

# Face Detection and YOLO v3 Object Detection on Windows

Draft  
2022-06-28

# Install Python 3.6 or above

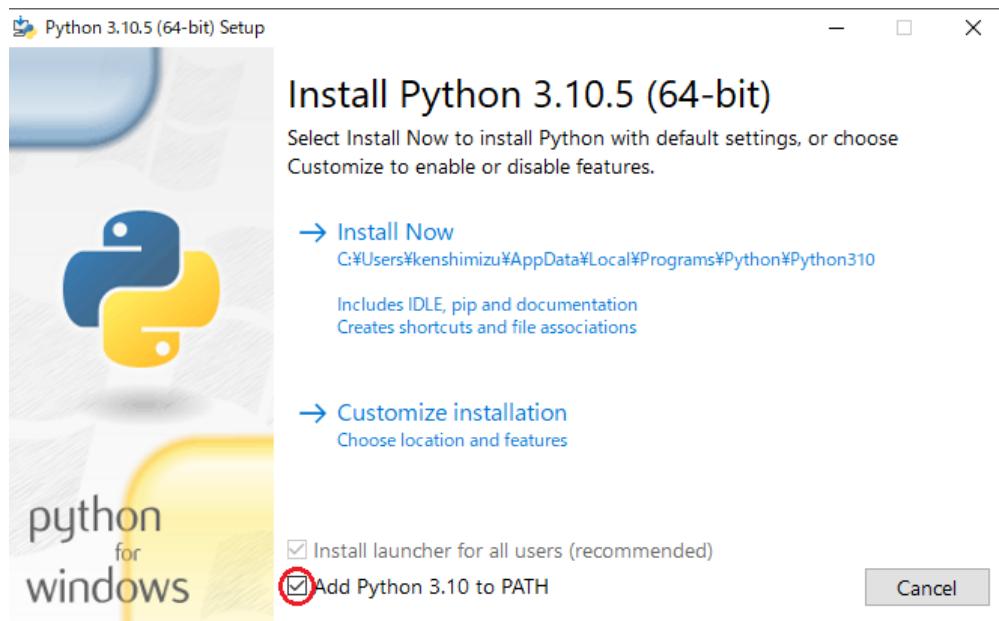
In my case, I choose latest Python 3.10.5

1. Go to <https://www.python.org/downloads/windows/> or <https://www.python.org/downloads/release/python-3105/>
2. Click Download Windows installer (64-bit) and execute it.

The screenshot shows the Python 3.10.5 (64-bit) download page. At the top, there is a navigation bar with various links. Below it, a note about quarks is displayed. A link to the full changelog is present. The main section is titled "Files" and contains a table with the following data:

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		d87193c077541e22f892ff1353fac76c	25628472	SIG
XZ compressed source tarball	Source release		f05727cb3489aa93cd57eb561c16747b	19361320	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	cdc24c5a91477ae446689711c53aa72	40430804	SIG
Windows embeddable package (32-bit)	Windows		86be4156e8a5d5c9added8aab2bc83d1	7596969	SIG
Windows embeddable package (64-bit)	Windows		d97e3c0c7a19db2c5019f5334bcb0b19	8558134	SIG
Windows help file	Windows		43c924ac07daed65acd85596eed1e33	9319556	SIG
Windows installer (32-bit)	Windows		eb59401a8da40051ec3b429897ae1203	27478768	SIG
Windows installer (64-bit)	Windows	Recommended	9a99ae597902b70b1273e88cc0d41abd	28637720	SIG

3. Select “Add Python x.x to PATH” in the Install Python Setup window, then select “Install Now”.



# Install opencv-python (OpenCV)

1. Open command prompt and execute following.

```
□ コマンドプロンプト  
Microsoft Windows [Version 10.0.19043.1706]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\kenshimizu>pip.exe install opencv-python
```

# Download model

1. Save following file.

```
# download from  
https://raw.githubusercontent.com/kipr/opencv/master/data/haarcascades/haar  
cascade\_frontalface\_default.xml
```

# Face Detection

1. Copy, paste the following script, and save as facedetect.py

```
import numpy as np
import cv2

# download from
#https://raw.githubusercontent.com/kipr/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Set up Video Camera
# 0 is the default camera.
cap = cv2.VideoCapture(0)

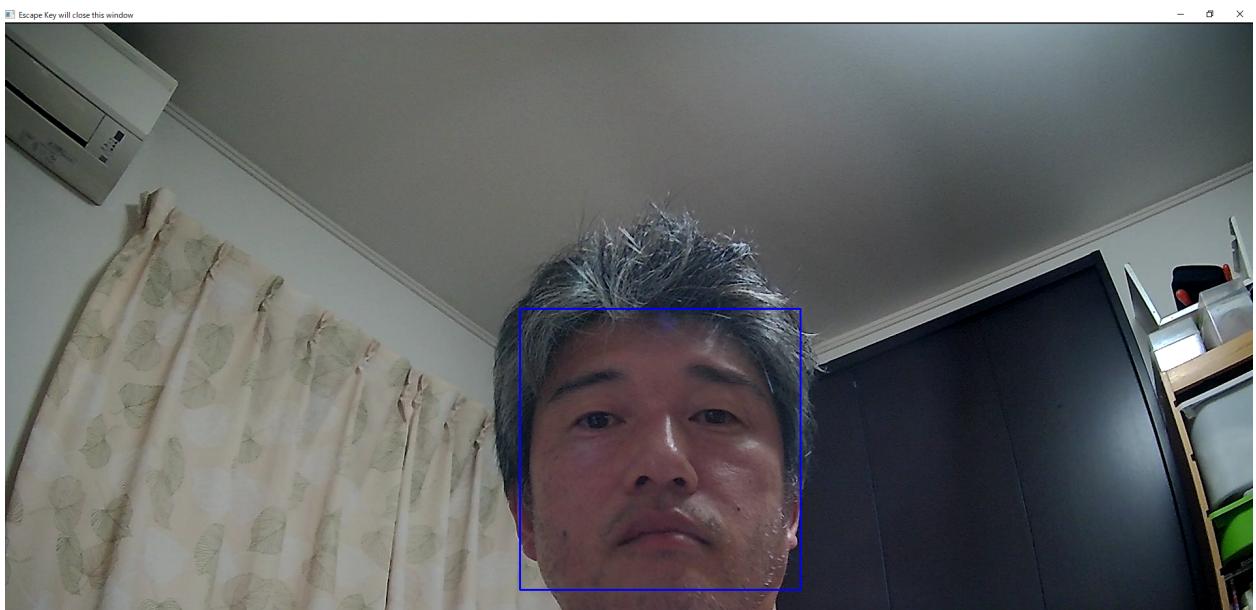
while True:
    # read picture from camera
    ret, img = cap.read()
    # convert color into gray.
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # detect face in the gray image
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)
    # draw face rectangle on the original image
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
    # show image in the window
    cv2.imshow('Escape Key will close this window', img)
    key = cv2.waitKey(1)
    if key == 27: # break if escape key
        break

cv2.destroyAllWindows()
```

## 2. Execute script

```
選択コマンドプロンプト - python.exe facedetect.py
Microsoft Windows [Version 10.0.19043.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kenshimizu>python.exe facedetect.py
```



# Object Detection

## 1. Download following files

Yolov3.cfg

<https://github.com/pjreddie/darknet/blob/master/cfg/yolov3.cfg>

Yolov3.weights

<https://pjreddie.com/media/files/yolov3.weights>

Coco.names

[https://opencv-tutorial.readthedocs.io/en/latest/\\_downloads/a9fb13cbea0745f3d11da9017d1b8467/coco.names](https://opencv-tutorial.readthedocs.io/en/latest/_downloads/a9fb13cbea0745f3d11da9017d1b8467/coco.names)

## 2. Copy, paste the following script, and save as objectdetect.py

```
import cv2
import numpy as np
import time

conf = 0.5

# Load coco.names and allocate color respectively.
classes = open('coco.names').read().strip().split('\n')
np.random.seed(42)
colors = np.random.randint(0, 255, size=(len(classes), 3), dtype='uint8')

# Give the configuration and weight files for the model and load the network.
net = cv2.dnn.readNetFromDarknet('yolov3.cfg', 'yolov3.weights')
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
# net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)

ln = net.getLayerNames()
ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]

# Set up Video Camera
cap = cv2.VideoCapture(0)

while True:
    # read picture from camera
    ret, img = cap.read()

    # construct a blob from the image
    blob = cv2.dnn.blobFromImage(img, 1/255.0, (416, 416), swapRB=True, crop=False)
    r = blob[0, 0, :, :]

    # set blob
    net.setInput(blob)
```

```

t0 = time.time()
outputs = net.forward(ln)
t = time.time()

boxes = []
confidences = []
classIDs = []
H, W = img.shape[:2]

for output in outputs:
    for detection in output:
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]
        if(confidence > conf):
            x, y, w, h = detection[:4] * np.array([W, H, W, H])
            p0 = int(x - w//2), int(y - h//2)
            p1 = int(x + w//2), int(y + h//2)
            boxes.append([*p0, int(w), int(h)])
            confidences.append(float(confidence))
            classIDs.append(classID)

"""
NMS Non-Maximum Suppression:
Given all scored regions in an image, we apply a greedy non-maximum suppression (for each class independently)
that rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold.
"""

indices = cv2.dnn.NMSBoxes(boxes, confidences, conf, conf-0.1)
if len(indices) > 0:
    for i in indices.flatten():
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])
        color = [int(c) for c in colors[classIDs[i]]]
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        text = "{}: {:.4f}".format(classes[classIDs[i]], confidences[i])
        cv2.putText(img, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)

# show image in the window
cv2.imshow('Escape Key will close this window', img)
key = cv2.waitKey(1) # wait 1 sec for preventing broken window.
if key == 27: # break if escape key
    break

cv2.destroyAllWindows()

```

### 3. Execute script

```
□ コマンドプロンプト
Microsoft Windows [Version 10.0.19043.1706]
(c) Microsoft Corporation. All rights reserved.

C:\$Users\$kenshimizu>python.exe objectdetect.py
```

