In [2]:

```
import numpy as np
import cv2
#import pickle
import import_ipynb
```

In [3]:

```
frameWidth= 640         # CAMERA RESOLUTION
frameHeight = 480
brightness = 180
threshold = 0.25        # PROBABLITY THRESHOLD
font = cv2.FONT_HERSHEY_SIMPLEX
```

In [4]:

```
# SETUP THE VIDEO CAMERA
cap = cv2.VideoCapture(0)
cap.set(4, frameWidth)
cap.set(5, frameHeight)
cap.set(15, brightness)
```

Out[4]:

```
True
```

In [5]:

```
# IMPORT THE TRAINED MODEL
#pickle_in=open("model_trained.p","rb")  ## rb = READ BYTE
#model=pickle.load(pickle_in)
import url_phishing1 as model
```

In [7]:

```
def grayscale(img):
    img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    return img
def equalize(img):
    img =cv2.equalizeHist(img)
    return img
def preprocessing(img):
    img = grayscale(img)
    img = equalize(img)
    img = img/255
    return img
def getCalssName(classNo):
    if   classNo == 0: return 'Adidas Logo'
    elif classNo == 1: return 'Adobe Logo'
    elif classNo == 2: return 'Airtel Logo'
    elif classNo == 3: return 'Amazon Logo'
    elif classNo == 4: return 'Android Logo'
    elif classNo == 5: return 'Apple Logo'
    elif classNo == 6: return 'Asus Logo'
    elif classNo == 7: return 'BMW Logo'
    elif classNo == 8: return 'Calvin-Klein Logo'
    elif classNo == 9: return 'Dell Logo'
    elif classNo == 10: return 'Facebook Logo'

    else : return 'This might be a phishing website'

while True:

# READ IMAGE
    cap = cv2.VideoCapture(0)
    success, imgOrignal = cap.read()
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-7-77ae52fe4e7f> in <module>
     29 # READ IMAGE
     30     cap = cv2.VideoCapture(0)
---> 31     success, imgOrignal = cap.read()

KeyboardInterrupt: 
```

In [ ]:

```
# PROCESS IMAGE
img = np.asarray(imgOrignal)
img = cv2.resize(img, (32, 32))
img = preprocessing(img)
cv2.imshow("Processed Image", img)
img = img.reshape(1, 32, 32, 1)
cv2.putText(imgOrignal, "CLASS: " , (20, 35), font, 0.75, (0, 0, 255), 2, cv2.LINE_AA)
cv2.putText(imgOrignal, "PROBABILITY: ", (20, 75), font, 0.75, (0, 0, 255), 2, cv2.LINE_AA)
# PREDICT IMAGE
predictions = model.predict(img)
classIndex = model.predict_classes(img)
probabilityValue =np.amax(predictions)
if probabilityValue > threshold:
#print(getCalssName(classIndex))
    cv2.putText(imgOrignal,str(classIndex)+" "+str(getCalssName(classIndex)), (120, 35), font, 0.75, (0, 0, 2
    cv2.putText(imgOrignal, str(round(probabilityValue*100,2) )+"%", (180, 75), font, 0.75, (0, 0, 255), 2, c
    cv2.imshow("Result", imgOrignal)

if cv2.waitKey(1) and 0xFF == ord('q'):
    break
```

In [ ]: