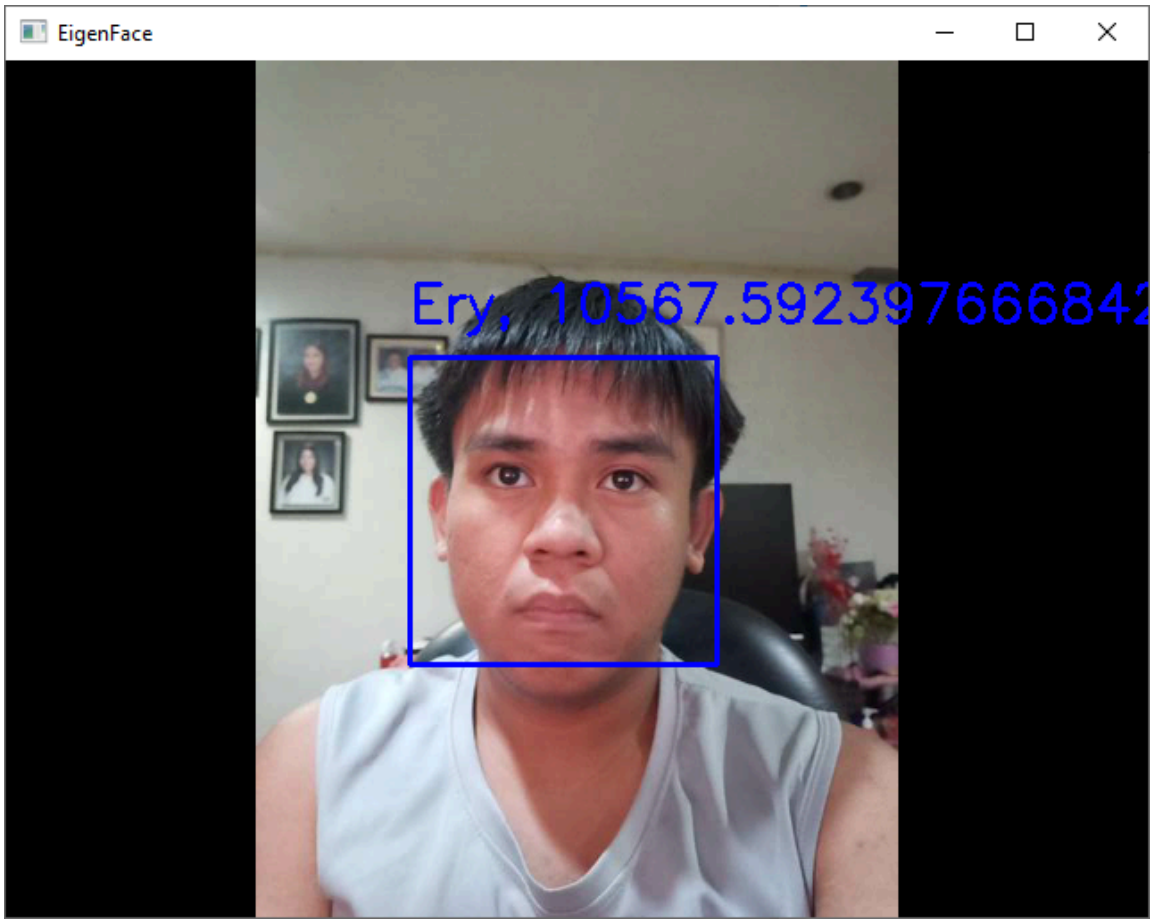
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

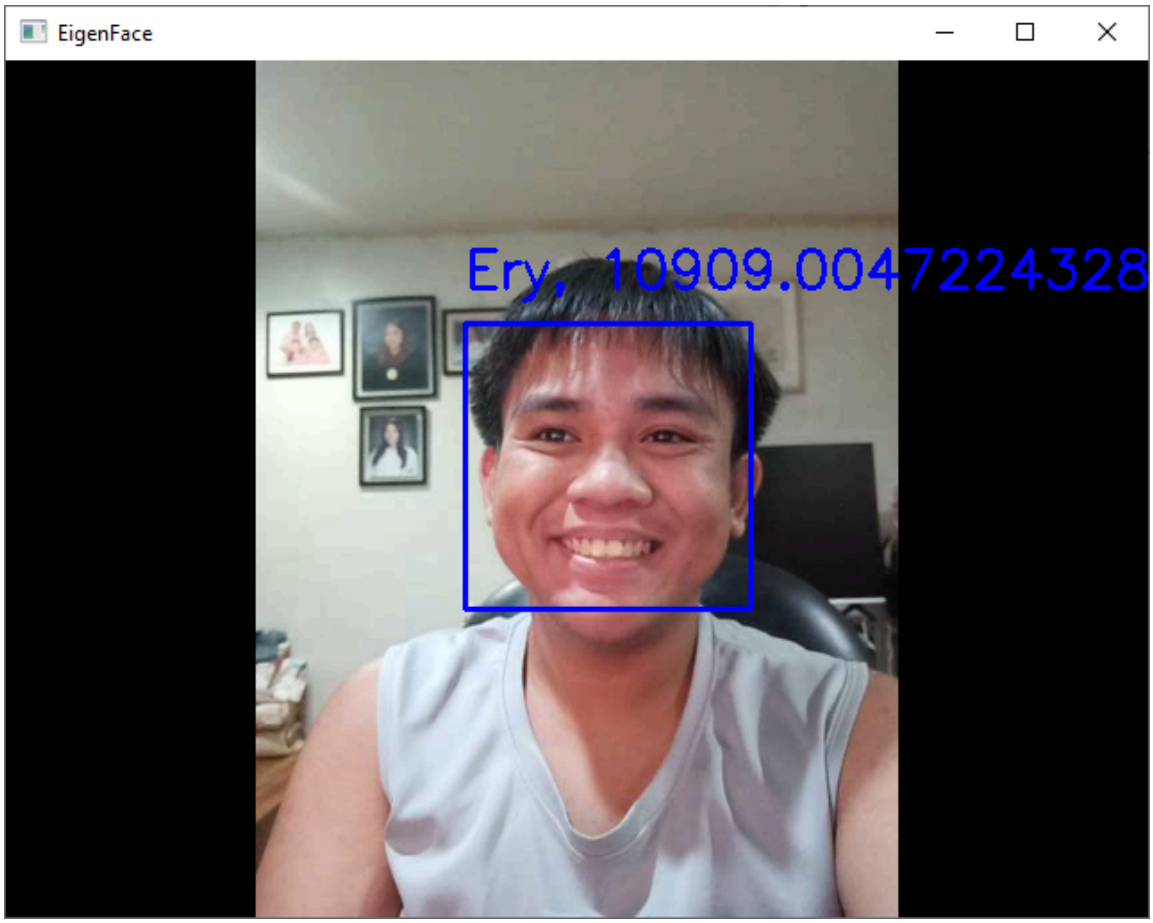

EigenFace
Ery, 10691.59496061506

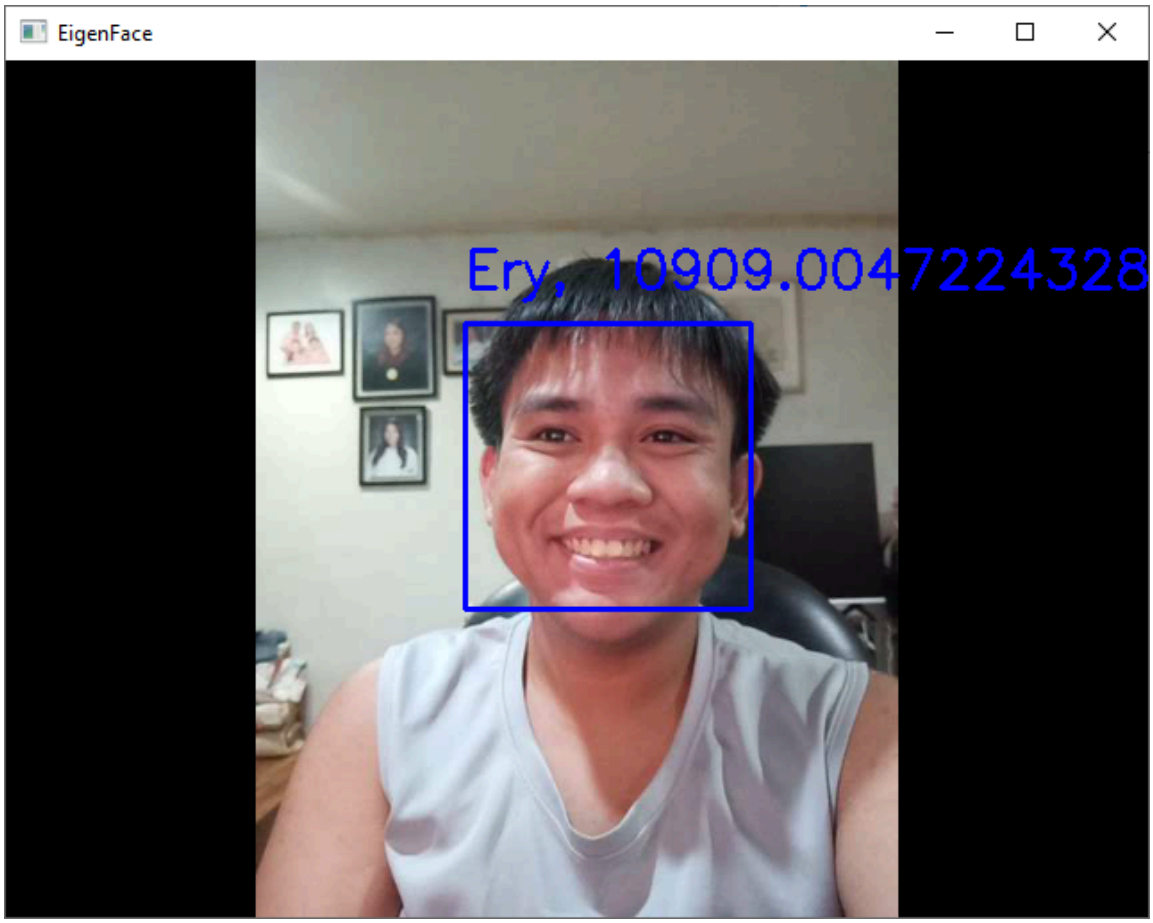

EigenFace
Ery, 10451.091281283647

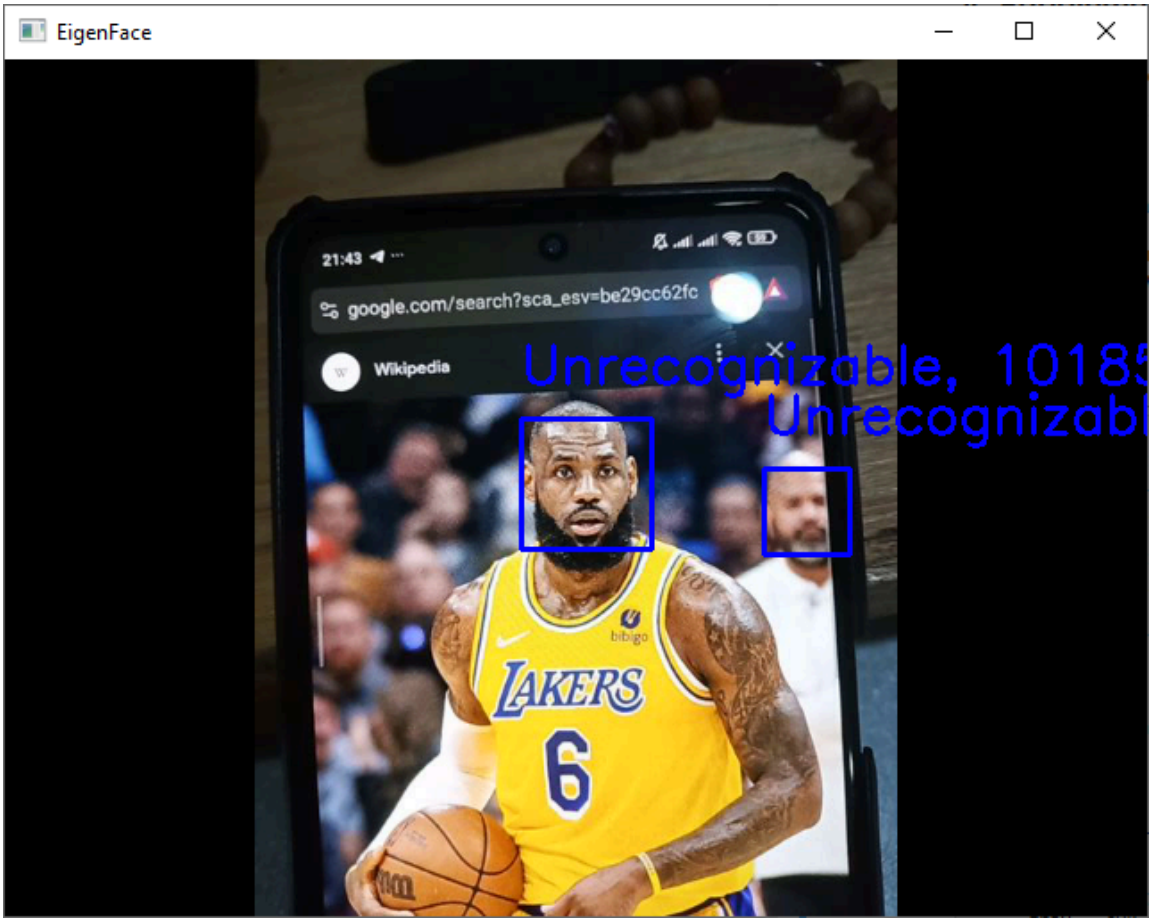

EigenFace
Ery, 10432.11235807659

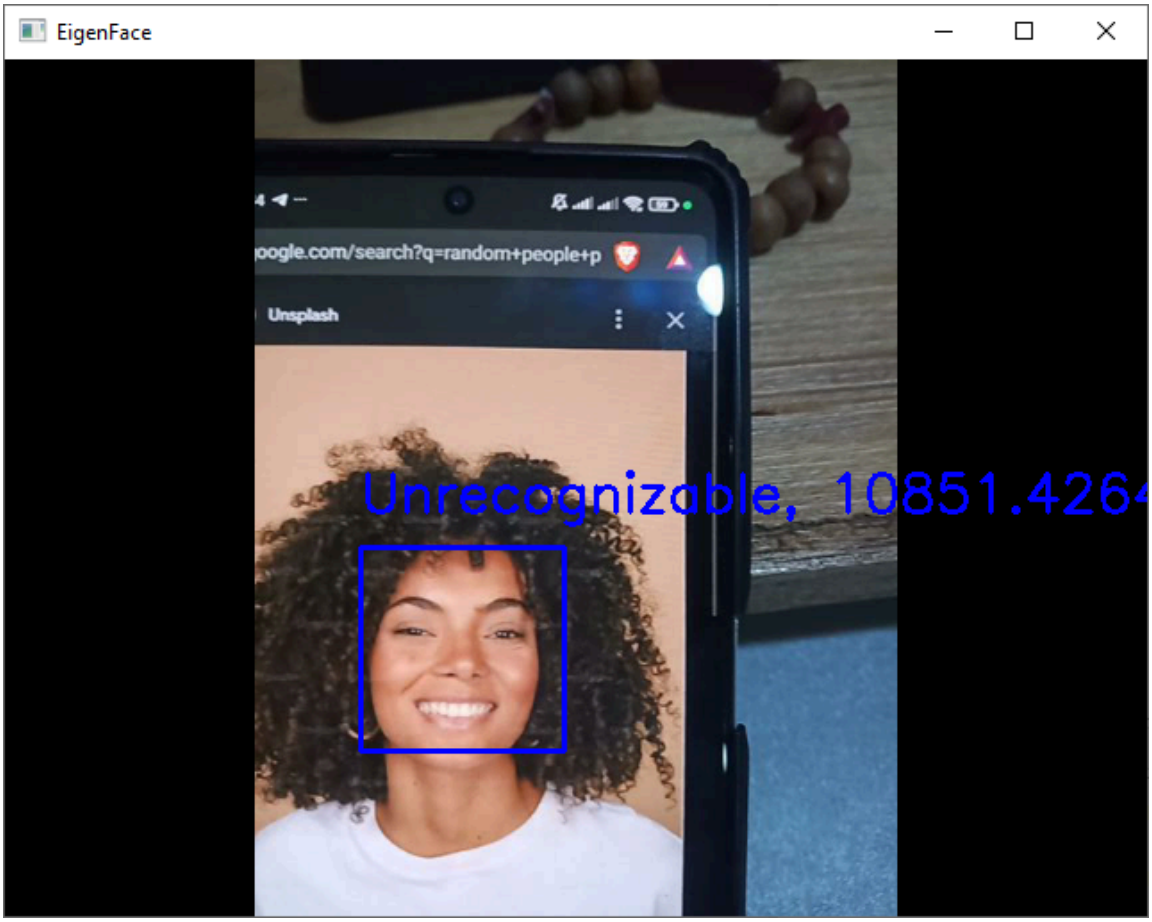

EigenFace
Unrecognizable, 9617.

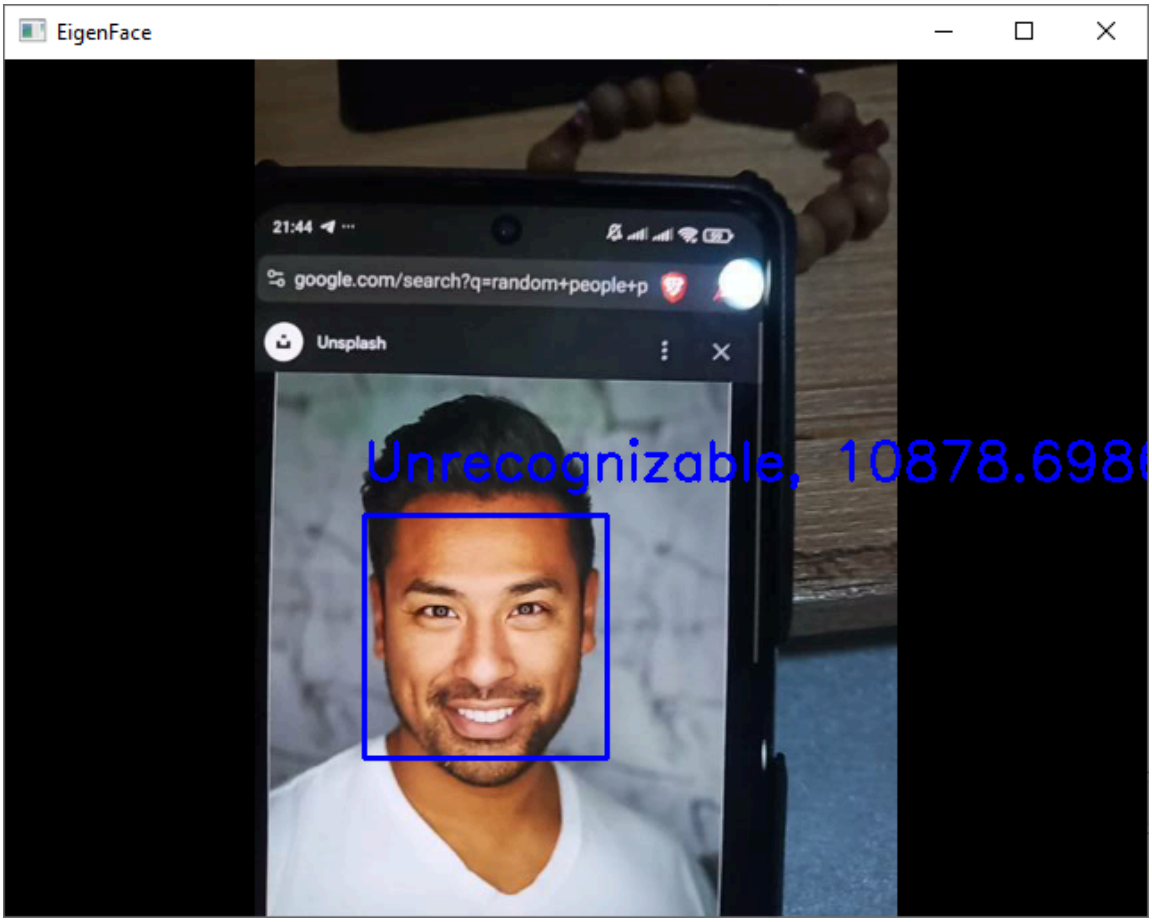

EigenFace
Ery, 10567.592397666842

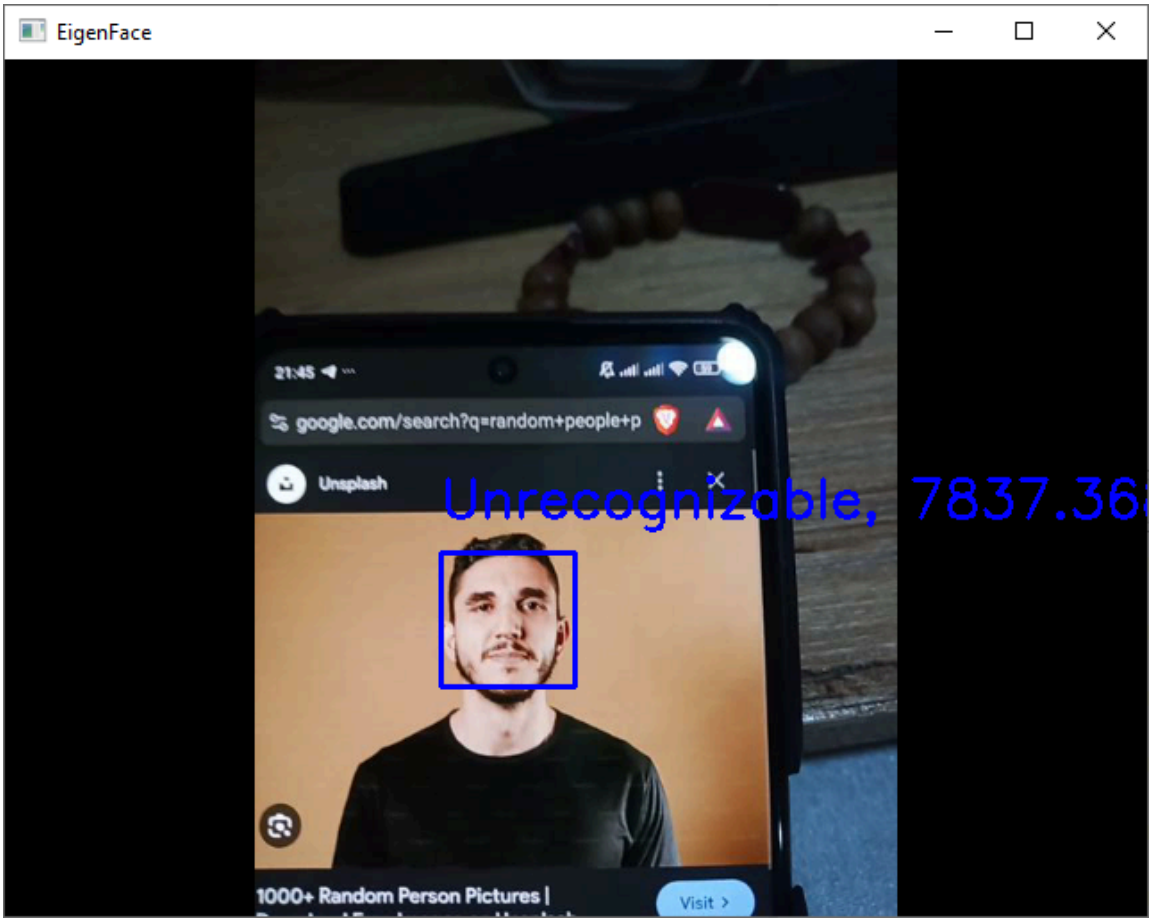

EigenFace
Ery, 10909.0047224328

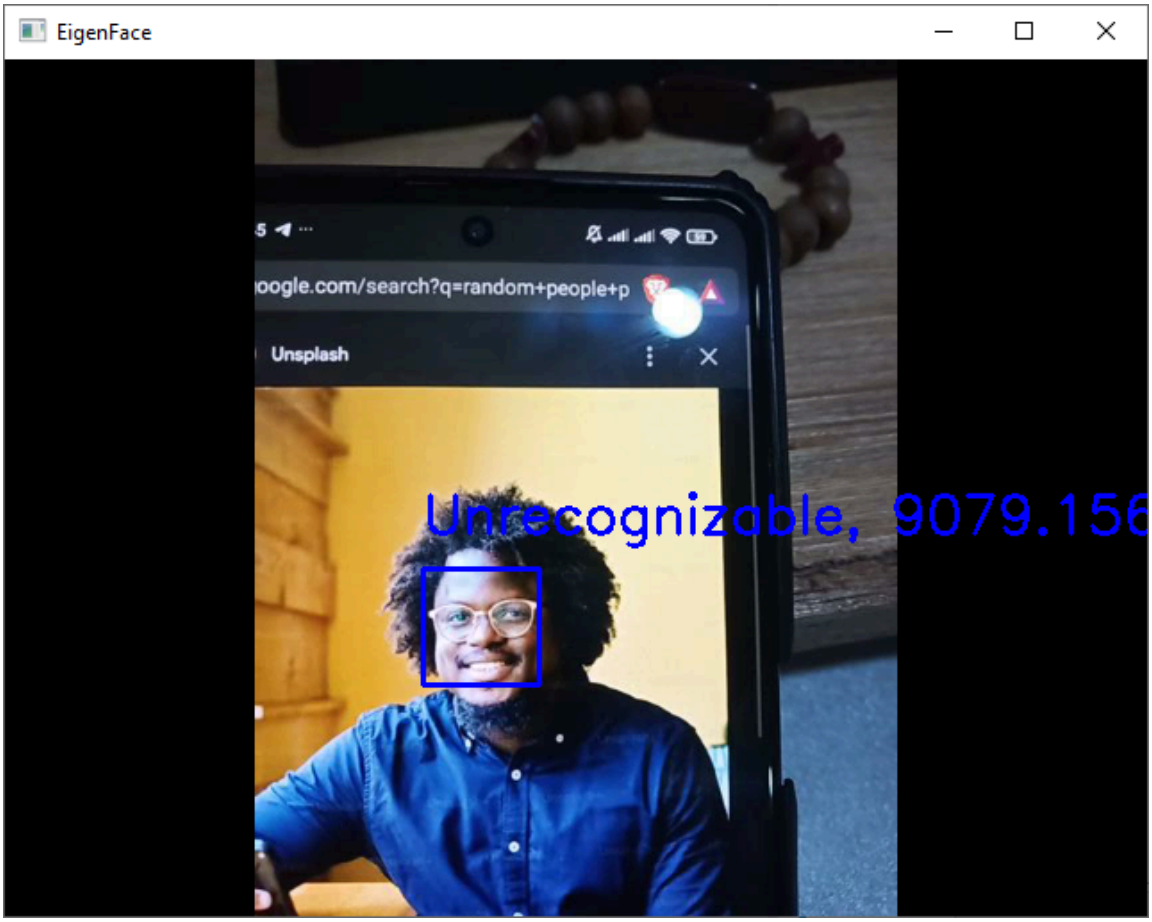

EigenFace
Ery, 10909.0047224328

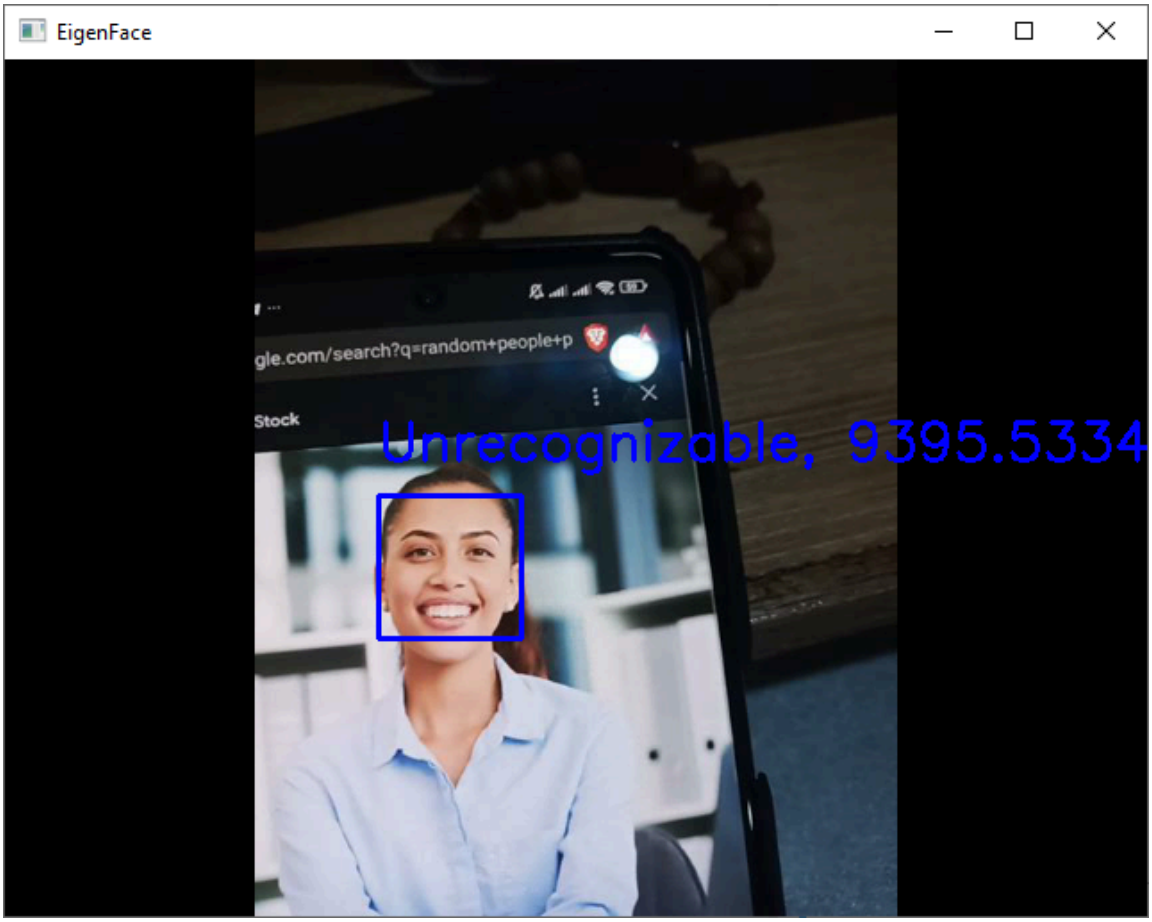

EigenFace
21:43
google.com/search?sca_esv=be29cc62fc
Wikipedia
Unrecognizable, 10185
Unrecognizabl
LAKERS
6

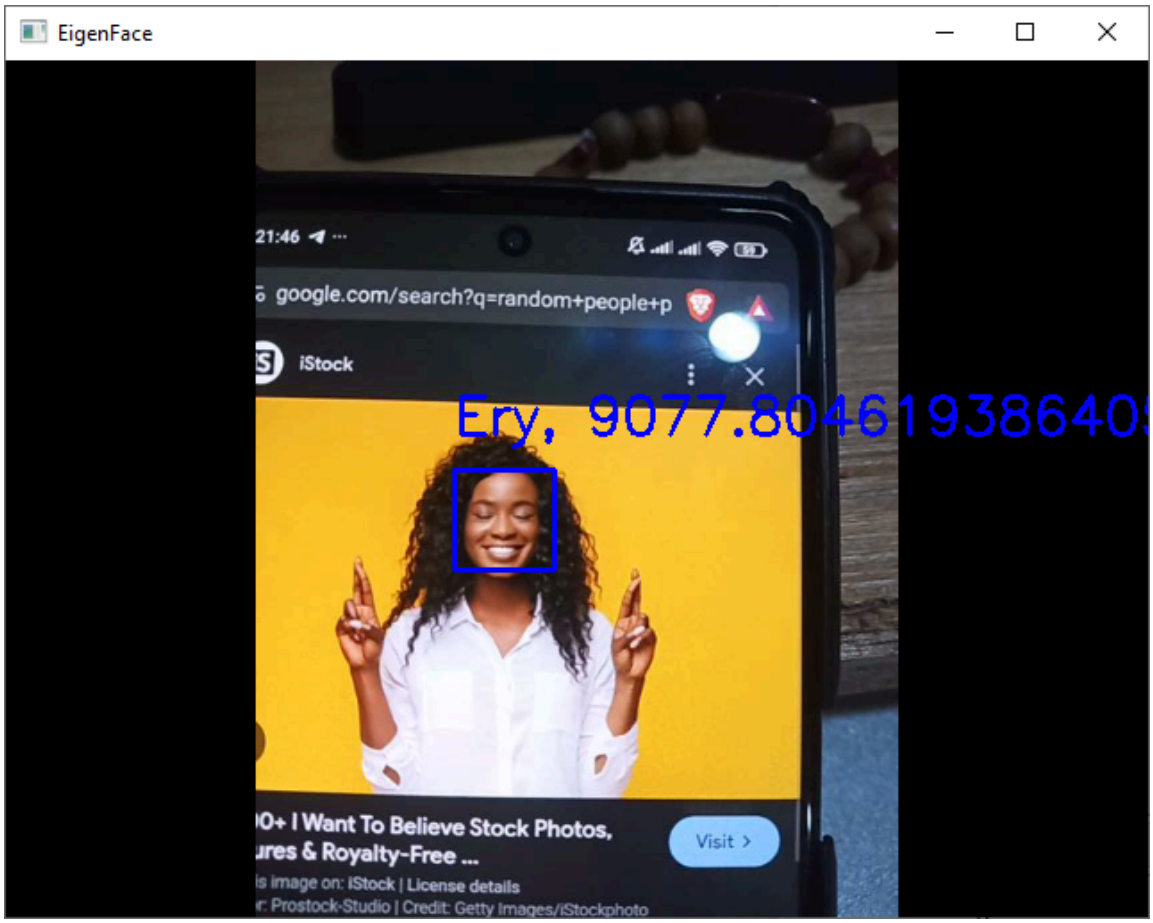

EigenFace
oogle.com/search?q=random+people+p
Unsplash
Unrecognizable, 10851.4264

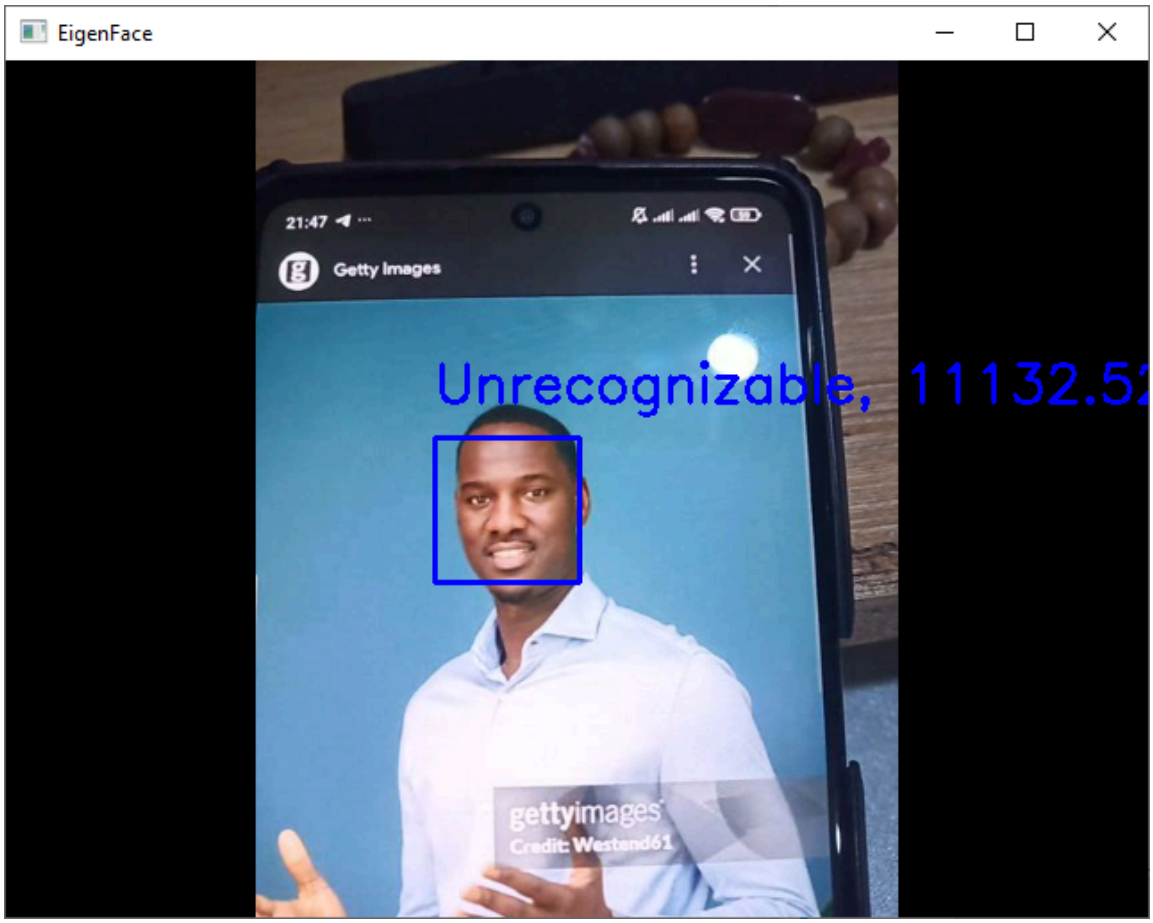

EigenFace
21:44
google.com/search?q=random+people+p
Unsplash
Unrecognizable, 10878.698

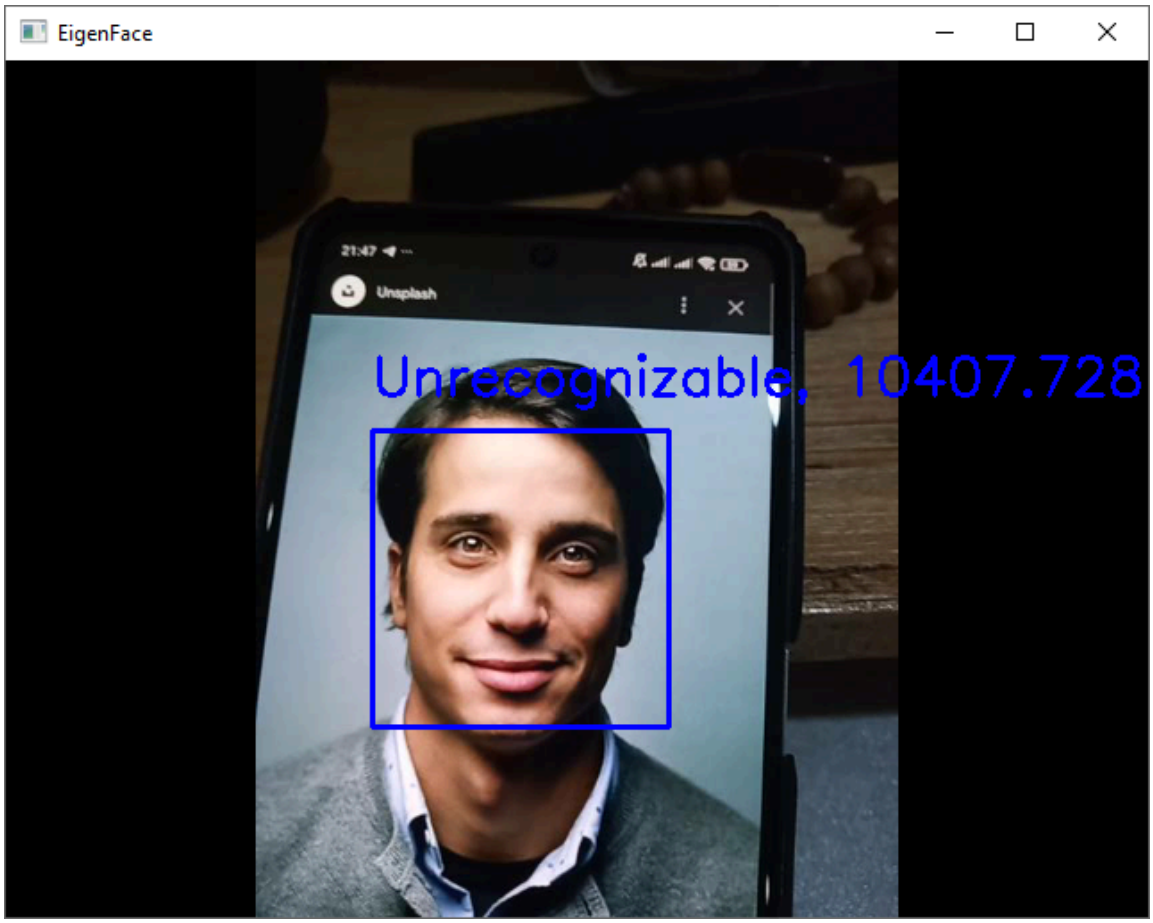

EigenFace
21:45
google.com/search?q=random+people+p
Unsplash
Unrecognizable, 7837.36
1000+ Random Person Pictures |
Visit ›

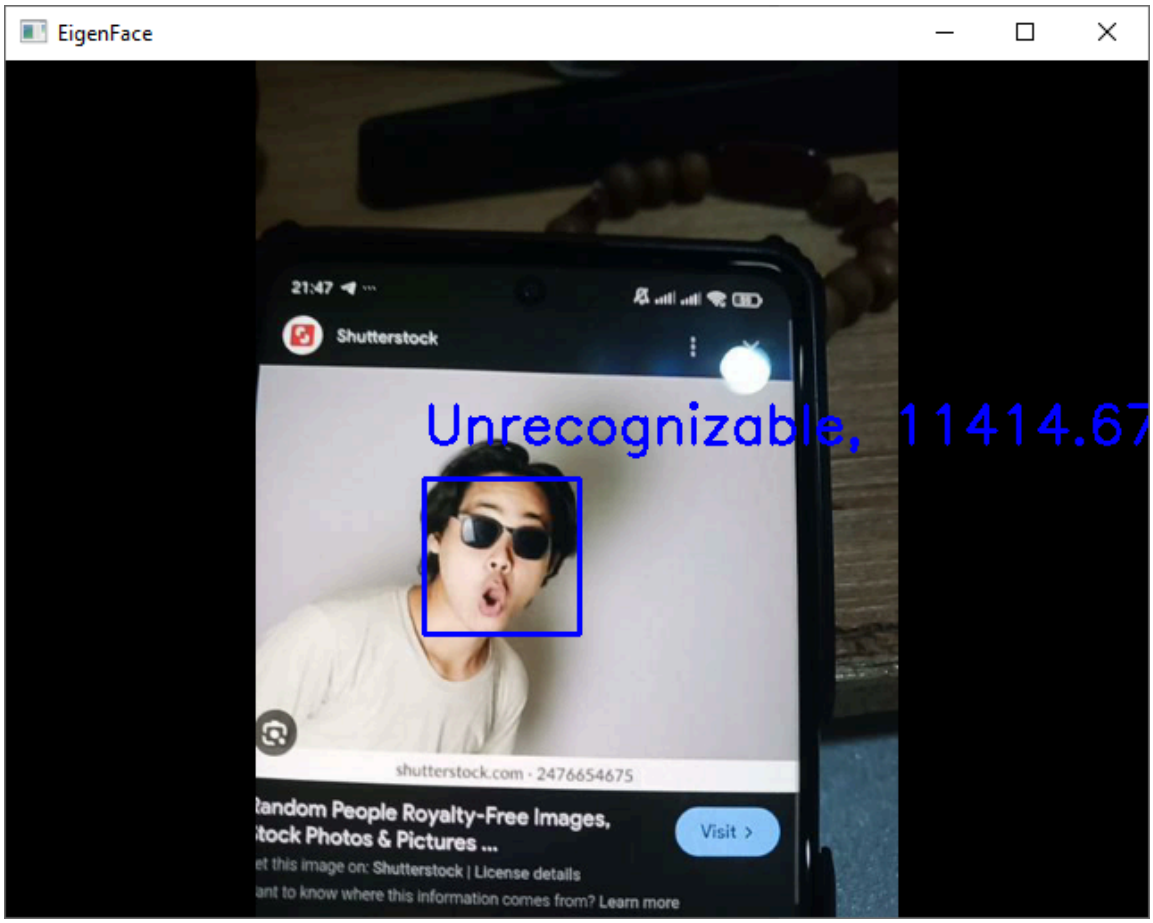

EigenFace
oogle.com/search?q=random+people+p
Unsplash
Unrecognizable, 9079.156

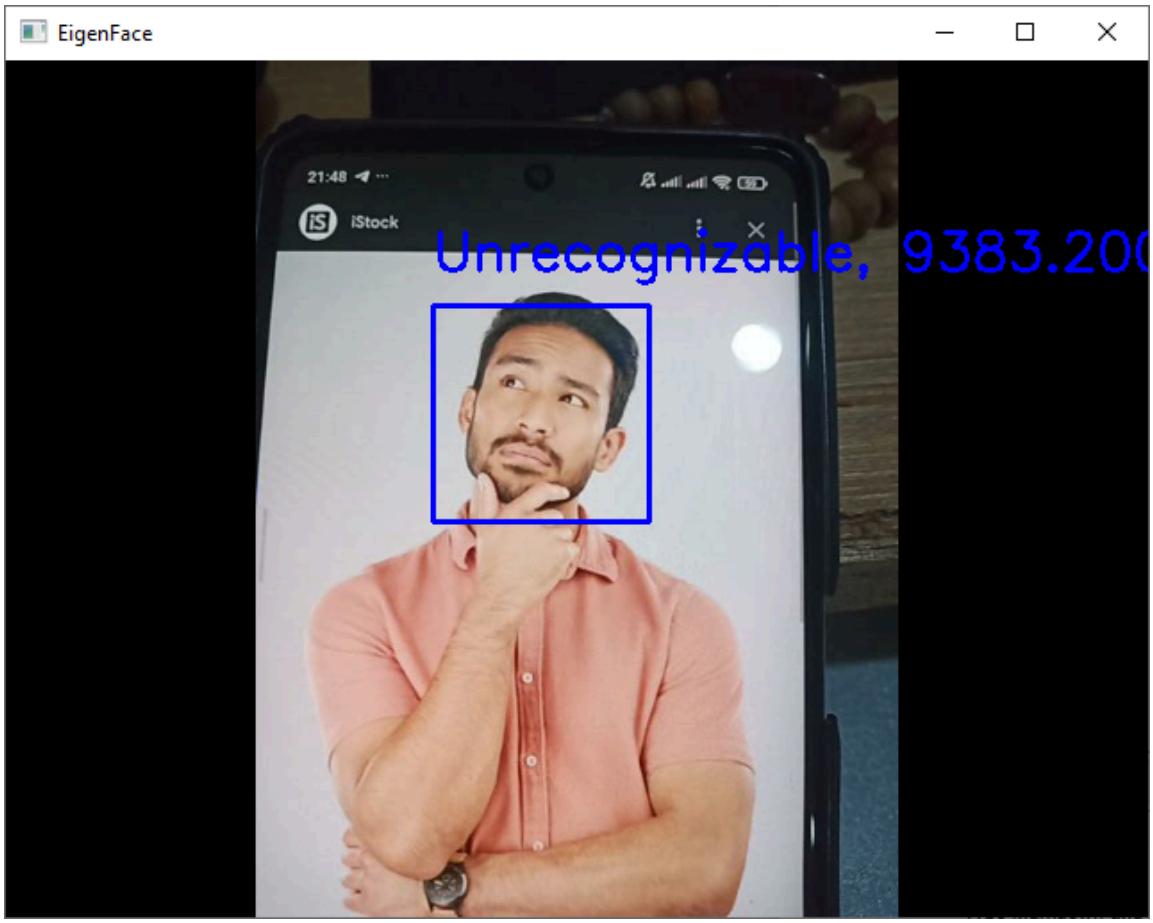

EigenFace
gle.com/search?q=random+people+p
Stock
Unrecognizable, 9395.5334

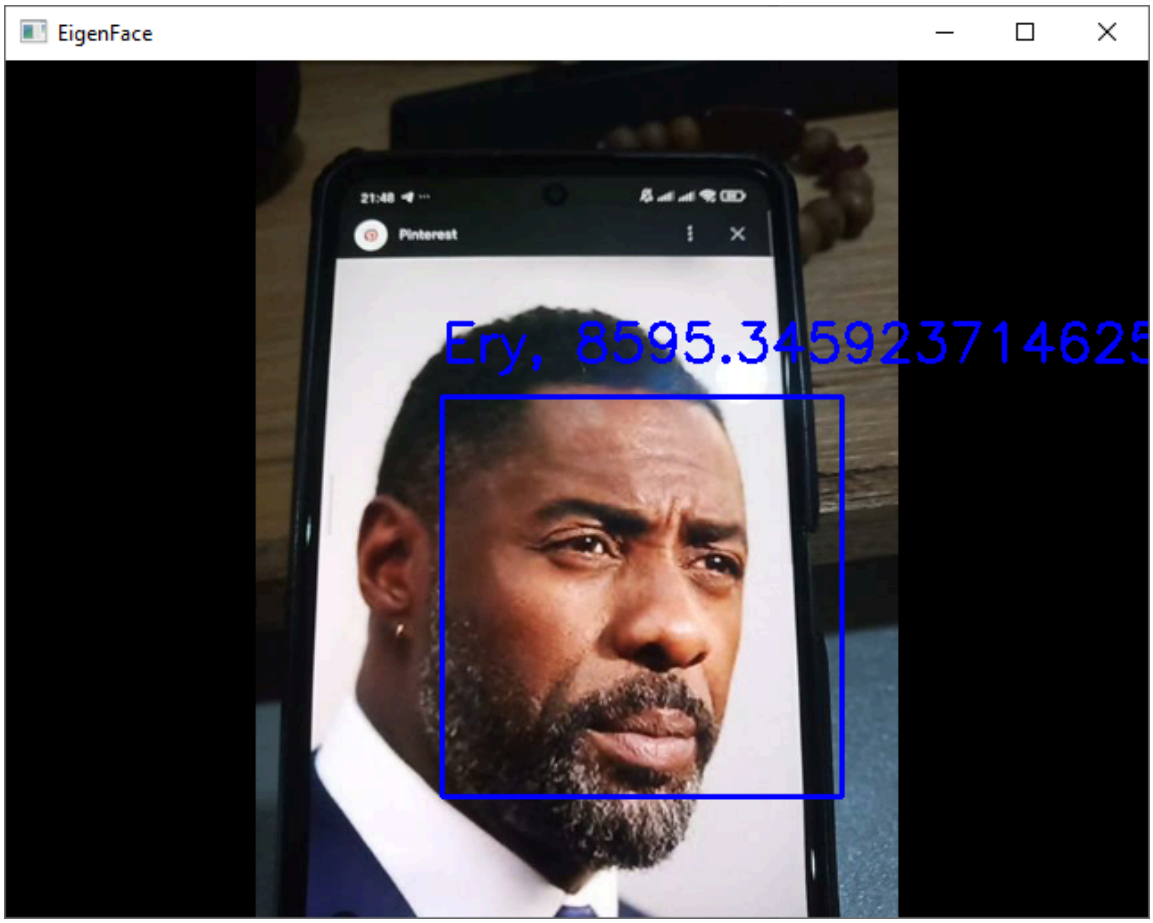

EigenFace
21:46
google.com/search?q=random+people+p
iStock
Ery, 9077.804619386405
Visit ›
Prostock-Studio | Credit: Getty Images/iStockphoto

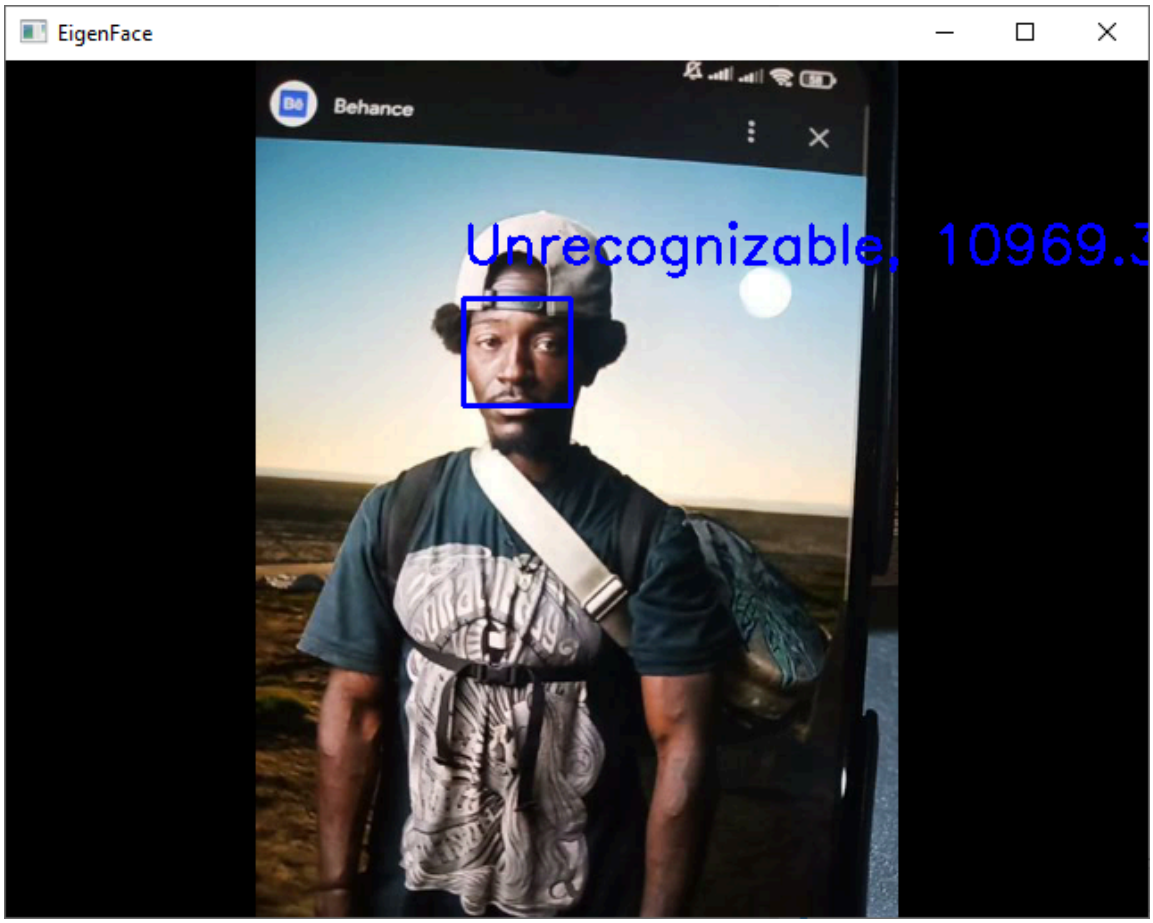

EigenFace
21:47
Getty Images
Unrecognizable, 11132.5
gettyimages
Credit: Westend61


EigenFace
Unrecognizable, 10407.728


EigenFace
Unrecognizable, 11414.67
Shutterstock
shutterstock.com · 2476654675
Visit ›
Learn more

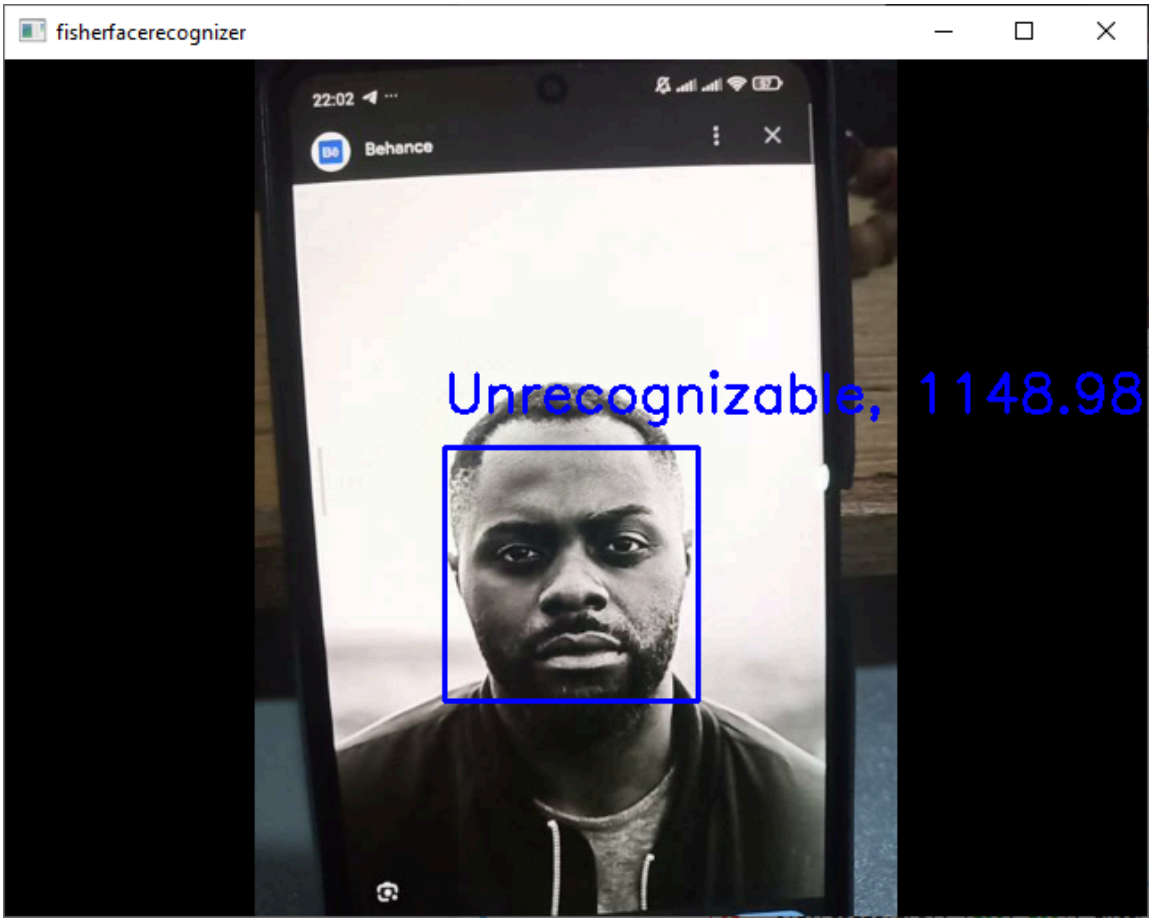

EigenFace
21:48
iStock
Unrecognizable, 9383.20

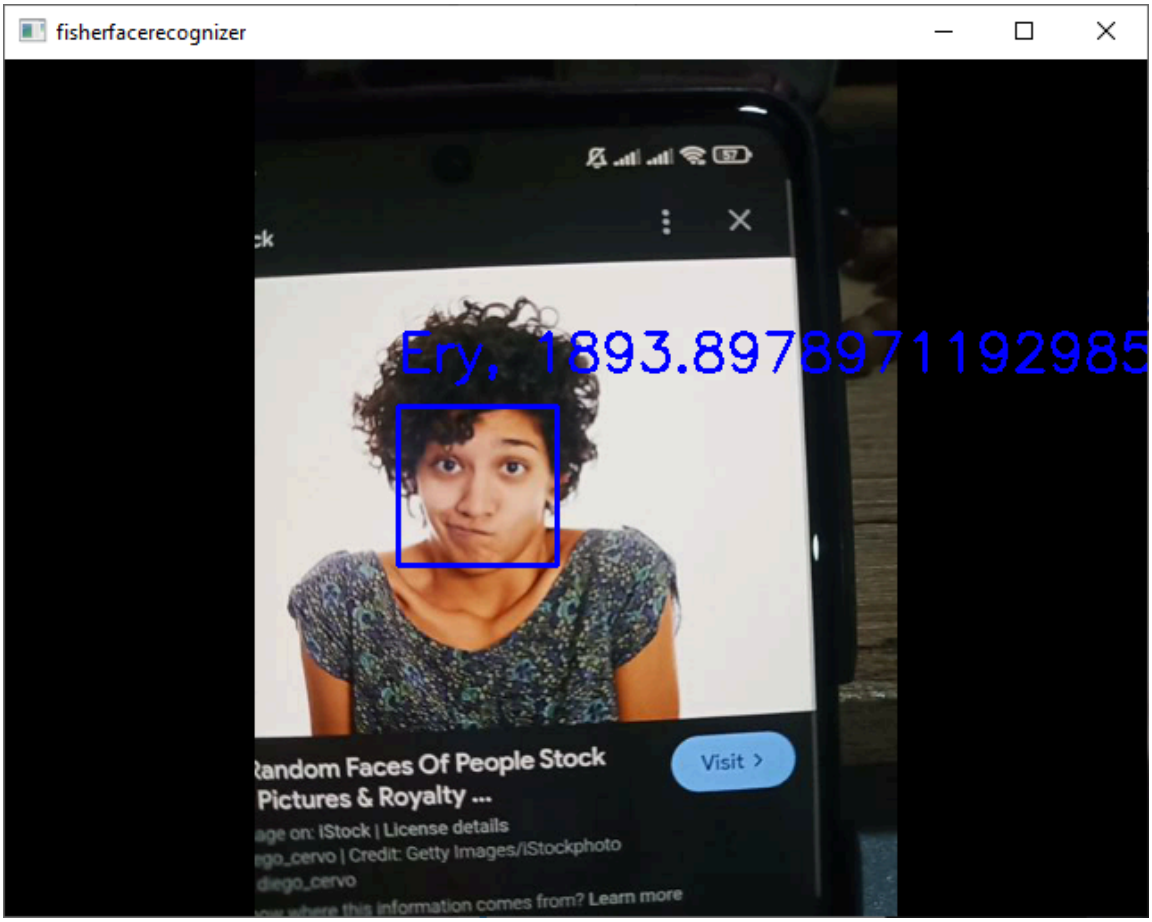

EigenFace
Pinterest
Ery, 8595.345923714625

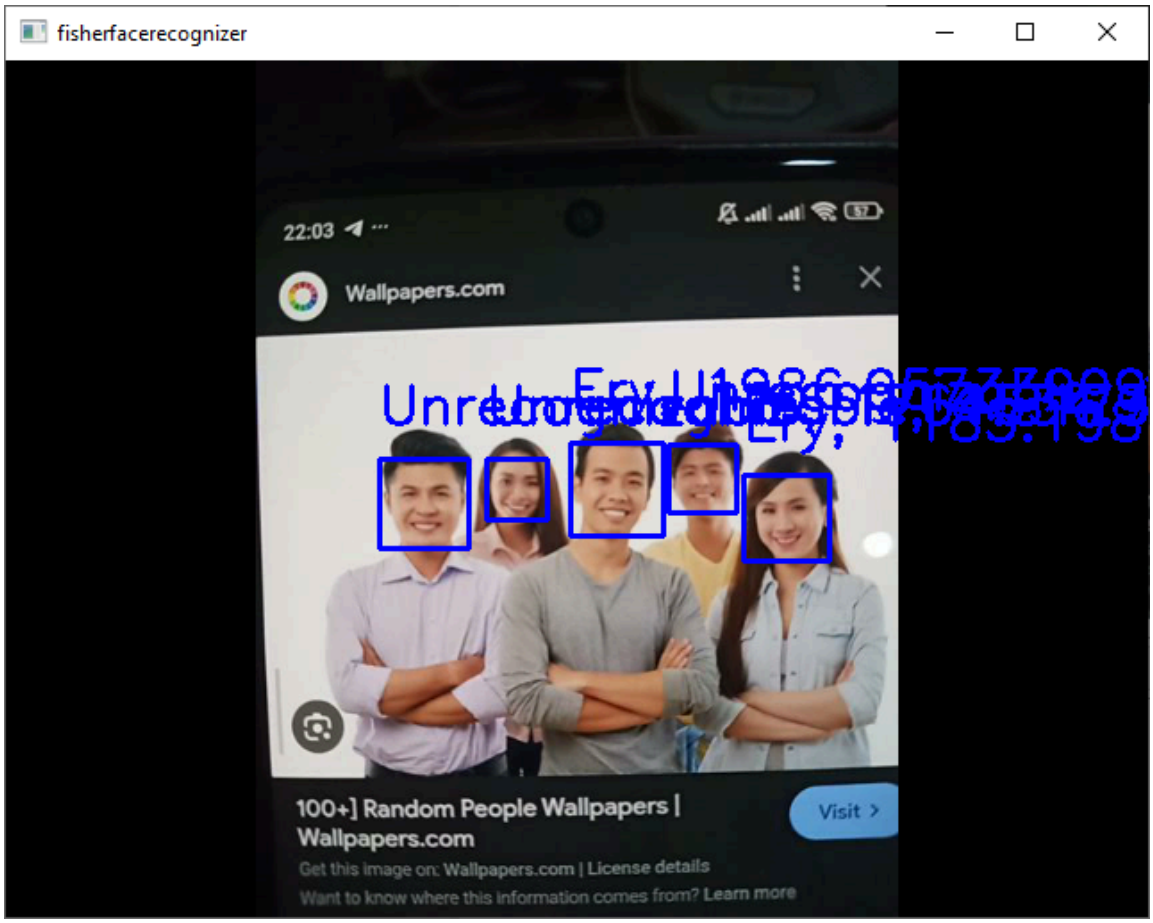


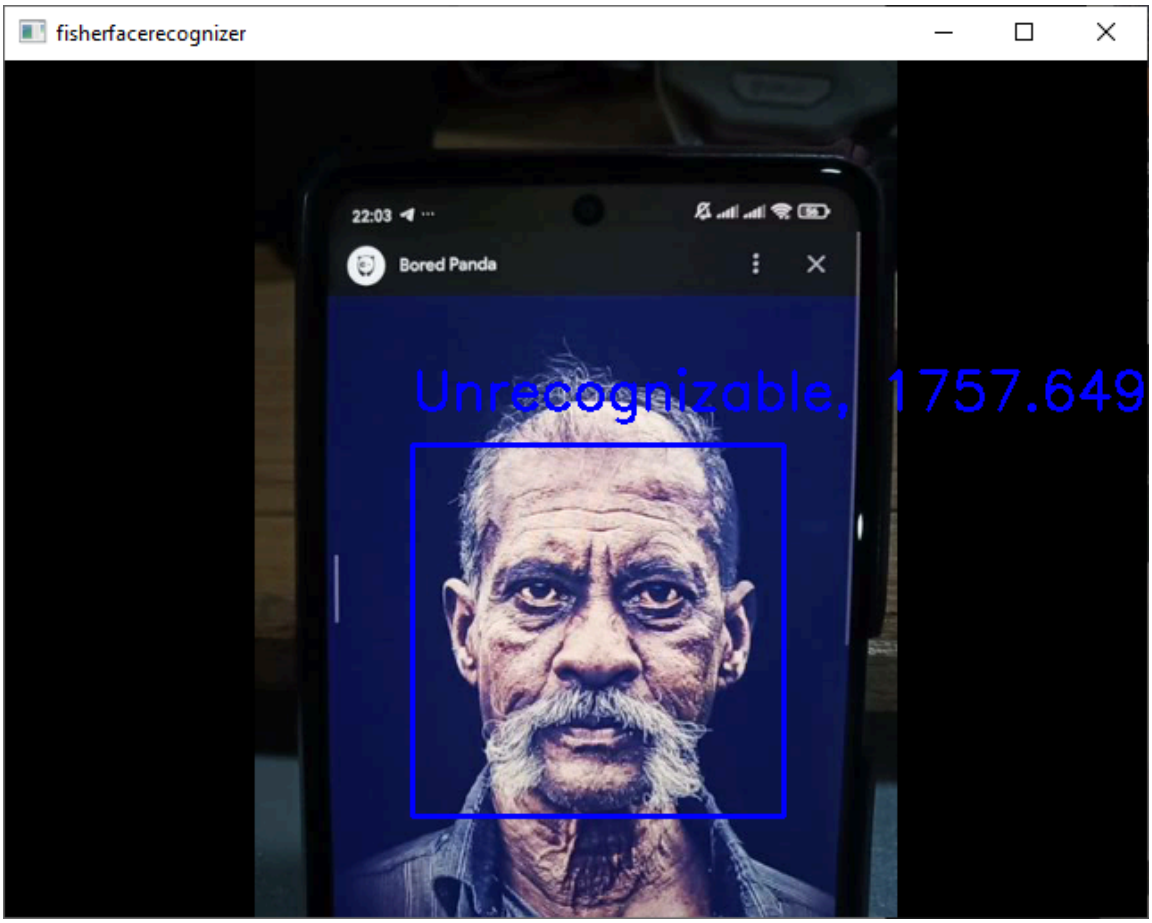
In [25]:
```
# fisher Face recognizer
def face_rec():
  names = ['Ery', 'Unrecognizable'] # Put your names here for faces to recognize
  [X, y] = read_images("Act7_suppl",1)
  y = np.asarray(y, dtype=np.int32)

  model = cv2.face.FisherFaceRecognizer_create()
  model.train(X, y)

  camera = cv2.VideoCapture(0)
  face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

  while True:
    ret, img = camera.read()
    if not ret:
      break

    faces = face_cascade.detectMultiScale(img, 1.3, 5)

    for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
      gray = cv2.cvtColor(img[y:y + h, x:x + w], cv2.COLOR_BGR2GRAY)
      roi = cv2.resize(gray, (200, 200), interpolation=cv2.INTER_LINEAR)

      try:
        params = model.predict(roi)
        label = names[params[0]]
```

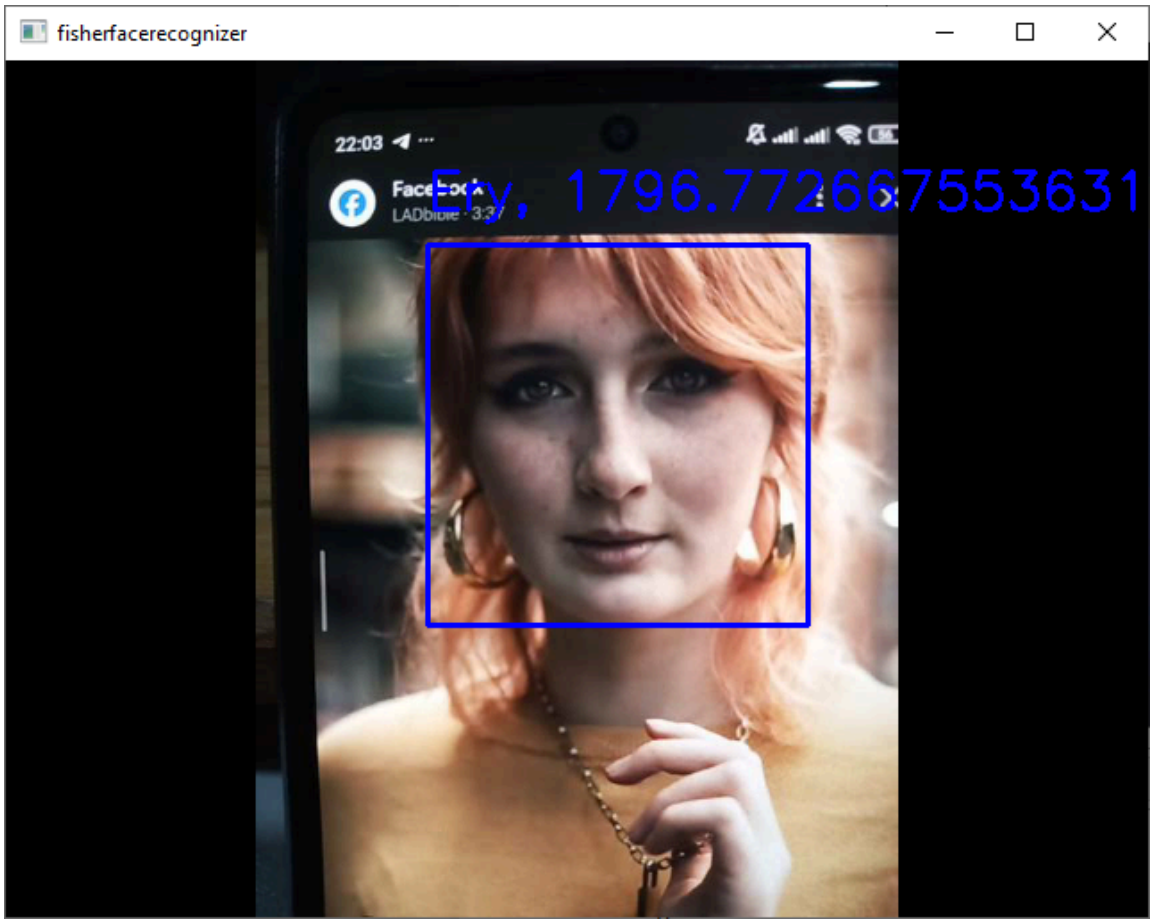
```
        cv2.putText(img, label + ", " + str(params[1]), (x, y - 20), cv2.FONT_HERSH
      except:
        continue

    cv2.imshow("fisherfacerecognizer", img)
    if cv2.waitKey(1) & 0xFF == ord("q"):
      break

  camera.release()
  cv2.destroyAllWindows()

if __name__ == "__main__":
  face_rec()
```

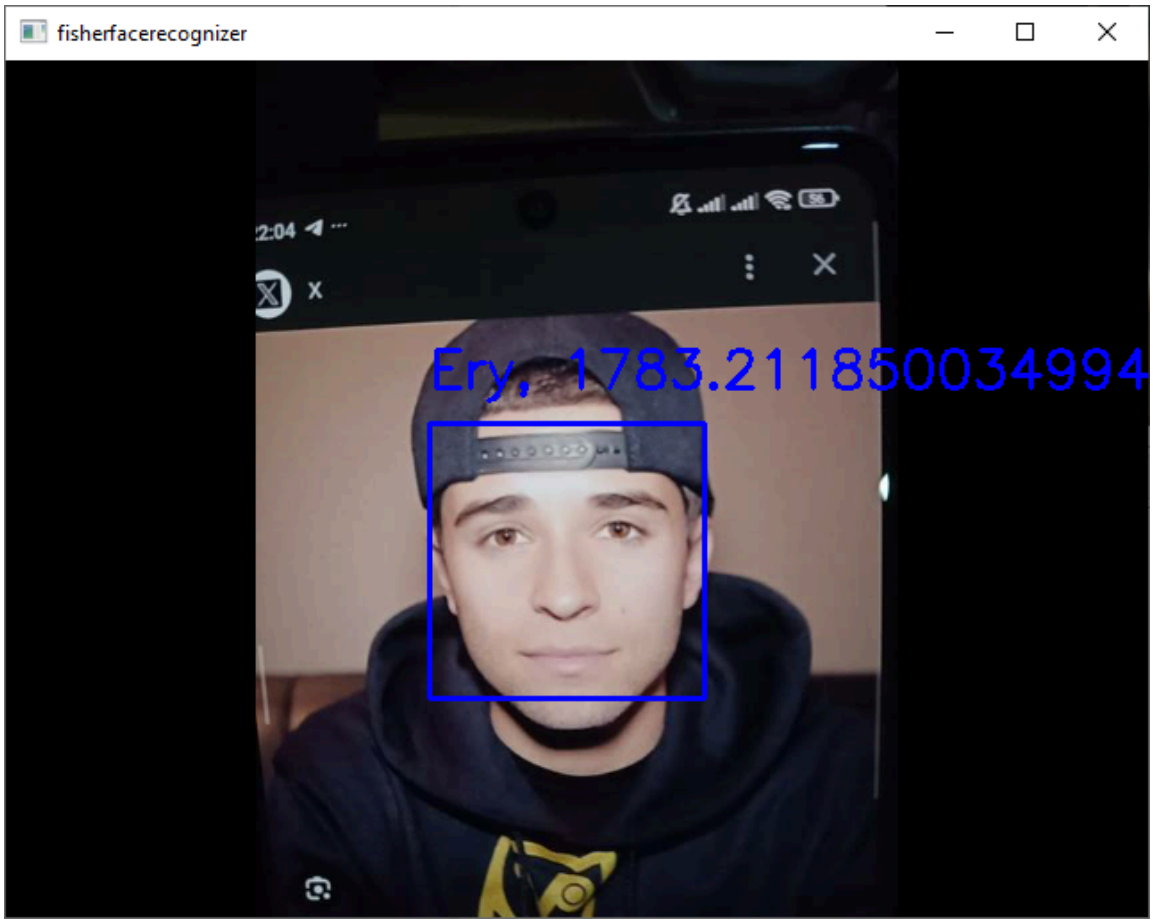


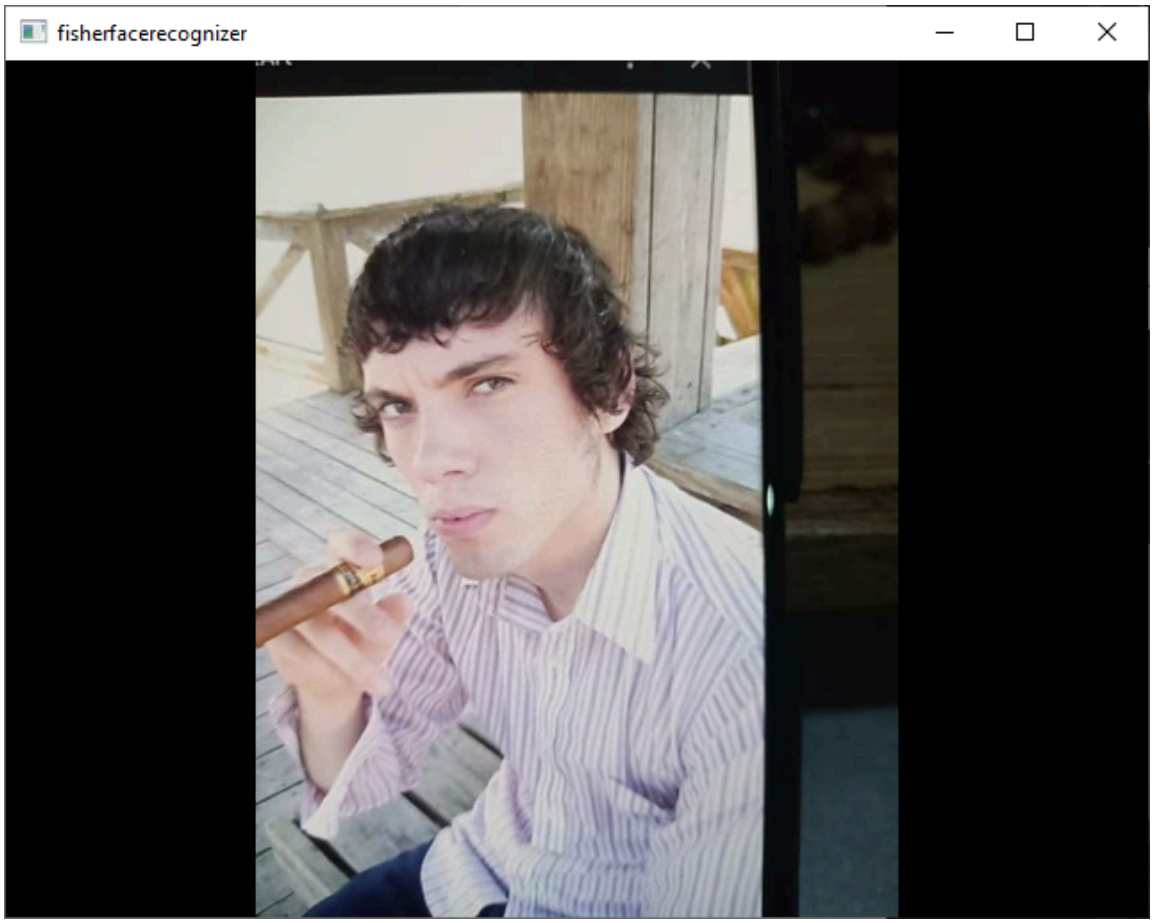

fisherfacerecognizer
Ery, 1893.8978971192985
Visit
Pictures & Royalty ...
License details
Credit: Getty Images/iStockphoto
Learn more

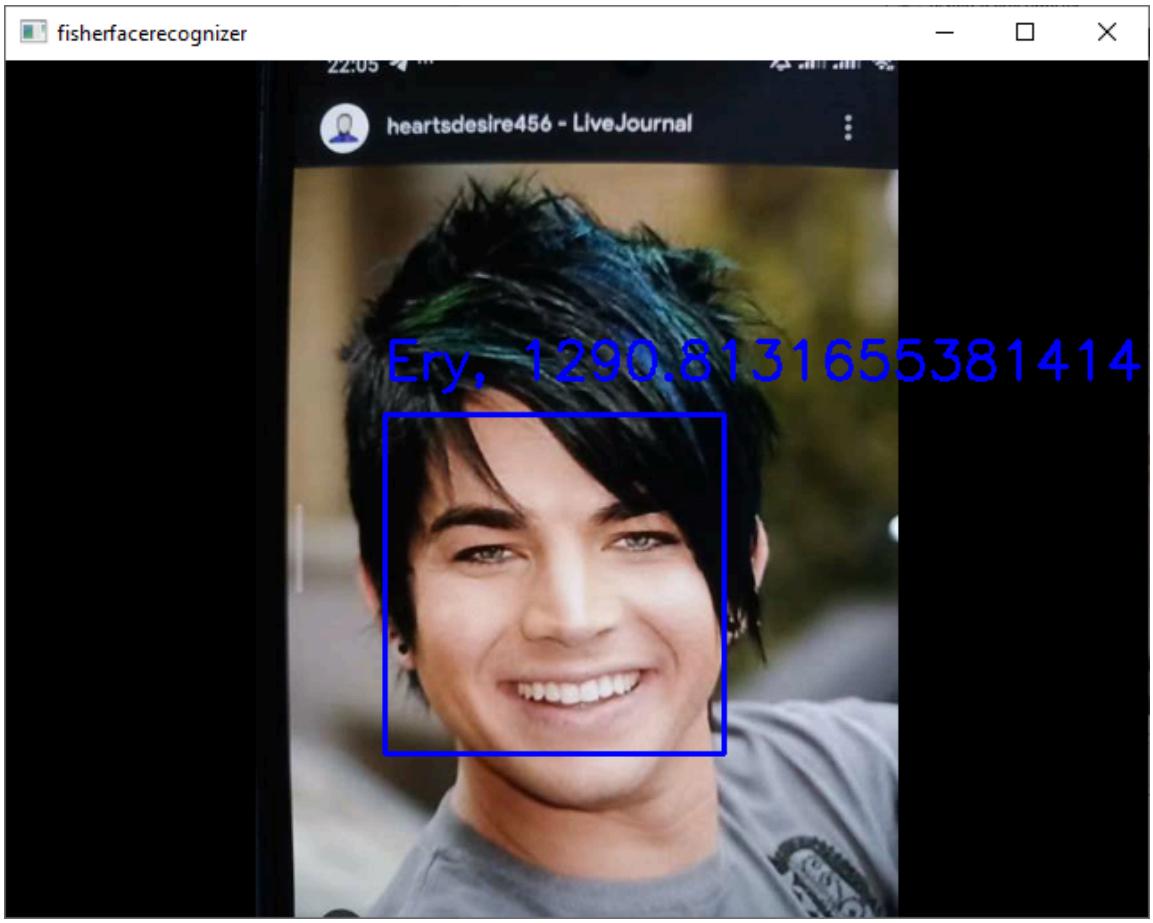

fisherfacerecognizer
22:03
Wallpapers.com
100+] Random People Wallpapers | Wallpapers.com
Visit
Get this image on: Wallpapers.com | License details
Want to know where this information comes from? Learn more

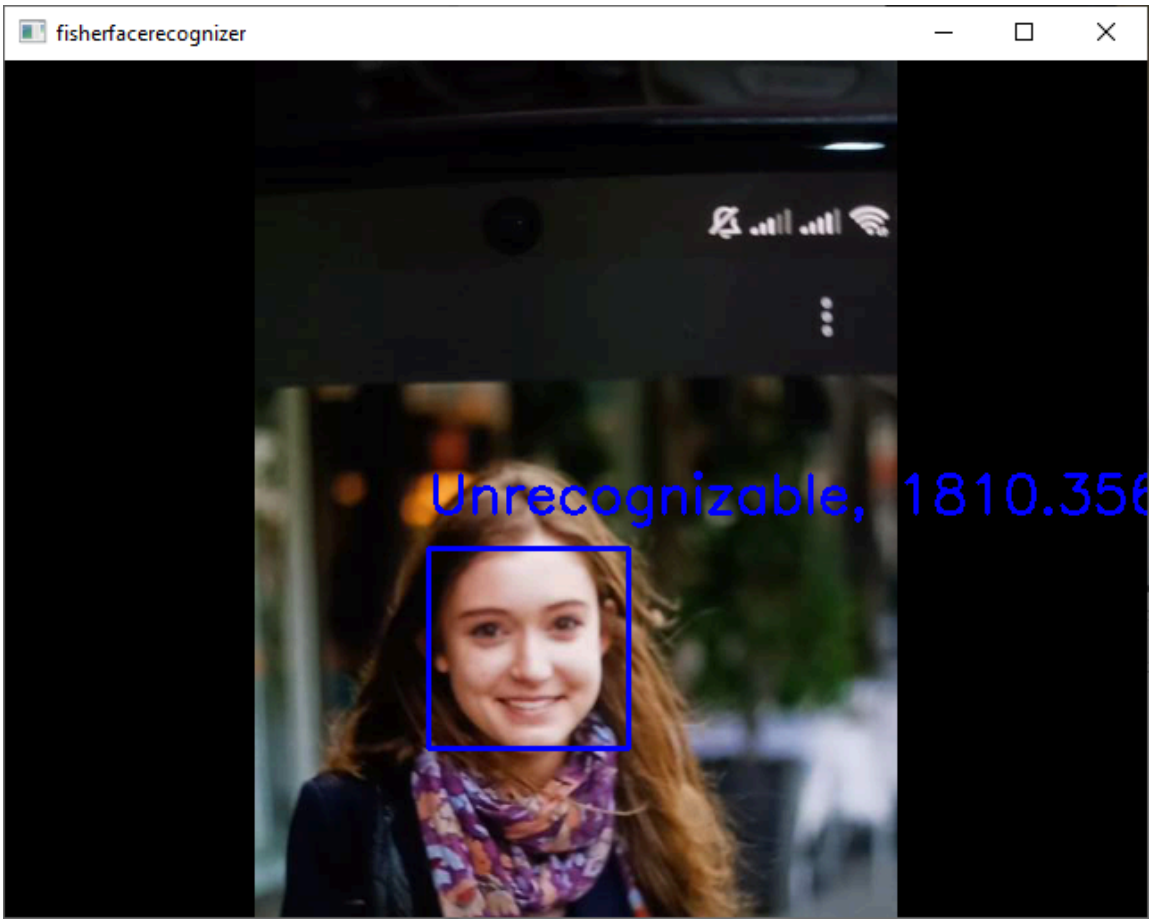

fisherfacerecognizer
22:03
Bored Panda
Unrecognizable, 1757.649

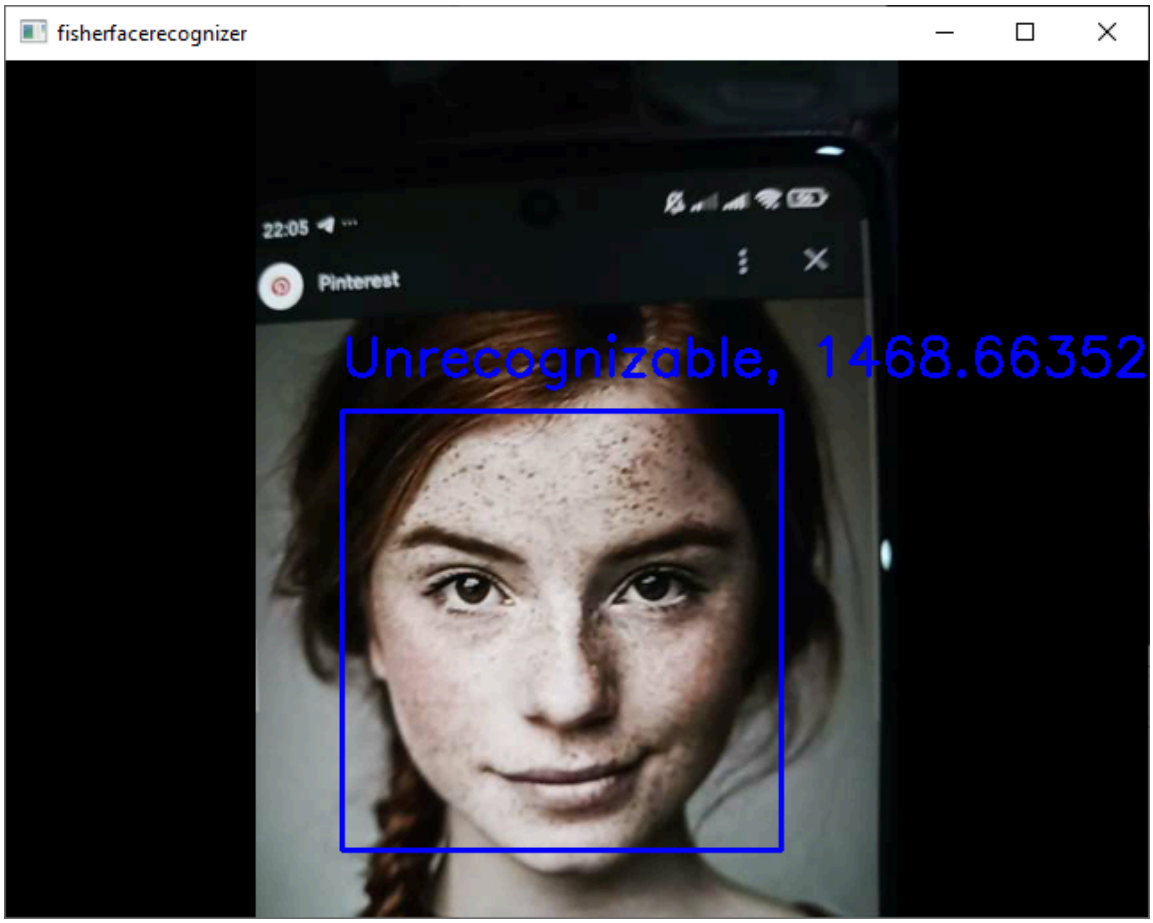

fisherfacerecognizer
22:03
Fey, 1796.772667553631

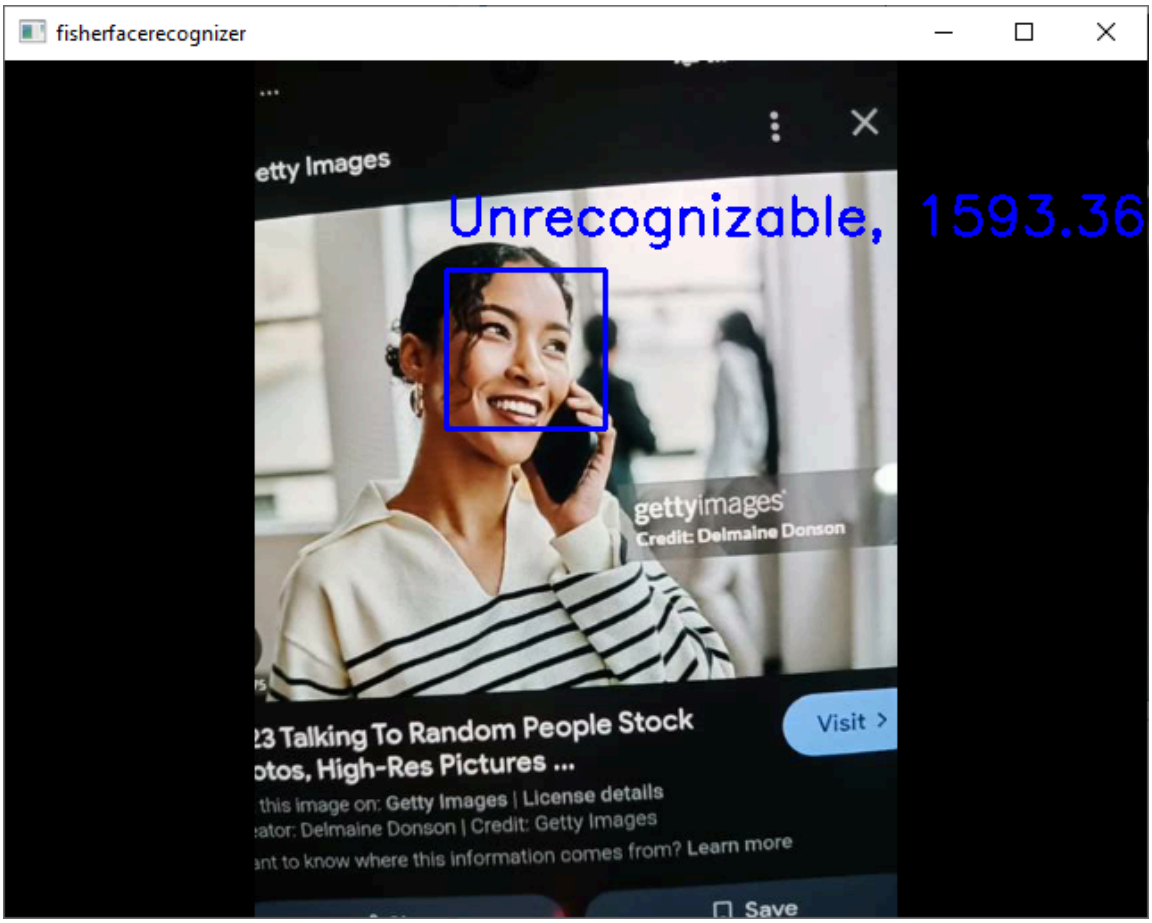

fisherfacerecognizer
Ery, 1783.211850034994

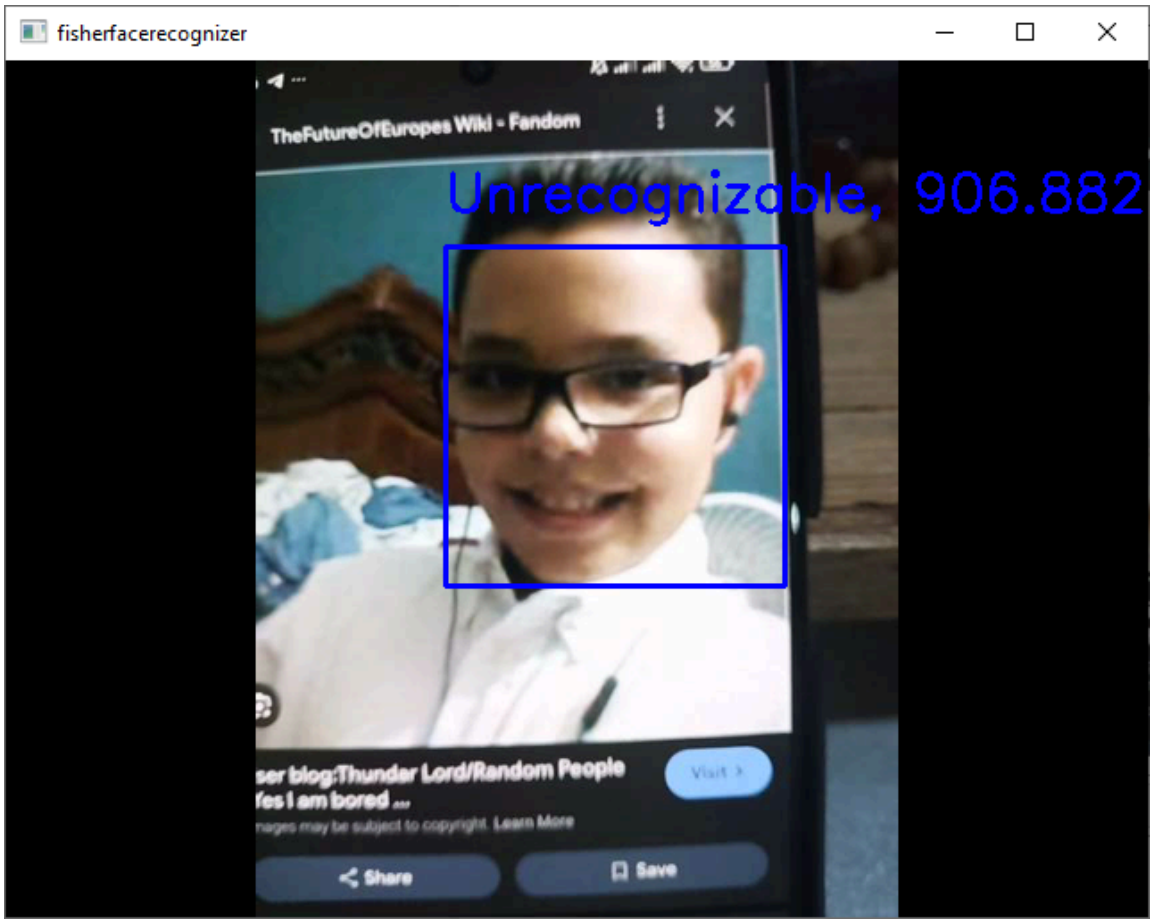

fisherfacerecognizer

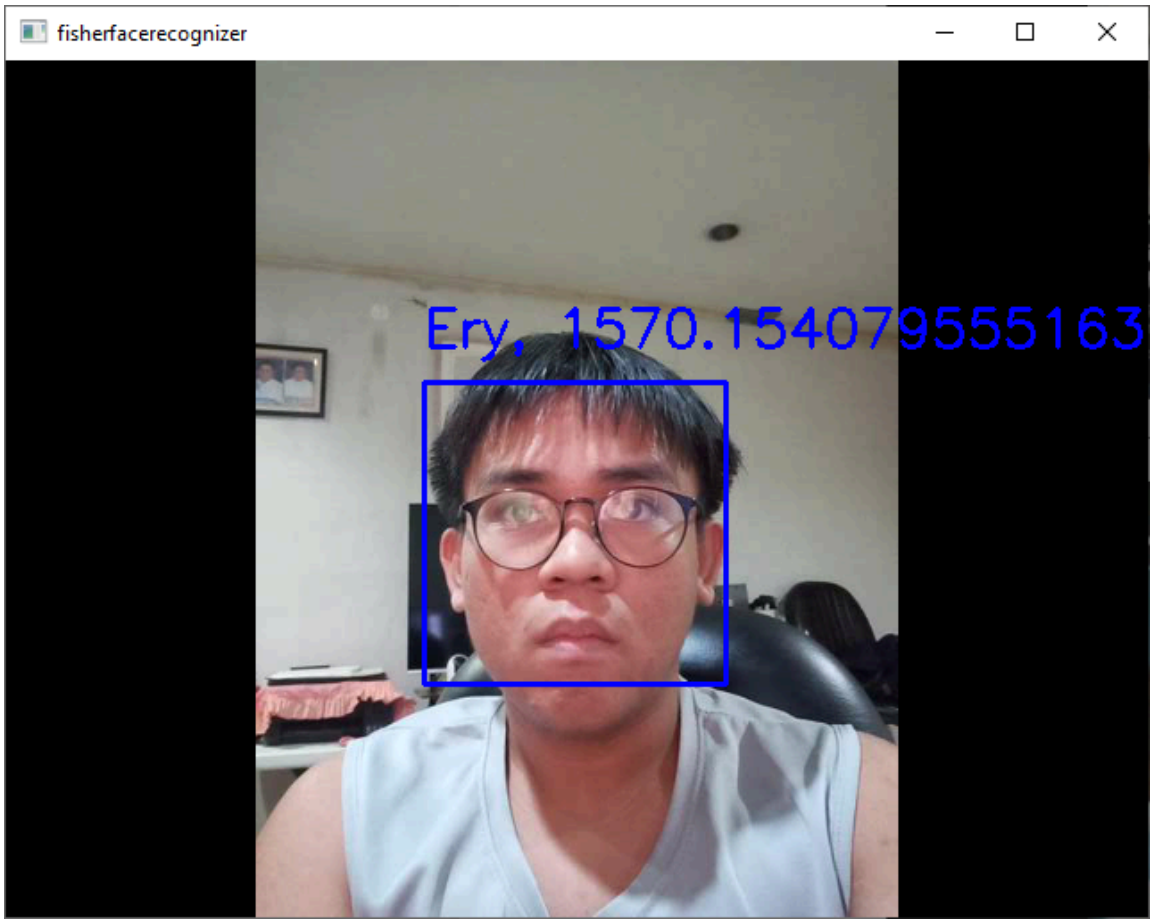

fisherfacerecognizer
heartsdesire456 - LiveJournal
Ery, 1290.8131655381414

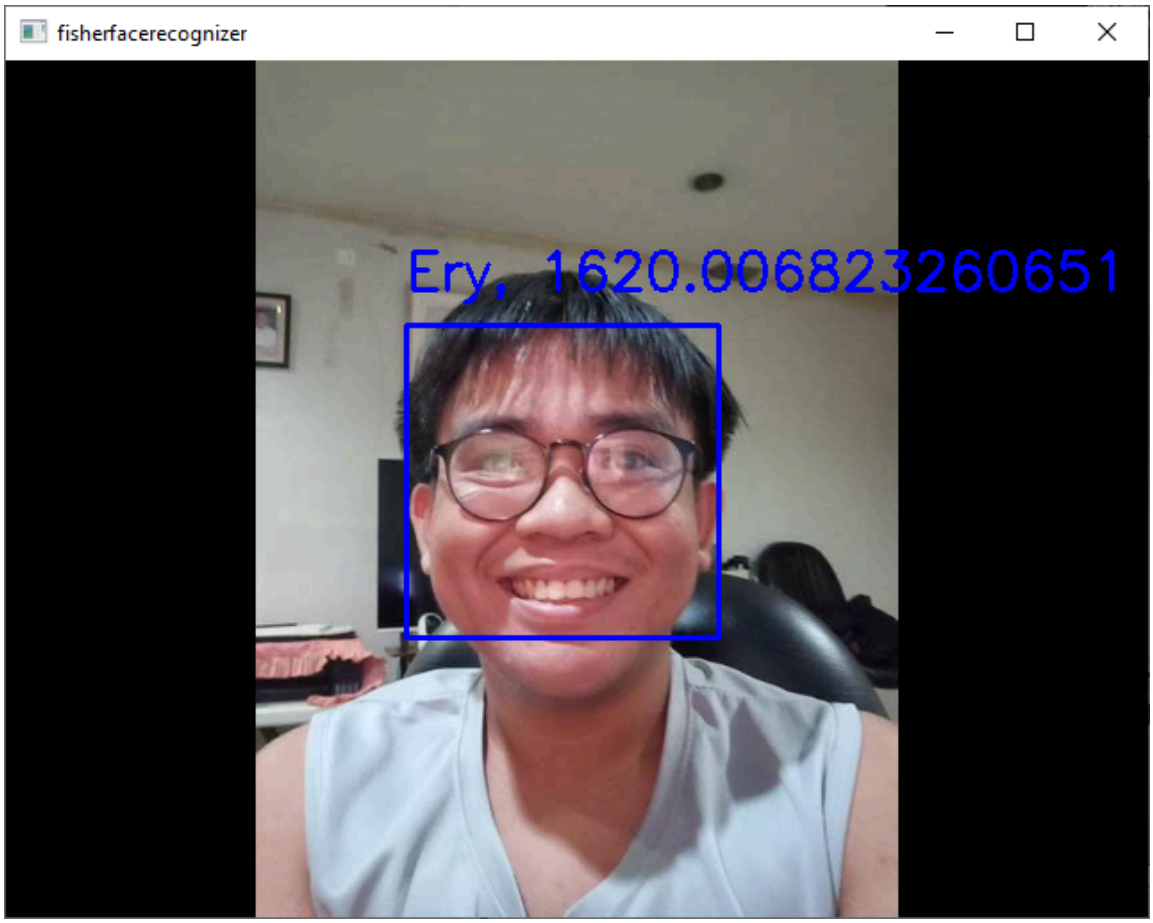

fisherfacerecognizer
Unrecognizable, 1810.356

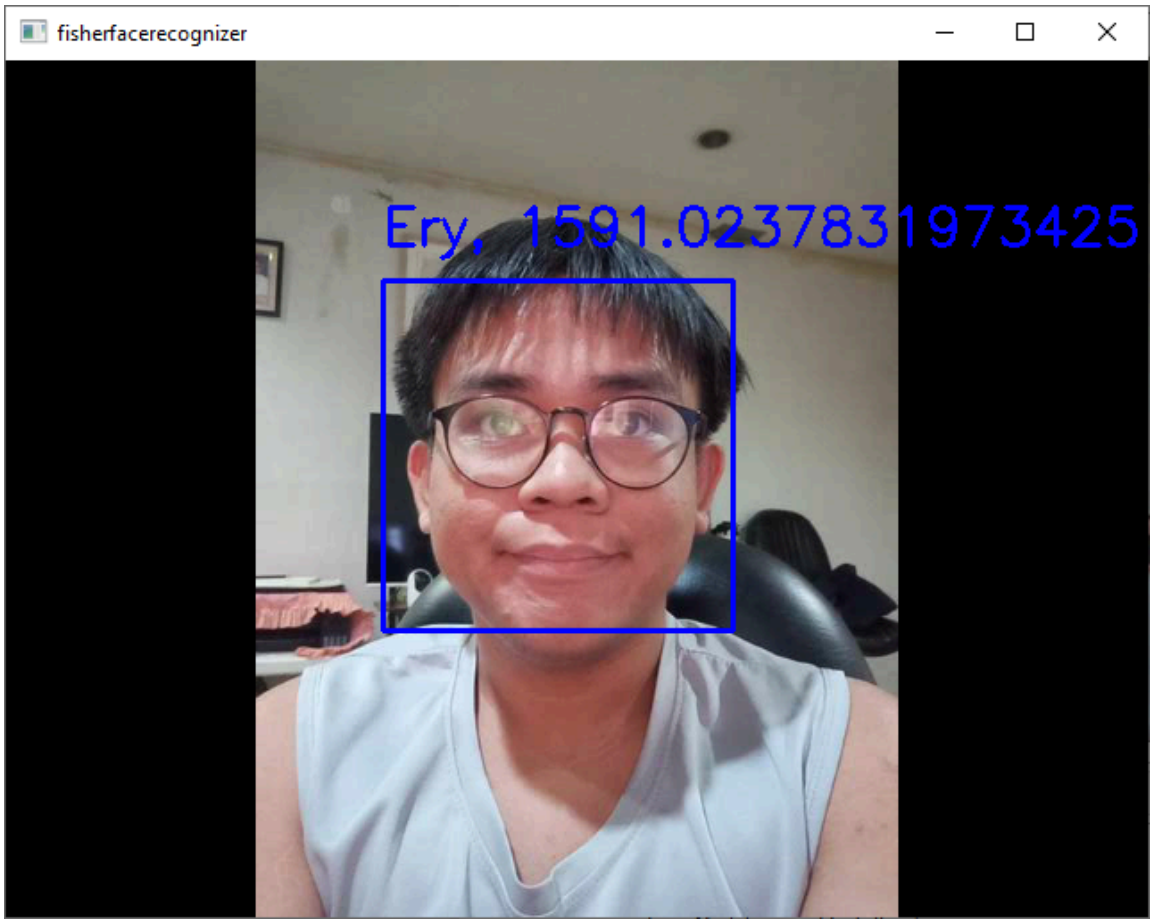

fisherfacerecognizer
22:05
Pinterest
Unrecognizable, 1468.66352

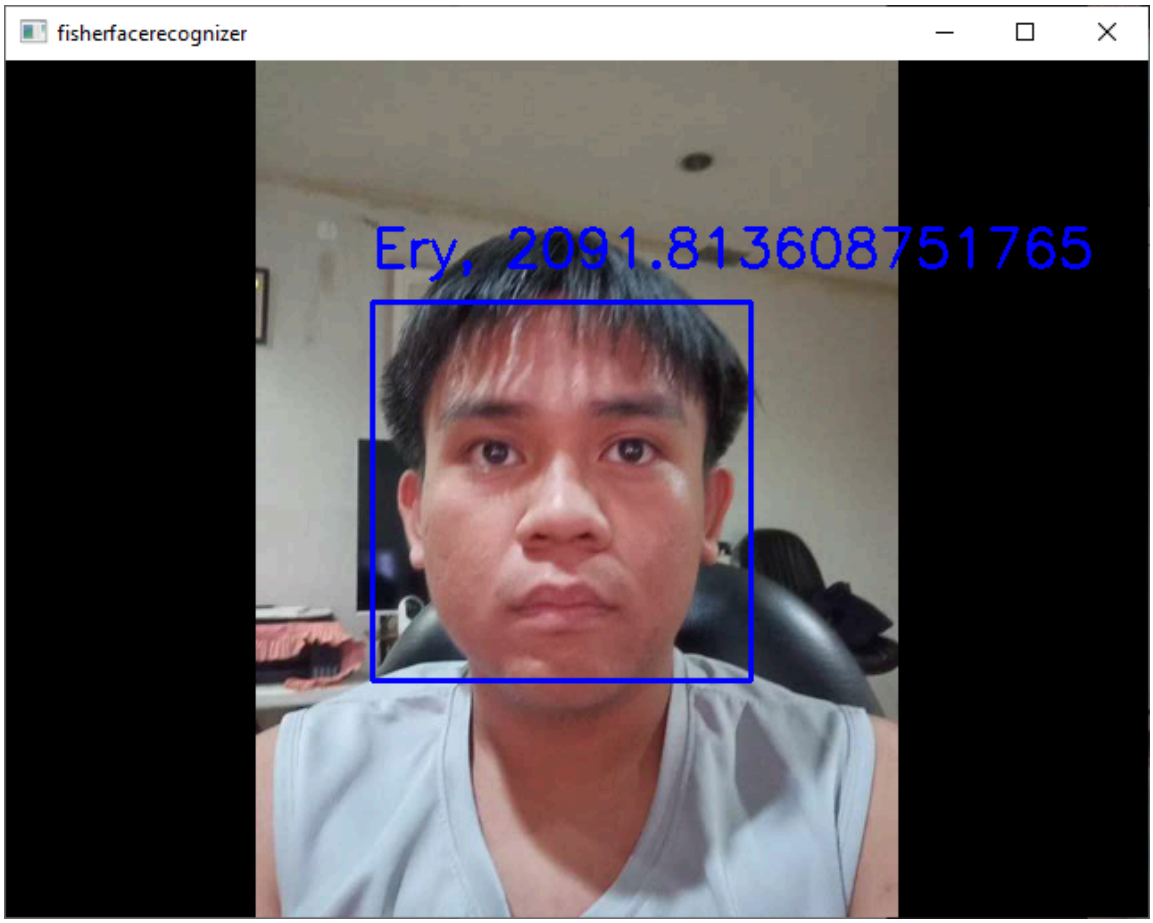

fisherfacerecognizer
Unrecognizable, 1593.36
gettyimages
Credit: Delmaine Donson
Visit
Save

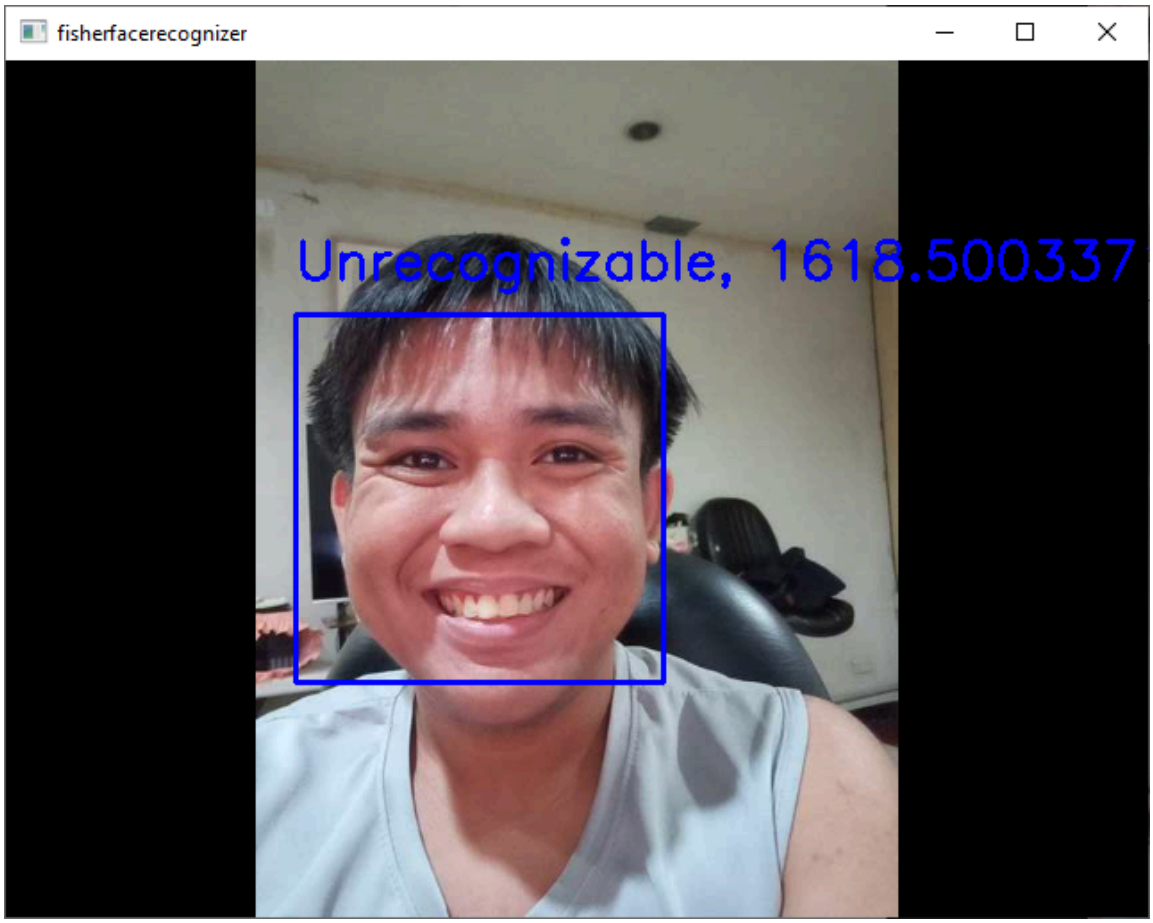

fisherfacerecognizer
TheFutureOfEuropes Wiki - Fandom
Unrecognizable, 906.882
ser blog:Thunder Lord/Random People
fes I am bored ...
Visit
Share
Save

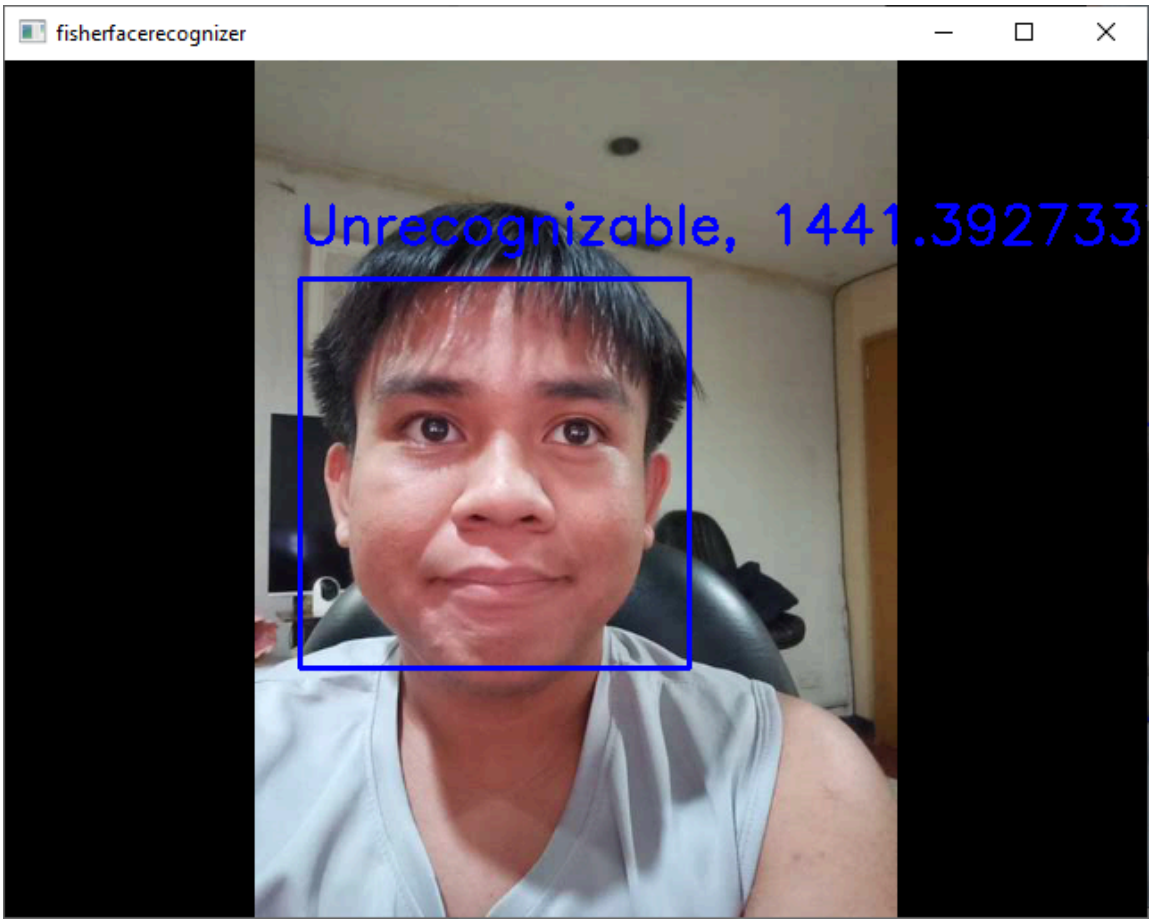

fisherfacerecognizer
Ery, 1570.154079555163

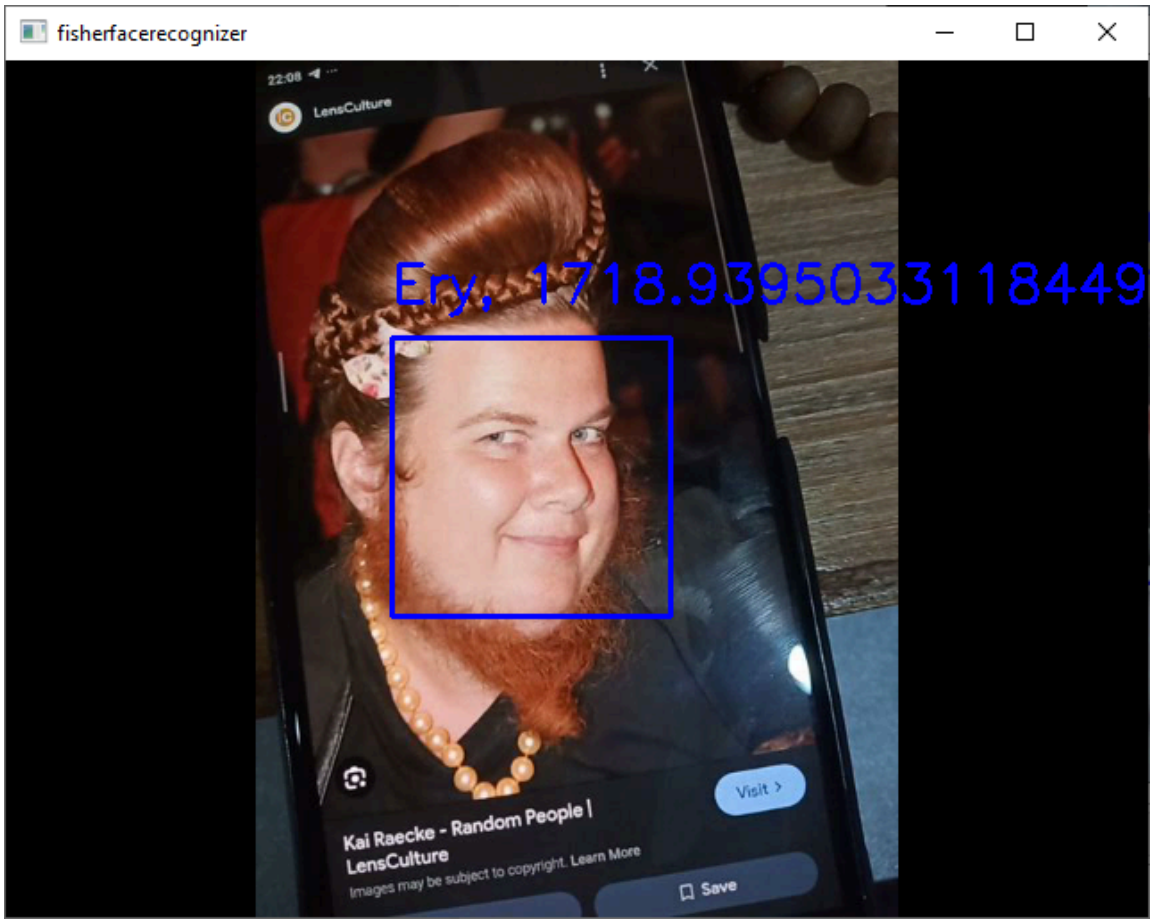

fisherfacerecognizer
Ery, 1620.006823260651

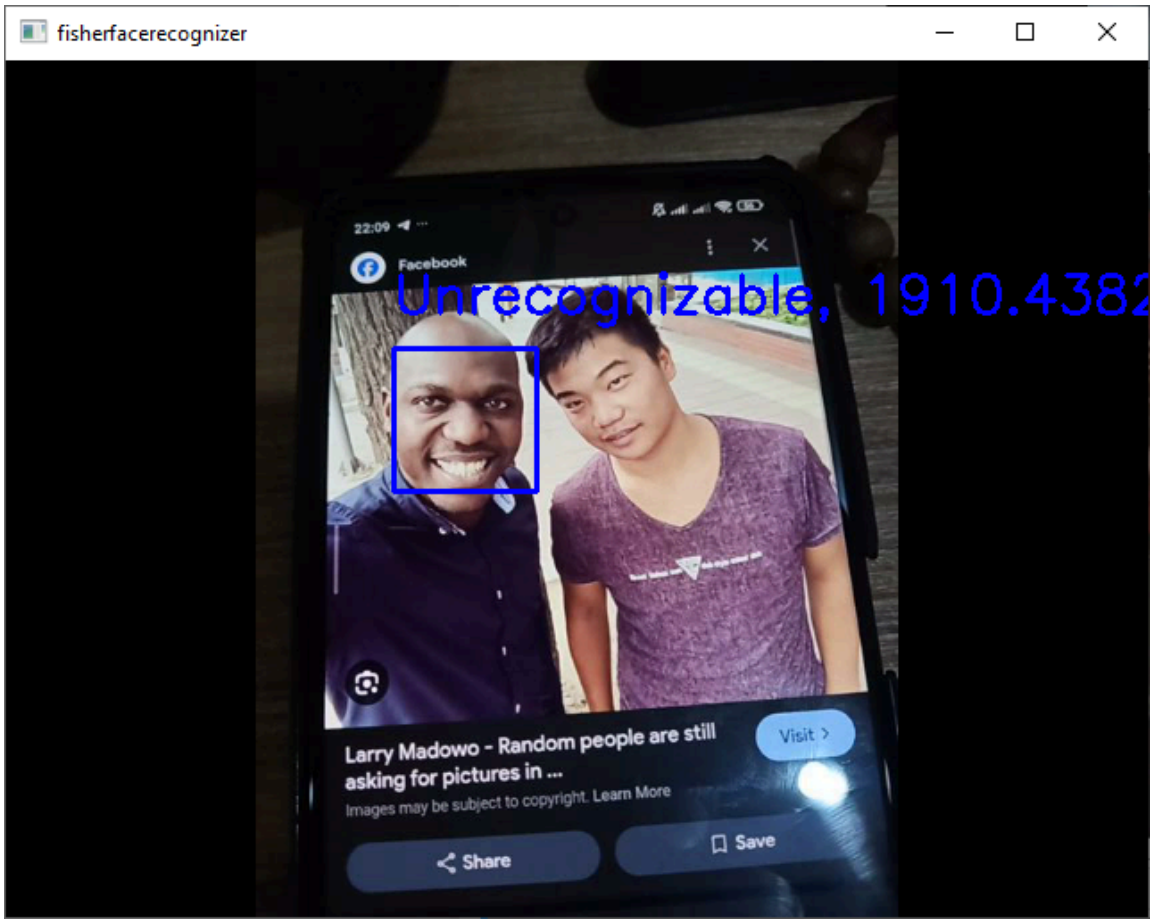

fisherfacerecognizer
Ery, 1591.0237831973425


fisherfacerecognizer
Ery, 2091.813608751765


fisherfacerecognizer
Unrecognizable, 1618.500337

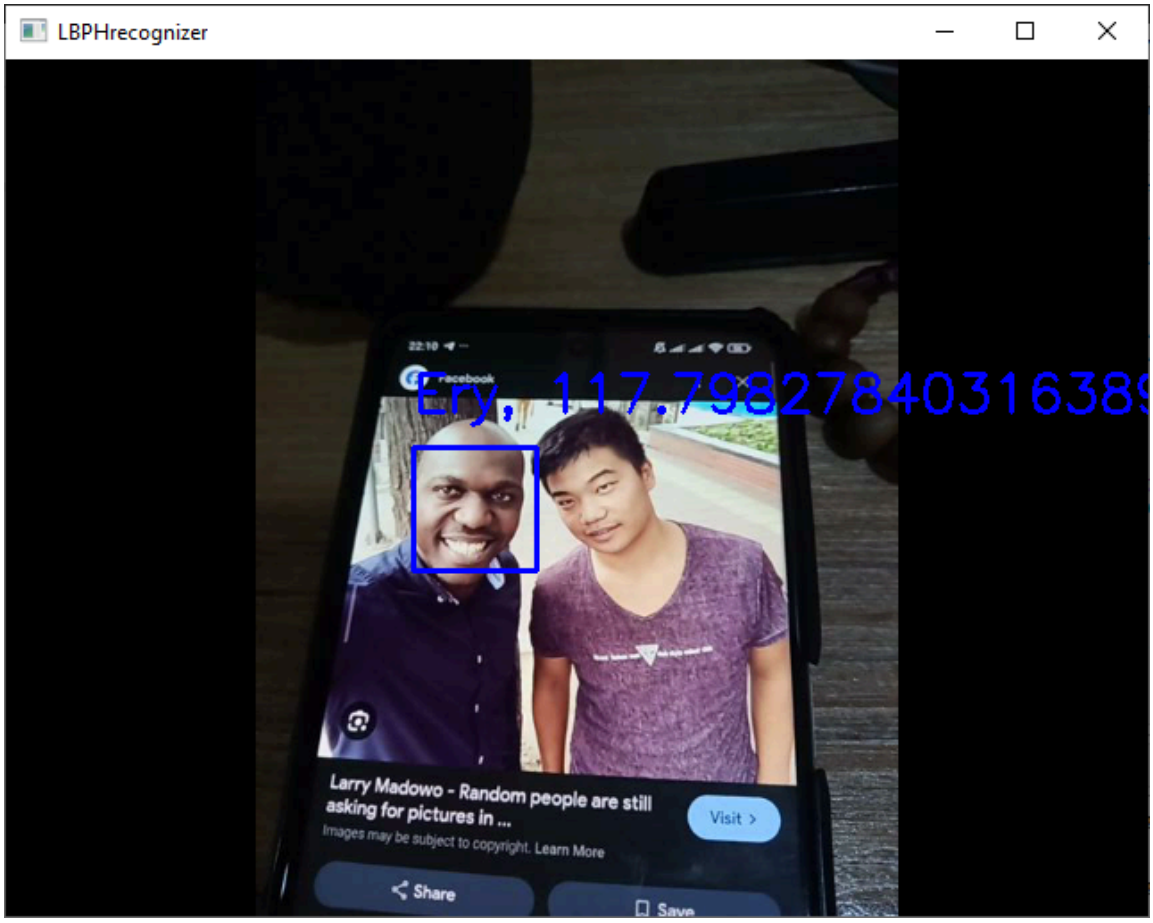

fisherfacerecognizer
Unrecognizable, 1441.392733

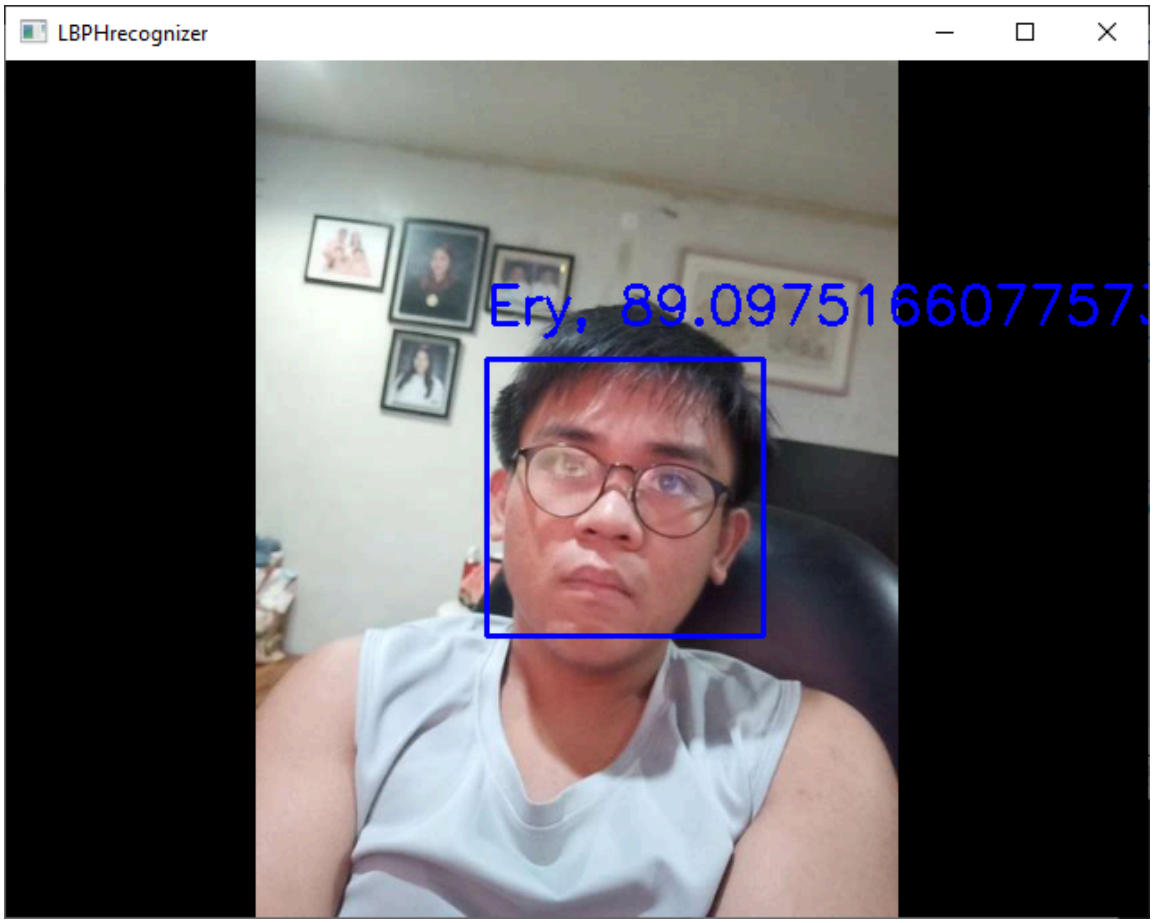

fisherfacerecognizer
LensCulture
Ery, 1718.9395033118449
Kai Raecke - Random People | LensCulture
Images may be subject to copyright. Learn More
Visit
Save

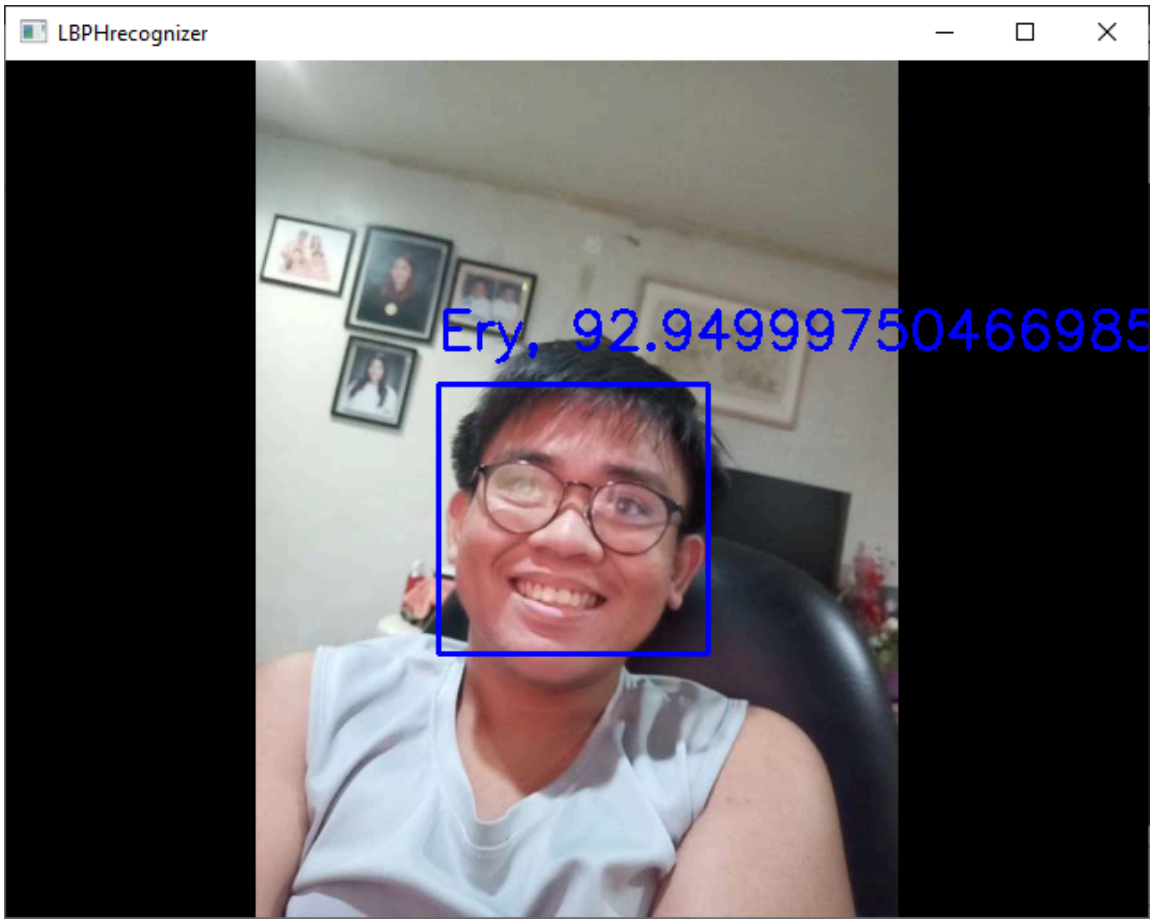


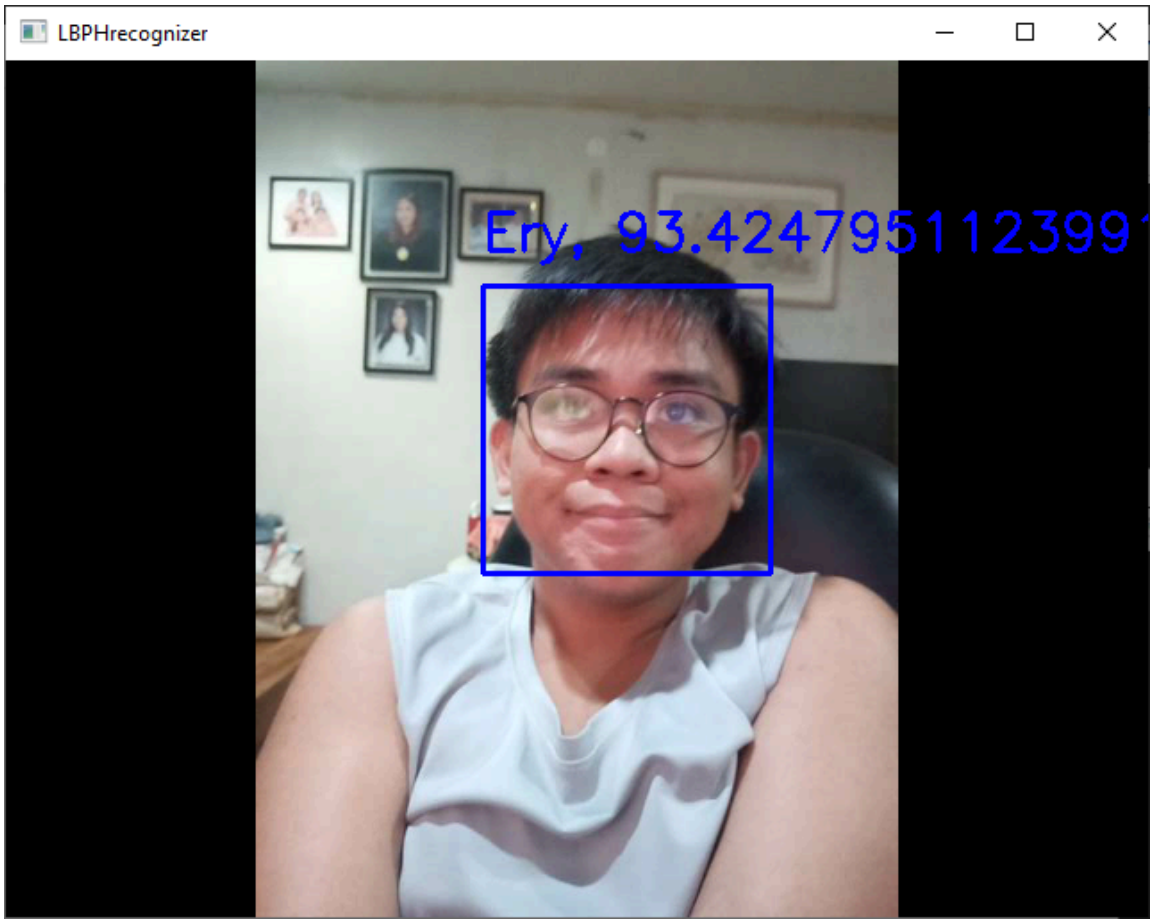
In [27]:
```
# LBPH Face Recognizer
def face_rec():
  names = ['Ery', 'Unrecognizable'] # Put your names here for faces to recognize
  [X, y] = read_images("Act7_suppl",1)
  y = np.asarray(y, dtype=np.int32)

  model = cv2.face.LBPHFaceRecognizer_create()
  model.train(X, y)

  camera = cv2.VideoCapture(0)
  face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

  while True:
    ret, img = camera.read()
    if not ret:
      break

    faces = face_cascade.detectMultiScale(img, 1.3, 5)

    for (x, y, w, h) in faces:
      cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
      gray = cv2.cvtColor(img[y:y + h, x:x + w], cv2.COLOR_BGR2GRAY)
      roi = cv2.resize(gray, (200, 200), interpolation=cv2.INTER_LINEAR)

      try:
        params = model.predict(roi)
        label = names[params[0]]
```

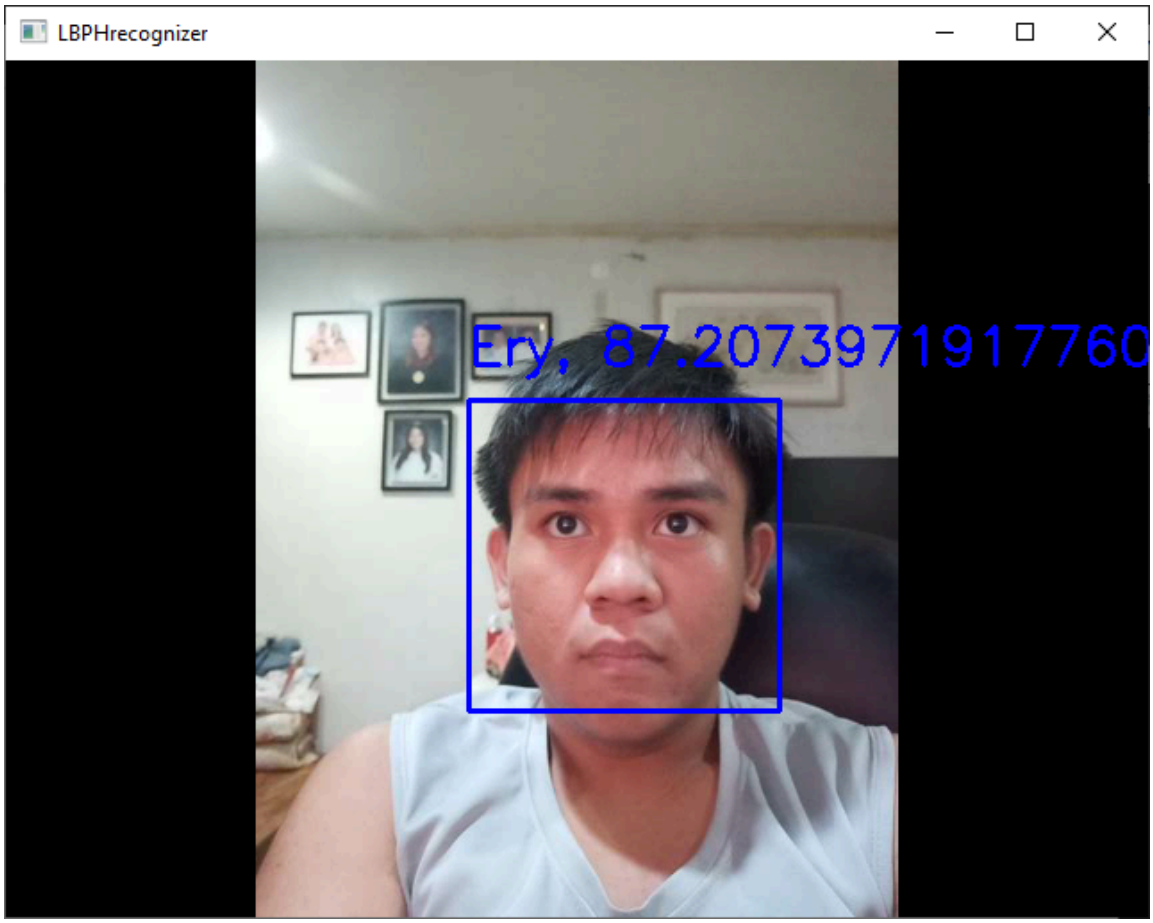
```
        cv2.putText(img, label + ", " + str(params[1]), (x, y - 20), cv2.FONT_HERSH
      except:
        continue

    cv2.imshow("LBPHrecognizer", img)
    if cv2.waitKey(1) & 0xFF == ord("q"):
      break

  camera.release()
  cv2.destroyAllWindows()

if __name__ == "__main__":
  face_rec()
```

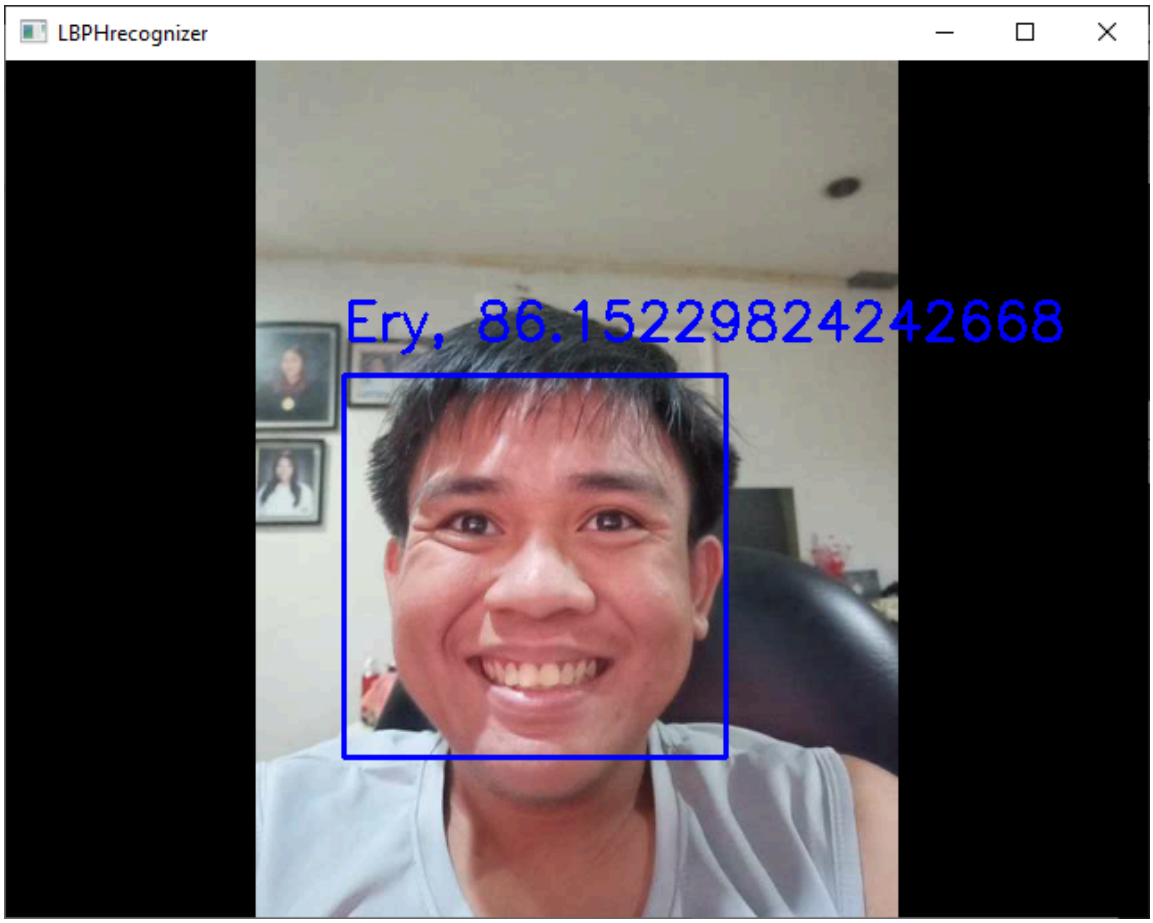


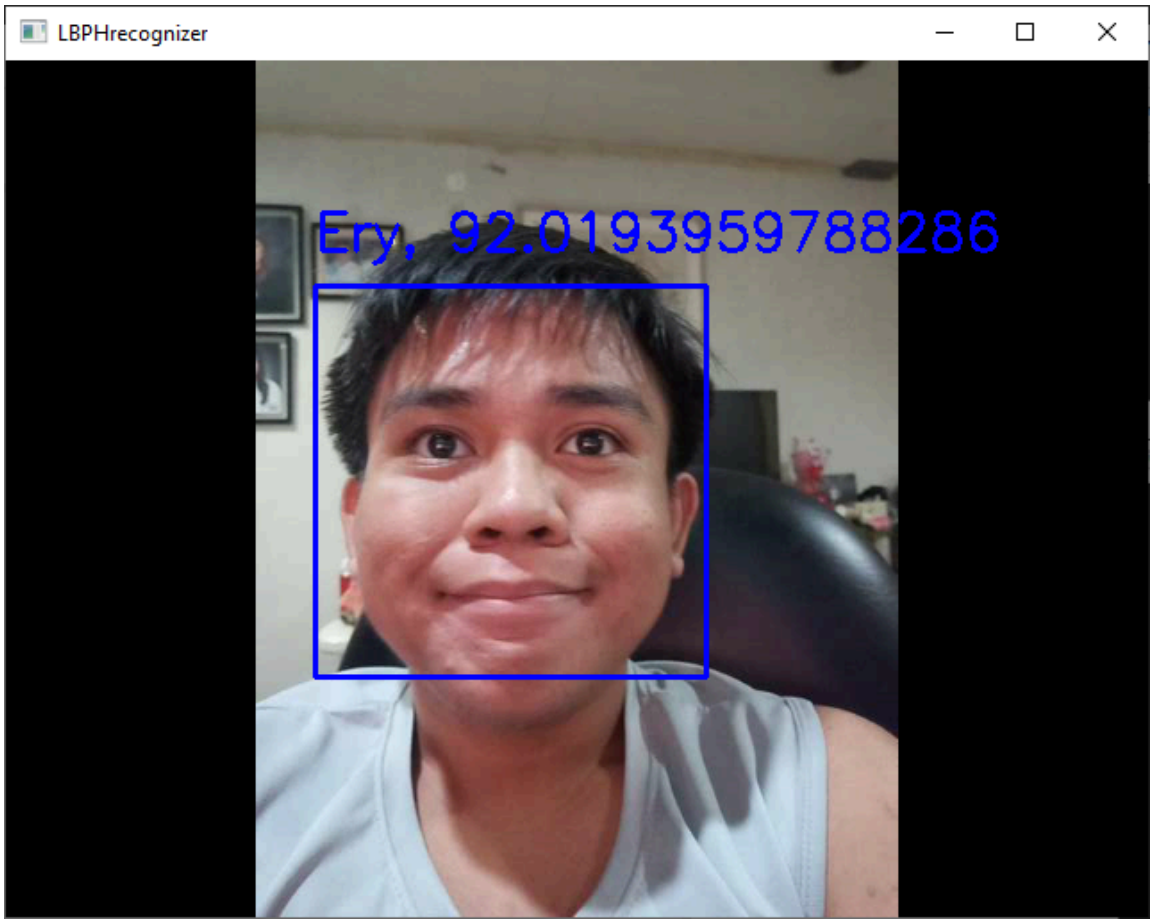

LBPHrecognizer
Ery, 89.0975166077573

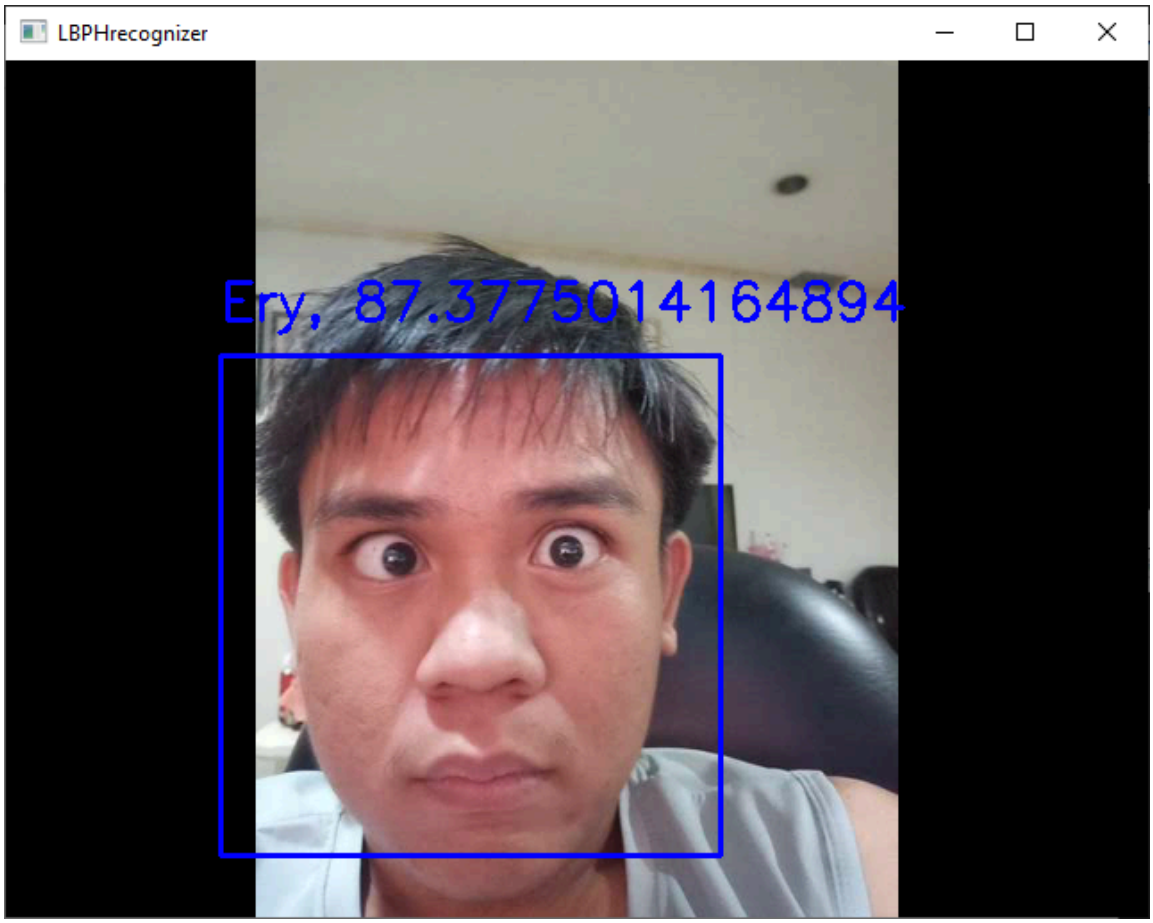

LBPHrecognizer
Ery, 92.94999750466985

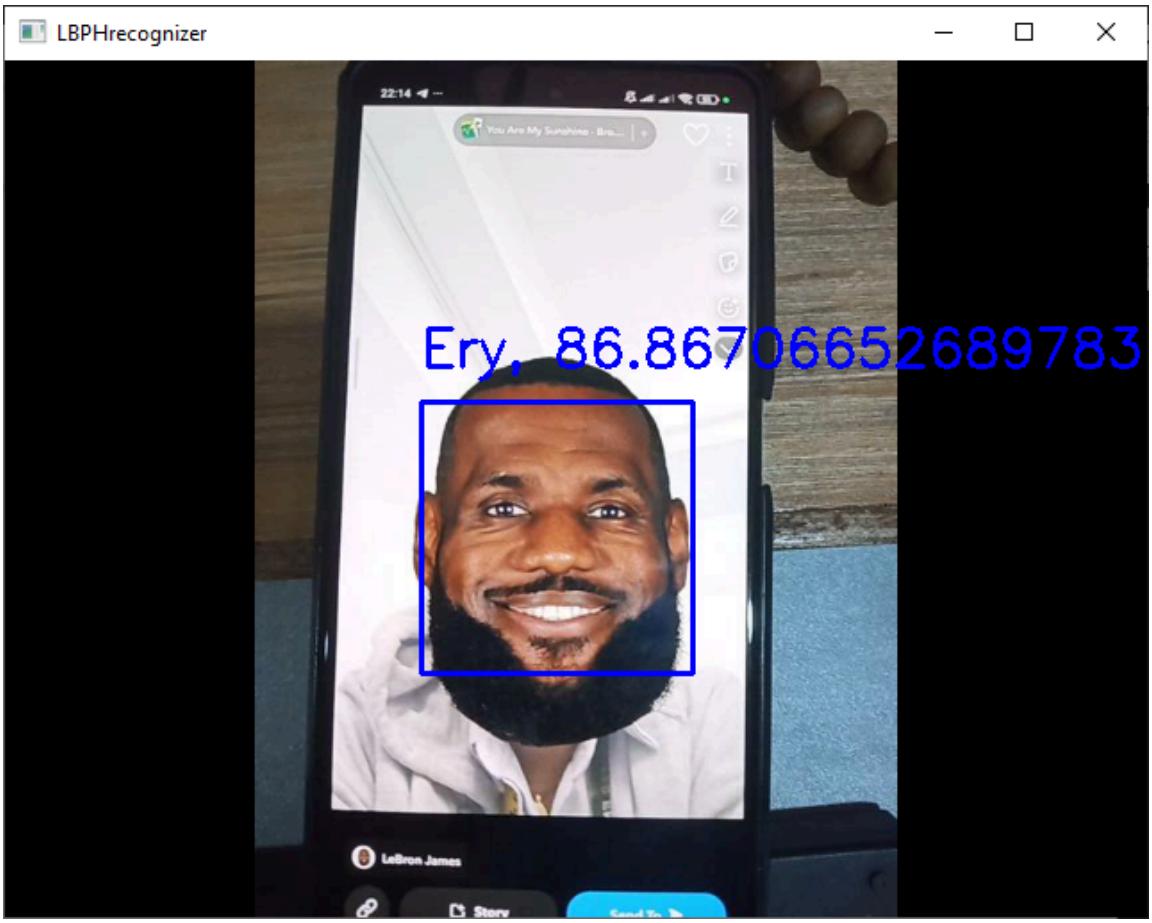

LBPHrecognizer
Ery, 93.4247951123991

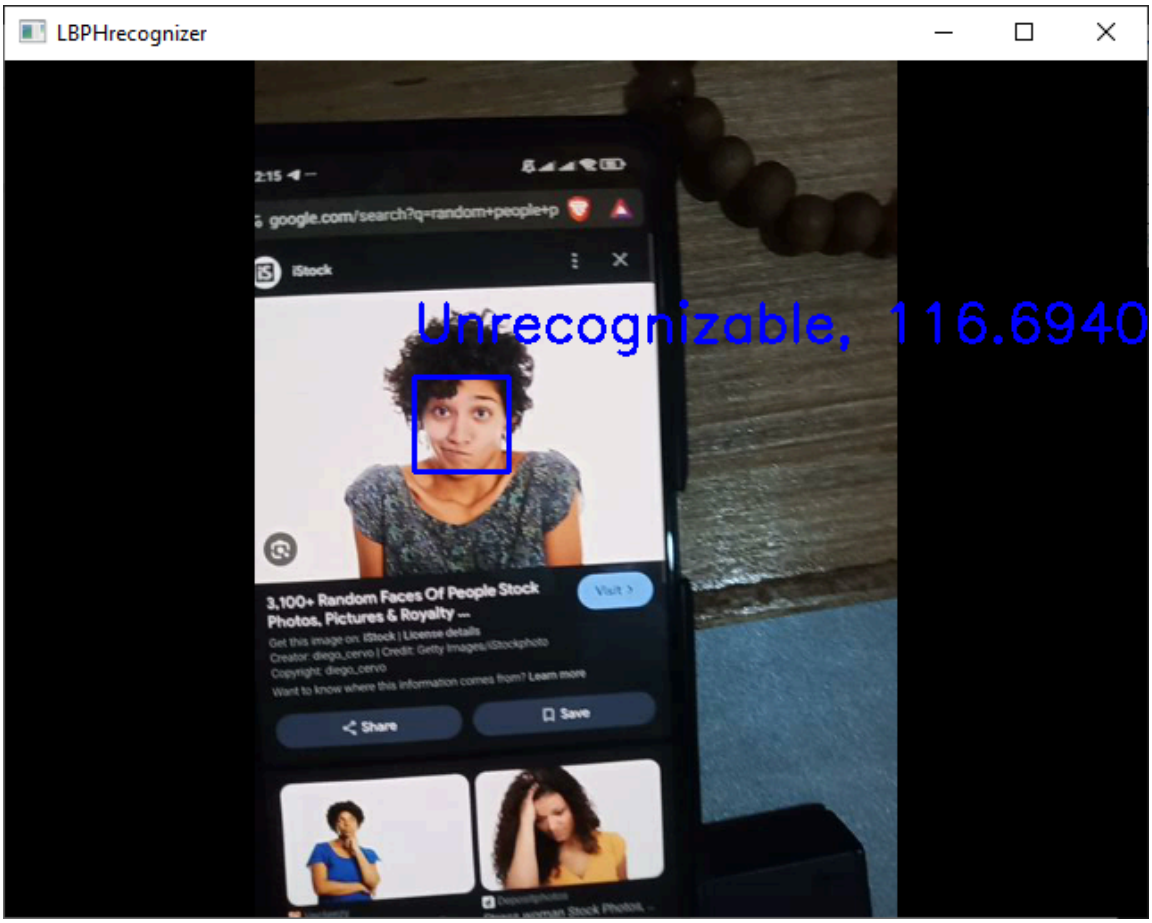

LBPHrecognizer
Ery, 87.2073971917760

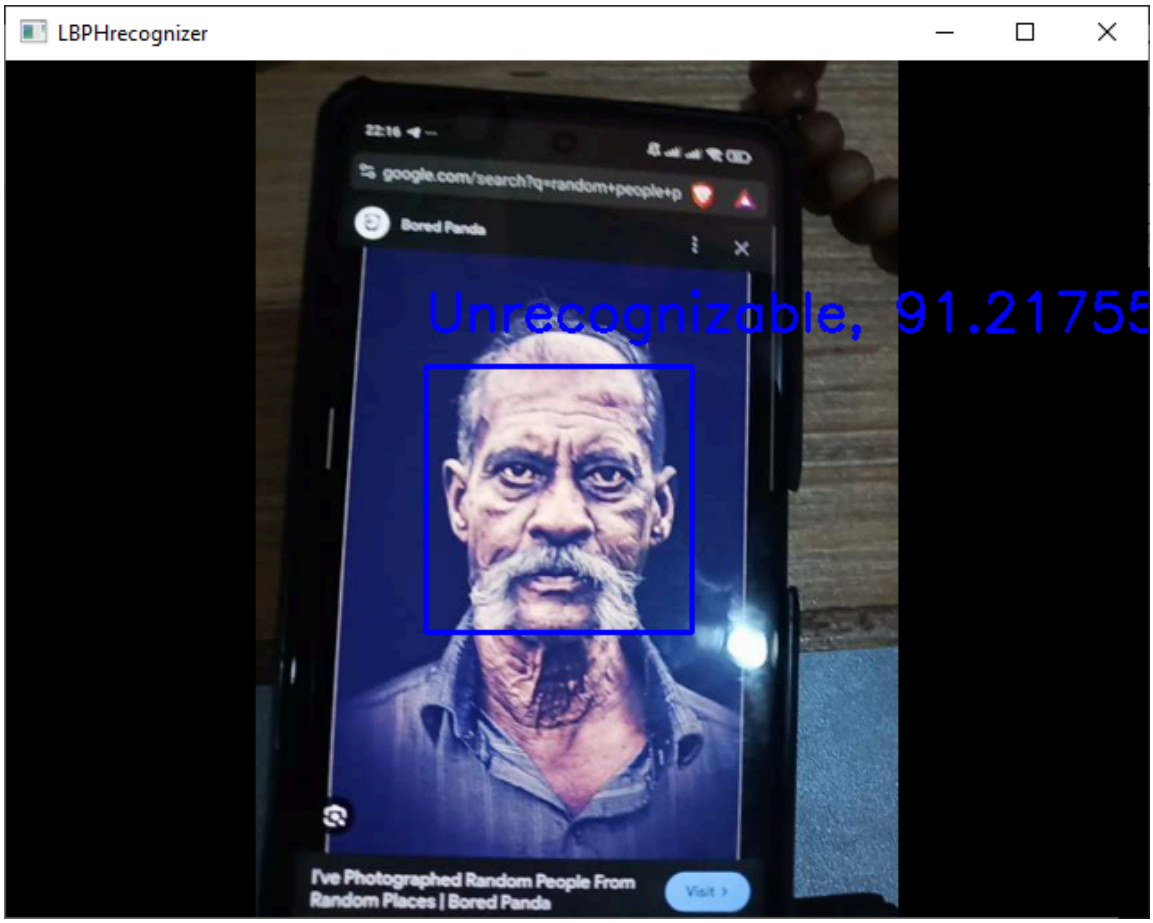

LBPHrecognizer
Ery, 86.15229824242668

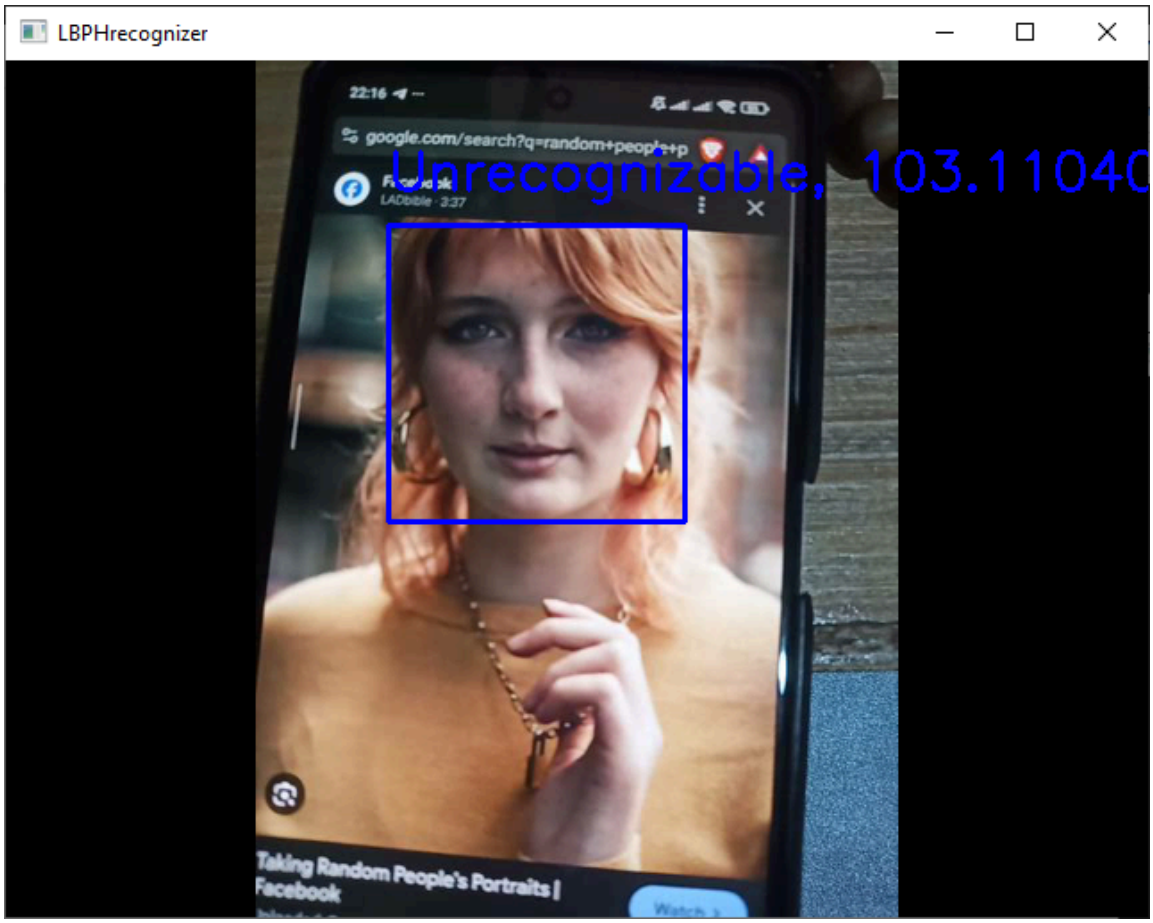

LBPHrecognizer
Ery, 92.01939597882 86

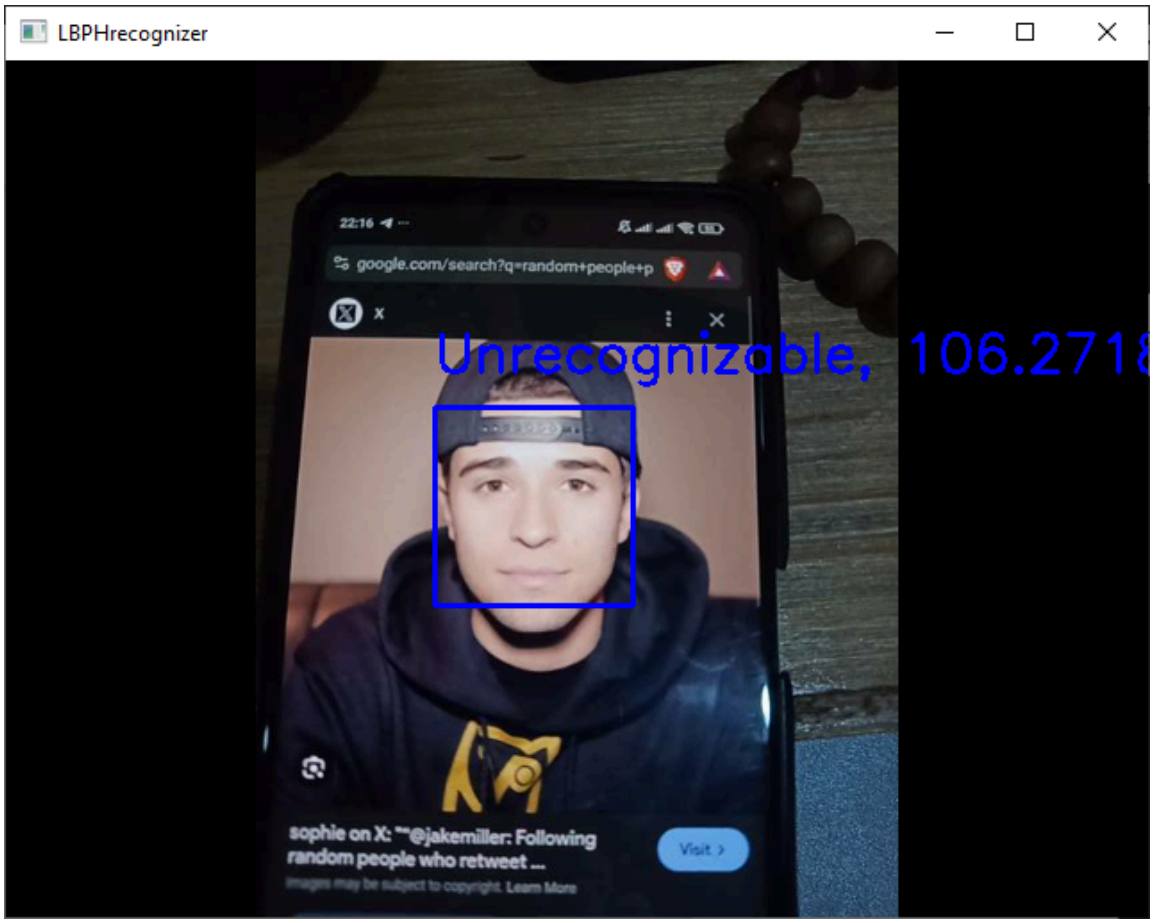

LBPHrecognizer
Ery, 87.3775014164894

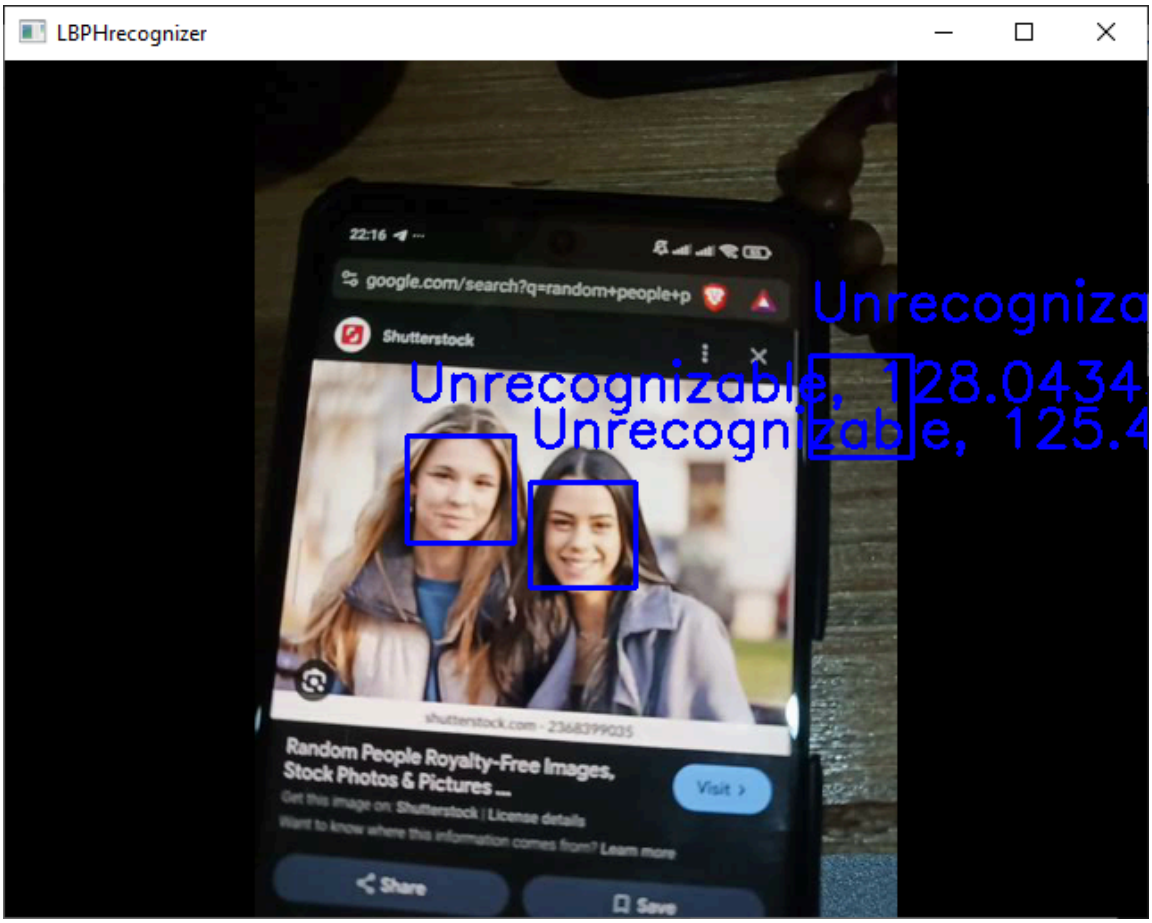

LBPHrecognizer
Ery, 86.86706652689783
LeBron James
Story

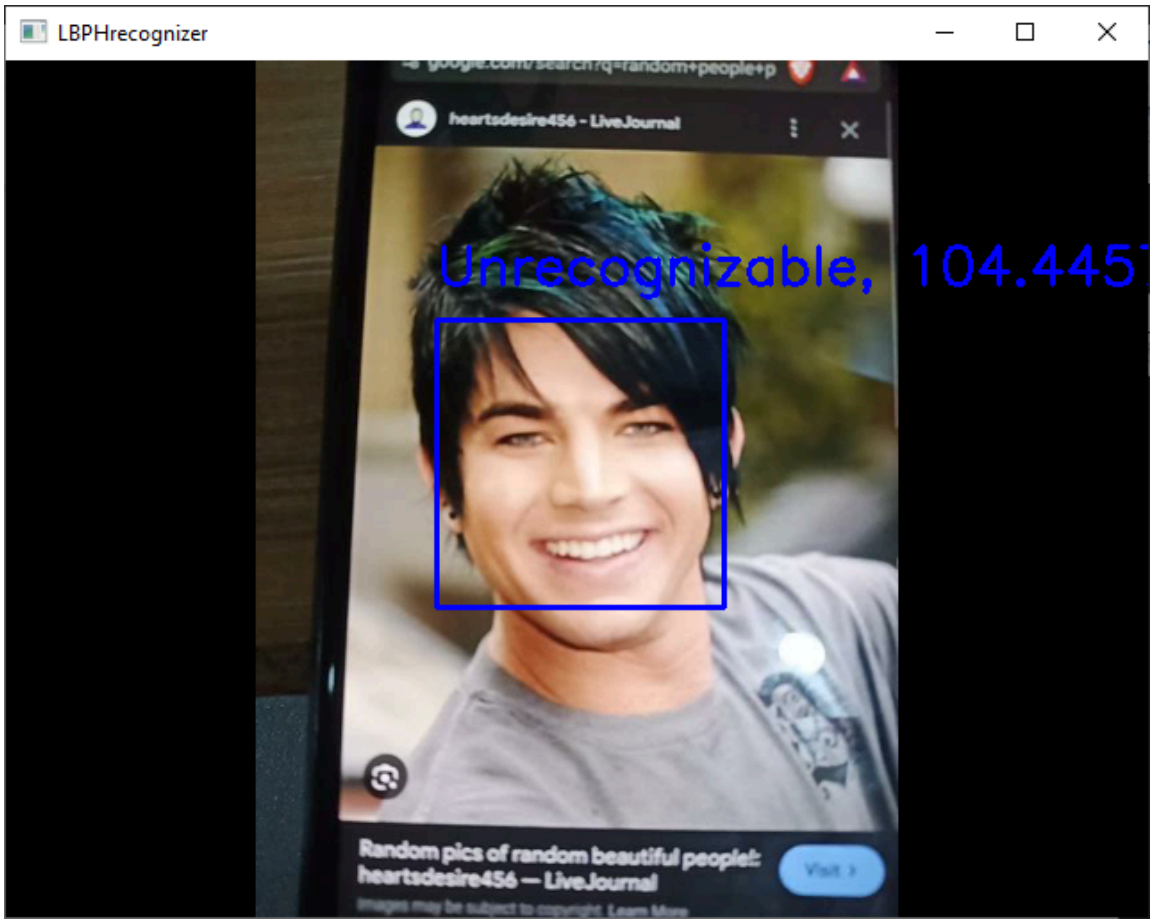

LBPHrecognizer
Unrecognizable, 116.6940
iStock
3,100+ Random Faces Of People Stock Photos, Pictures & Royalty ...
Share
Save

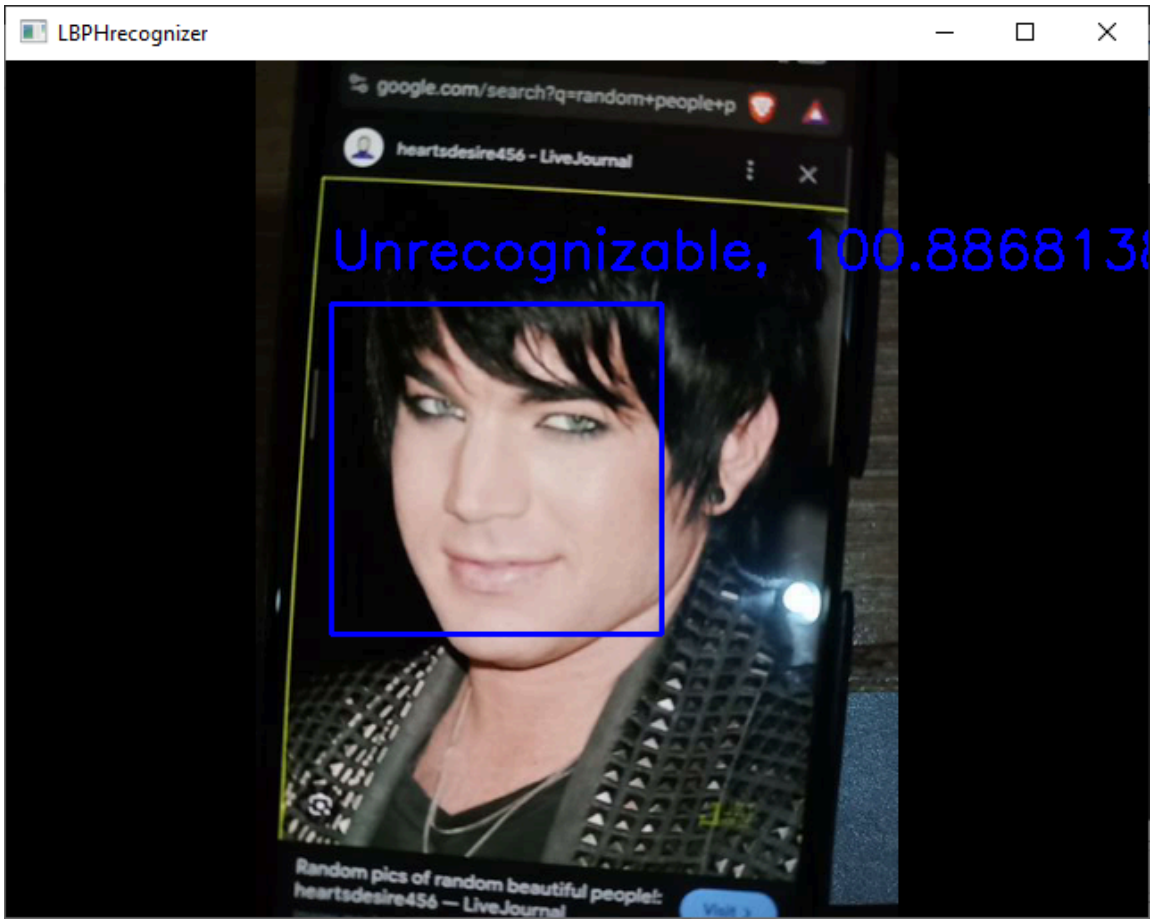

LBPHrecognizer
Unrecognizable, 91.21755
I've Photographed Random People From Random Places | Bored Panda
Visit ›

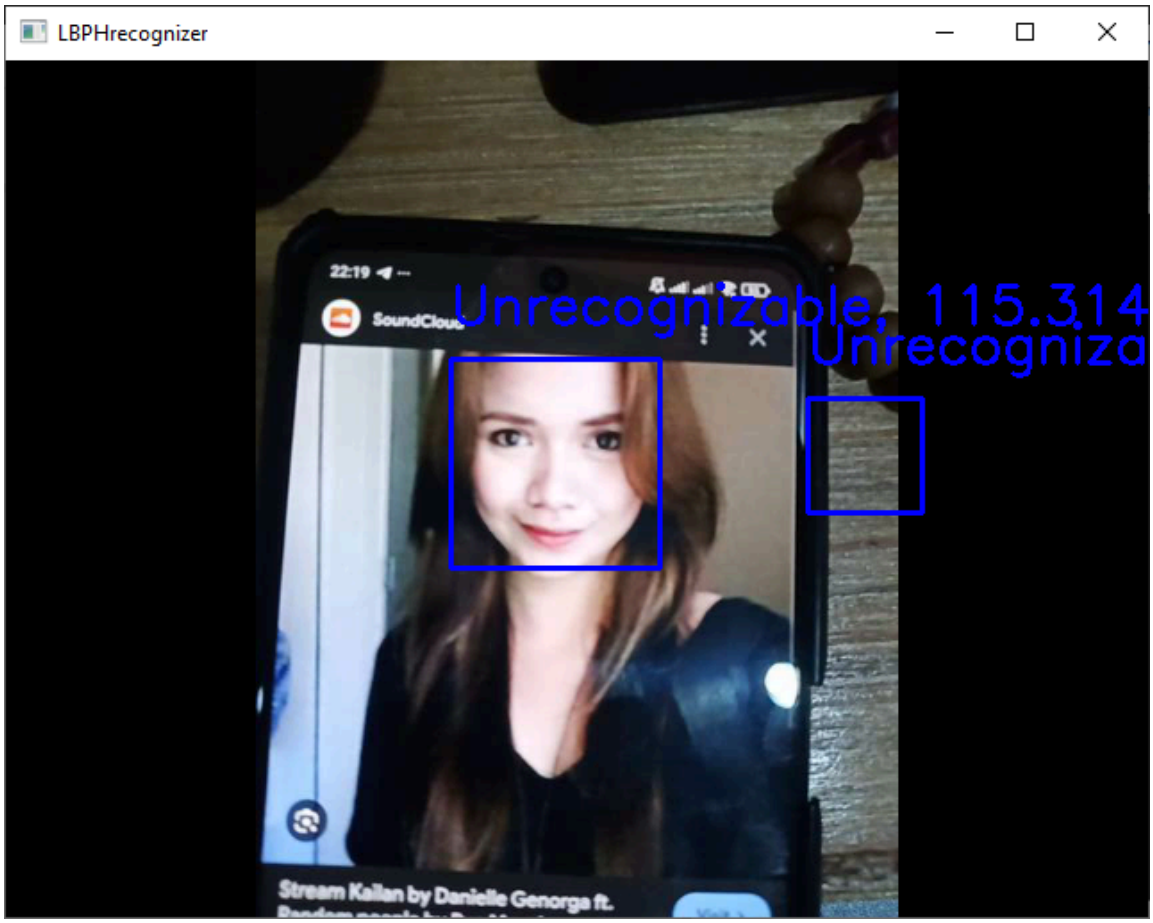

LBPHrecognizer
Unrecognizable, 103.11040
Taking Random People's Portraits | Facebook

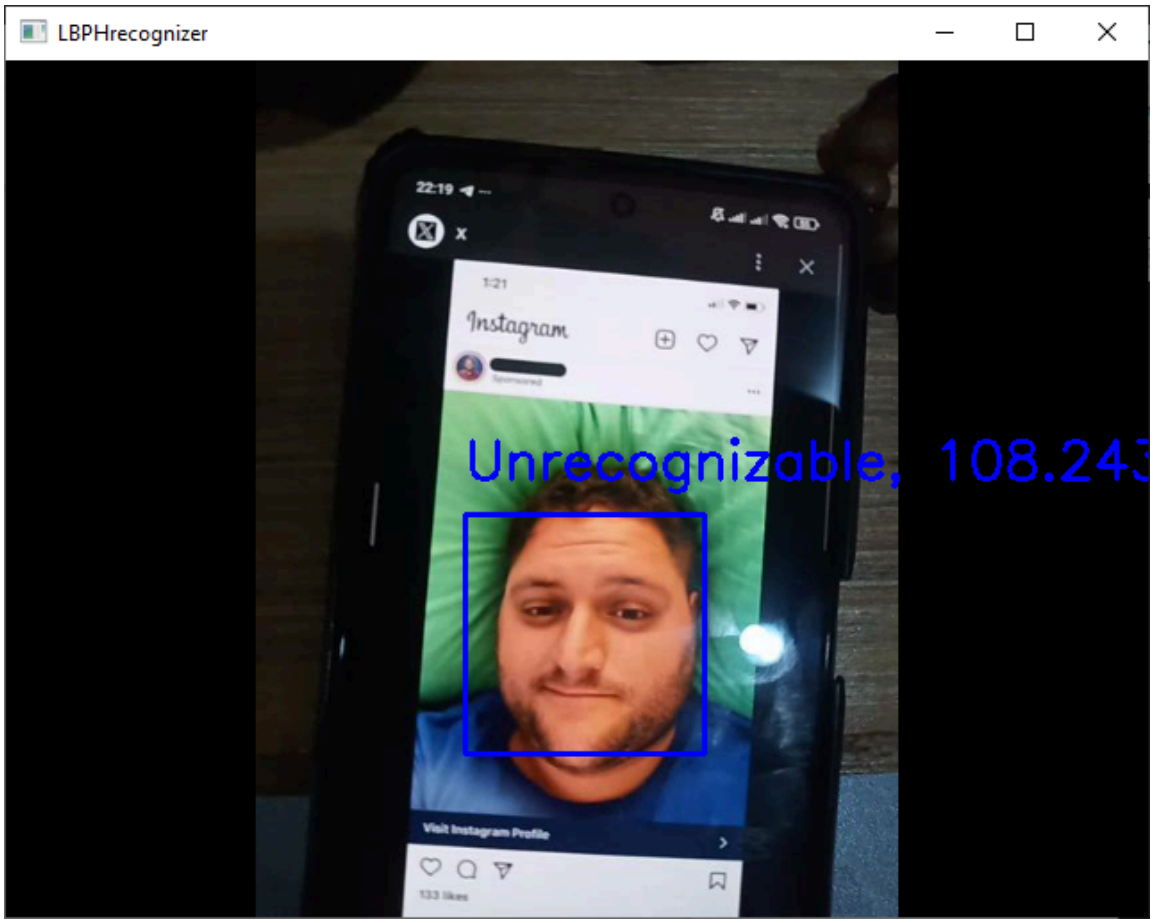

LBPHrecognizer
22:16
google.com/search?q=random+people+p
x
Unrecognizable, 106.2718
sophie on X: ""@jakemiller: Following random people who retweet ...
Visit ›

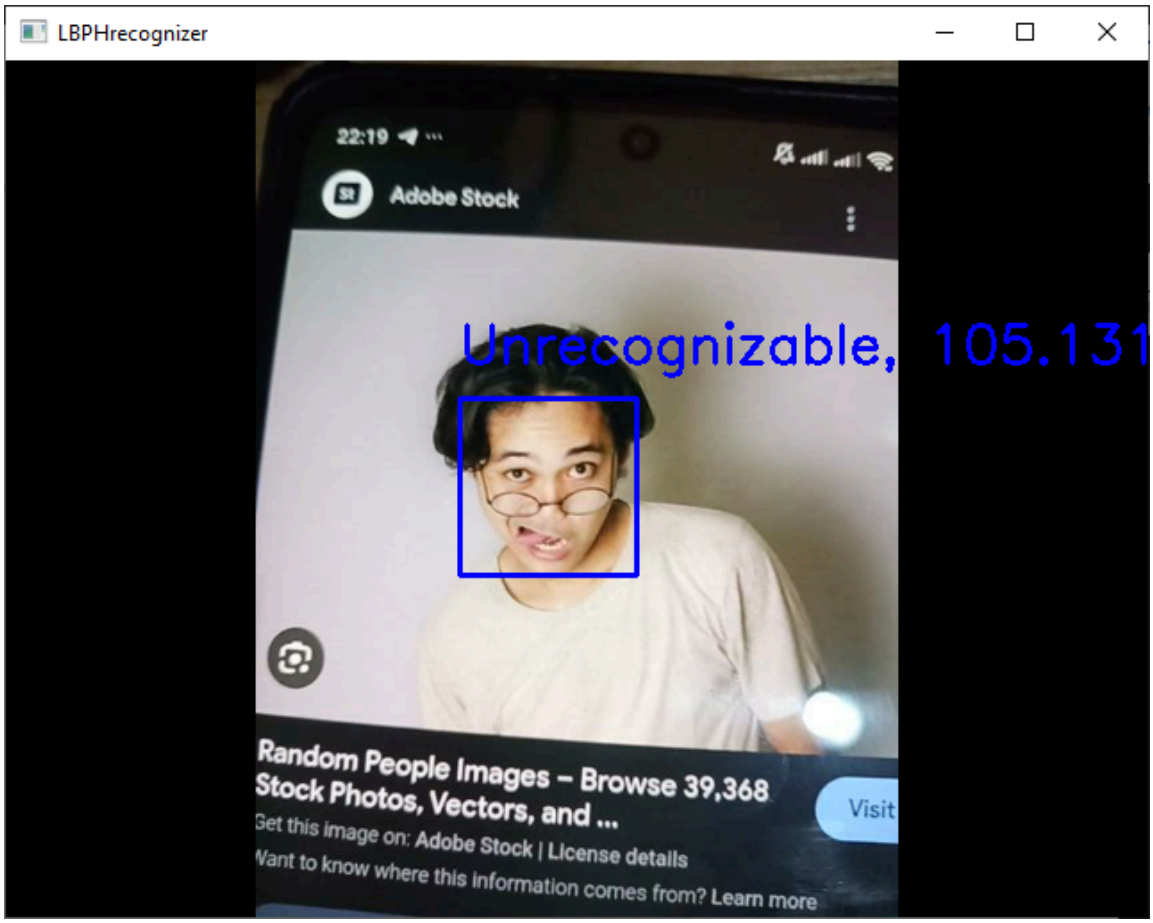

LBPHrecognizer
Unrecognizable, 128.0434
Unrecognizable, 125.4
22:16
google.com/search?q=random+people+p
Shutterstock
Random People Royalty-Free Images, Stock Photos & Pictures ...
Visit
Share
Save

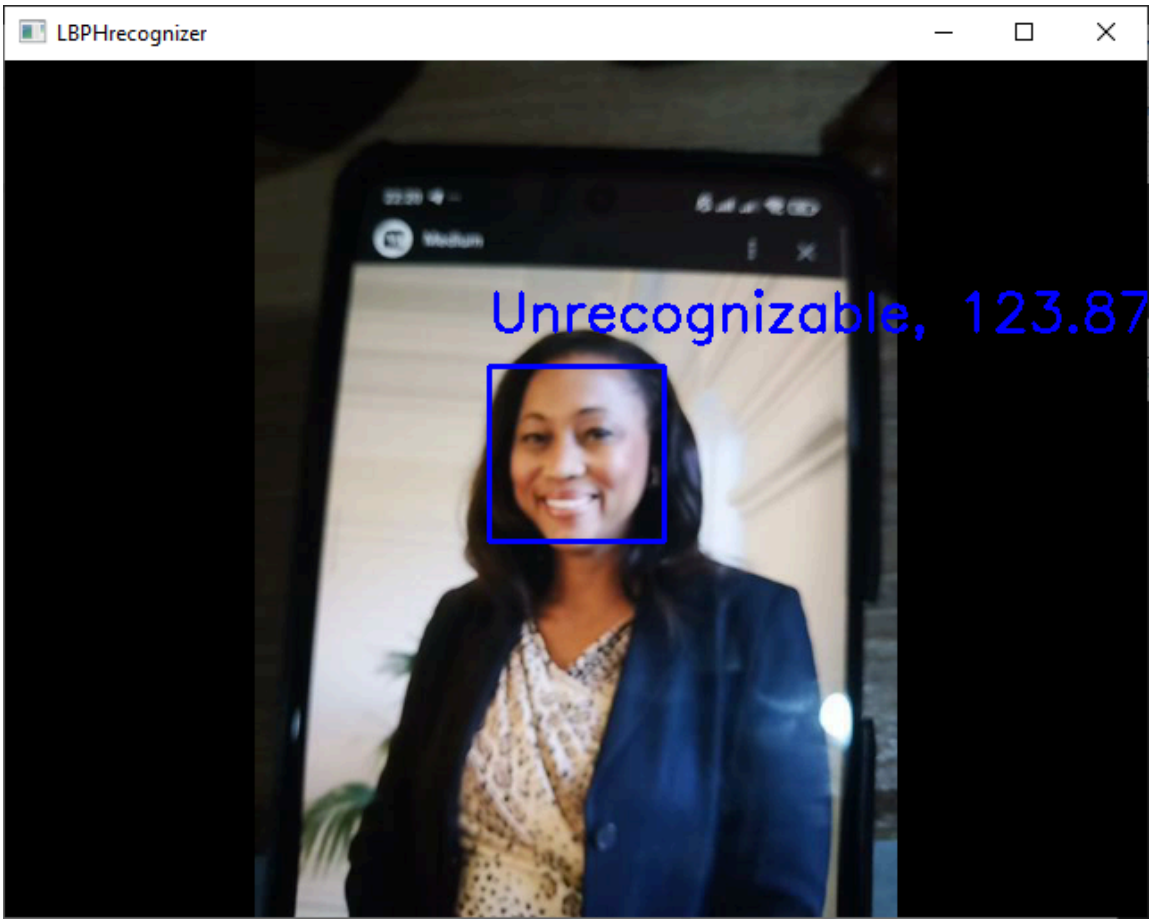

LBPHrecognizer
heartsdesire456 - LiveJournal
Unrecognizable, 104.445
Random pics of random beautiful people!: heartsdesire456 — LiveJournal
Visit


LBPHrecognizer
google.com/search?q=random+people+p
heartsdesire456 - LiveJournal
Unrecognizable, 100.886813
Random pics of random beautiful people!:
heartsdesire456 — LiveJournal


LBPHrecognizer
Unrecognizable, 115.314
Unrecogniza
Stream Kailan by Danielle Genorga ft.


LBPHrecognizer
Unrecognizable, 108.243
Instagram
Visit Instagram Profile


LBPHrecognizer
22:19
Adobe Stock
Unrecognizable, 105.131
Random People Images – Browse 39,368
Stock Photos, Vectors, and ...
Visit
Get this image on: Adobe Stock | License details
Want to know where this information comes from? Learn more



The evaluation of the performance of the face recognition models is that the LBPH and the eigen face performed better than the fisher algorithm because it got more misclassifications than the other two algorithms, I also observed that in my model it is harding to recognize what face to classify.

5. Summary, Conclusions and Lessons Learned

In this Activity I have learned how to recognize or classify faces with 3 different algorithm models that is used in opencv package, those are the eigenface, fisherface, and lgph, but before the training begin we need to first quantify the images so that the computer will understand the image we will be inputting, after quantifying the data we need to write in into csv file then all of the features and the label will be put to the model

---