Create Dataset for Afghan Sign Langauge

Import Required Modules

In [1]:
```
import cv2
import numpy as np
import time
import os
```

First way to Capture the Image via OpenCV

In [2]:
```
def write_image_into_folder(destination1, destination2, ech_fldr, typ):
    """
        DOCSTR:: this function gets the images from WebCam of PC and
        write it to the certain folder for how times that it's called!
    """

    # Letters Dictionary
    letters = {
        'ا': "Alef",
        'آ': "Alef A",
        'ب': "Beh",
        'پ': "Peh",
        'ت': "Teh",
        'ث': "Seh",
        'ج': "Gem",
        'ح': "Heh",
        'خ': "Kheh",
        'د': "Dal",
        'ذ': "Zal",
        'ر': "Reh",
        'ز': "Zeh",
        'س': "Sen",
        'ش': "Shen",
        'ص': "Saad",
        'ض': "Zaad",
        'ط': "Tooy",
        'ظ': "Zooy",
        'ع': "Ayen",
        'غ': "Ghayn",
        'ف': "Pheh",
        'ق': "Qaaf",
        'ک': "Kaaf",
        'گ': "Ghaf",
        'ل': "Laam",
        'م': "Meem",
        'ن': "Noon",
        'ه': "Heeh",
        'و': "Woww",
        'ء': "Hamza",
        'ی': "Yaa",
    }

    # Image size
    IMAGE_SIZE = 96

    # ROI(Region of Interest) coordinates or Window coordinates!
```

```
left, top, right, bottom = 50, 300, 300, 550
# (350, 50), (600, 300), (top, left), (bottom, right)
camera = cv2.VideoCapture(0)

i = 1
while True:
    (t, frame) = camera.read()
    # Flip the frame, because the first situation it doesn't have the mirror view, now it has.
    frame = cv2.flip(frame, 1)

    # Get the ROI
    roi = frame[left:top, right:bottom]

    # Convert the roi to grayscale and blur it
    gray_img = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    gray_img = cv2.GaussianBlur(gray_img, (7, 7), 0)
    # Resize the image
    gray_img = cv2.resize(gray_img, (IMAGE_SIZE, IMAGE_SIZE))

    # Write image file to the destination for 1300 times!
    cv2.imwrite("%s/%d.jpeg"%(destination1, i), gray_img)
    cv2.imwrite("%s/%d.jpeg"%(destination2, i), gray_img)
    cv2.putText(frame, "AFG Sign Langauge", (200, 30), cv2.FONT_HERSHEY_SCRIPT_COMPLEX, 1, (255, 230, 155), 1)

    if typ == "n":
        cv2.putText(frame, "Current Gesture --> " + str(ech_fldr), (15, 70), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 0), 1
        cv2.rectangle(frame, (10, 50), (260, 80), (97, 125, 255), 2)
    else:
        cv2.putText(frame, "Current Gesture --> " + letters[str(ech_fldr)], (15, 70), cv2.FONT_HERSHEY_PLAIN, 1, (255,
        cv2.rectangle(frame, (10, 50), (260, 80), (97, 125, 255), 2)

    print("Image #%d"%(i))
    i += 1
    if i > 1300:
        break

    # Draw the hand position or segmantation!
    cv2.rectangle(frame, (top, left), (bottom, right), (195, 210, 180), 2)

    cv2.imshow("Video Feed 1", gray_img)
    cv2.imshow("Video Feed", frame)

    # wait for get the keypress by user!
    keypress = cv2.waitKey(1)

    # If the user pressed the Esc key, program has to exit.
```

```
        if keypress == 27:
            break

    # Free up the memory
    camera.release()
    cv2.destroyAllWindows()

def useless_images(train_folder, test_folder):
    """
        DOCSTR:: Remove the garbeg images from folder
    """
    for i in range(1, 301):
        os.remove("%s/%d.jpeg"%(train_folder, i))
        os.remove("%s/%d.jpeg"%(test_folder,i))
```

Second Way to Capture Image Via OpenCV

In [3]:
```
def letters(destination1, destination2, img, ech_dir):
    """
        DOCSTR:: Get the how number of images do you want.
    """

    interrupt = cv2.waitKey(10)
    if interrupt & 0xFF == 27:  # break the program when the Esc key has pressed!
        return True

    if interrupt & 0xFF == ord(str(ech_dir)):
        cv2.imwrite(destination1, img)
        cv2.imwrite(destination2, img)

    interrupt = cv2.waitKey(1)
    if interrupt & 0xFF == ord('q'):          # esc key
        return True


def numbers(destination1, destination2, img, ech_dir):
    """
        DOCSTR:: Get the how number of images do you want.
    """

    interrupt = cv2.waitKey(10)
    if interrupt & 0xFF == 27: # break the program when the Esc key has pressed!
        return True

    if interrupt & 0xFF == ord(str(ech_dir)):
        cv2.imwrite(destination1, img)
        cv2.imwrite(destination2, img)

    interrupt = cv2.waitKey(1)
    if interrupt & 0xFF == ord('q'): # esc key
        return True



def write_image_into_directories1(destination1, destination2, ech_dir, typ = "l"):
    """
        DOCSTR:: Take a snapshot from video stream from WebCam of PC
        then write it to the specific folder ...
    """

    print("""
    \n\n
    Alert!
```

```
Press the q key or Esc key for the ignore this Gesture and ready for the next Gesture ...
""")
directory = destination1
capture = cv2.VideoCapture(0)
# intterupt = -1
while True:
    _, frame = capture.read()

    # Simulating mirror view
    frame = cv2.flip(frame, 1)

    count_numbers = {  #change this according to the order you want
        '0': len(os.listdir(directory)),
        '1': len(os.listdir(directory)),
        '2': len(os.listdir(directory)),
        '3': len(os.listdir(directory)),
        '4': len(os.listdir(directory)),
        '5': len(os.listdir(directory)),
        '6': len(os.listdir(directory)),
        '7': len(os.listdir(directory)),
        '8': len(os.listdir(directory)),
        '9': len(os.listdir(directory)),
    }

    count_letters = {   # change this according to the order you want
        'ا': [len(os.listdir(directory)), ("a", "Alef")],
        'آ': [len(os.listdir(directory)), ("b", "Alef A")],
        'ب': [len(os.listdir(directory)), ("c", "Beh")],
        'پ': [len(os.listdir(directory)), ("d", "Peh")],
        'ت': [len(os.listdir(directory)), ("e", "Teh")],
        'ث': [len(os.listdir(directory)), ("f", "Seh")],
        'ج': [len(os.listdir(directory)), ("g", "Gem")],
        'ح': [len(os.listdir(directory)), ("j", "Heh")],
        'خ': [len(os.listdir(directory)), ("i", "Kheh")],
        'د': [len(os.listdir(directory)), ("j", "Dal")],
        'ذ': [len(os.listdir(directory)), ("k", "Zal")],
        'ر': [len(os.listdir(directory)), ("l", "Reh")],
        'ز': [len(os.listdir(directory)), ("m", "Zeh")],
        'س': [len(os.listdir(directory)), ("n", "Sen")],
        'ش': [len(os.listdir(directory)), ("o", "Shen")],
        'ص': [len(os.listdir(directory)), ("p", "Saad")],
        'ض': [len(os.listdir(directory)), ("q", "Zaad")],
        'ط': [len(os.listdir(directory)), ("r", "Tooy")],
        'ظ': [len(os.listdir(directory)), ("s", "Zooy")],
        'ع': [len(os.listdir(directory)), ("t", "Ayen")],
        'غ': [len(os.listdir(directory)), ("u", "Ghayn")],
```

```
    'ف': [len(os.listdir(directory)), ("v", "Pheh")],
    'ق': [len(os.listdir(directory)), ("w", "Qaaf")],
    'ک': [len(os.listdir(directory)), ("x", "Kaaf")],
    'گ': [len(os.listdir(directory)), ("y", "Ghaf")],
    'ل': [len(os.listdir(directory)), ("z", "Laam")],
    'م': [len(os.listdir(directory)), ("1", "Meem")],
    'ن': [len(os.listdir(directory)), ("2", "Noon")],
    'ہ': [len(os.listdir(directory)), ("3", "Heeh")],
    'و': [len(os.listdir(directory)), ("4", "Woww")],
    'ء': [len(os.listdir(directory)), ("5", "Hamza")],
    'ی': [len(os.listdir(directory)), ("6", "Yaa")],
}

cv2.putText(frame,"Alert! Press the Esc key for the ignore this Gesture and ready for the next Gesture ... ",
           (15, 25), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 0), 1)

cv2.rectangle(frame, (10, 10), (frame.shape[1] - 10, 30), (0, 0, 255), 1)

# Printing the each part of count_numbers on the screen dynamically.
if typ == "l":
    cv2.putText(frame, str(count_letters[ech_dir][1][0]) + " -> " + str(count_letters[ech_dir][1][1]) + " : " + st
    # Draw a rectangle for count_numbers.
    cv2.rectangle(frame, (10, 40), (160, 80), (0, 255, 0), 1)
    # Destination of letters
    ltr_dest1 = destination1+'/'+str(count_letters[str(ech_dir)][0])+'.jpg'
    ltr_dest2 = destination2+'/'+str(count_letters[str(ech_dir)][0])+'.jpg'
else:
    cv2.putText(frame, str(ech_dir) + " : " + str(count_numbers[str(ech_dir)]), (15, 60), cv2.FONT_HERSHEY_PLAIN,
    # Draw a rectangle for count_numbers.
    cv2.rectangle(frame, (10, 40), (70, 80), (255, 0, 0), 1)
    # Destination of numbers
    num_dest1 = destination1+'/'+str(count_numbers[str(ech_dir)])+'.jpg'
    num_dest2 = destination2+'/'+str(count_numbers[str(ech_dir)])+'.jpg'

# Coordinates of the ROI
x1 = int(0.5*frame.shape[1])
y1 = 40
x2 = frame.shape[1]-10
y2 = int(0.5*frame.shape[1])

# Extracting the ROI
roi = frame[y1:y2, x1:x2]

# Drawing the ROI
cv2.rectangle(frame, (x1-1, y1-1), (x2+1, y2+1), (255,0,0) ,1)
```

```
        # convert image to grayscale image
        gray_img = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
        gray_img = cv2.GaussianBlur(gray_img, (7, 7), 0)

        cv2.imshow("Frame", frame)
        cv2.imshow("ROI", gray_img)

        if typ == "l":
            if letters(ltr_dest1, ltr_dest2, gray_img, count_letters[ech_dir][1][0]):
                return
            else:
                pass
        else:
            if numbers(num_dest1, num_dest2, gray_img, ech_dir):
                return
            else:
                pass

    capture.release()
    cv2.destroyAllWindows()
```

Basic Functions

In [*]:
```
def create_directory(path, typ = "l"):
    """
        DOCSTR:: Manage the directories will use in this Dataset of theses project "Afghan Sign Language"!
    """

    train_path = path[0]
    test_path = path[1]
    directories_ltrs = ["ی" ,'ه' ,'و' ,'ن' ,'م' ,'ل' ,'گ' ,'ک' ,'ق' ,'ف' ,'غ' ,'ع' ,'ظ' ,'ط' ,'ض' ,'ص' ,'ش' ,'س' ,'ز' ,'ر'
    directories_nmbr = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "0"]

    if typ == 'l':
        folders = directories_ltrs
    else:
        folders = directories_nmbr

    print("""
    Which method is better for you?
    1- [First Way for a larg number of images with a little quality]
        recommanded -> the project needs huge dataset(Letters) ....
    2- [Second way for a small number of images with high quality]
        recommanded -> the project needs small dataset(Numbers) ...

    """)

    input_way = int(input("Choose the option, [Just Number]: "))

    if input_way != 1 and input_way != 2:
        print("""
        Some errors occured!
        please delete the folders are created a few minutes ago, then run the program again ...\n
        """)
        return # stop the function

    for ech_dir in folders:
        os.mkdir(train_path + "/" + ech_dir)
        os.mkdir(test_path + "/" + ech_dir)
        if input_way == 1:
            write_image_into_folder(train_path + "/" + ech_dir, test_path + "/" + ech_dir, ech_dir, typ)
            try:
                useless_images(train_path + "/" + ech_dir, test_path + "/" + ech_dir)
            except:
                print("Some errors occured! ... ;( ")
            print("Next Gesture ... ")
            time.sleep(3) # waiting for 3 seconds
        else:
            write_image_into_directories1(train_path + "/" + ech_dir, test_path + "/" + ech_dir, ech_dir, typ)
```

```
            print("Next Gesture ... ")
            time.sleep(3)

    print("Operations successfully done ... ")


def basic_operation(option=""):
    """
        DOCSTR:: Create the Root Directory of Train & Test folders of dataset.
        return -> A list includes of path for train and test data ...
    """

    root_dir = input("Enter root direcoty: ")
    path = []
    if option == 'letters':
        try:
            os.mkdir(root_dir)
            os.mkdir(root_dir + "/Train")
            os.mkdir(root_dir + "/Test")
            path.append(root_dir + "/Train")
            path.append(root_dir + "/Test")
        except:
            print("Some errors occured! :( ")
    else:
        try:
            os.mkdir(root_dir)
            os.mkdir(root_dir + "/Train")
            os.mkdir(root_dir + "/Test")
            path.append(root_dir + "/Train")
            path.append(root_dir + "/Test")
        except:
            print("Some errors occured! :( ")

    return path

def selected_menu():
    """
        DOCSTR:: shows which option selected by user.
    """

    input_menu = int(input("Selecte the menu, [Just Number]: "))

    if input_menu == 1:
        create_directory(basic_operation(option = "letters"), typ = "l")
    elif input_menu == 2:
        create_directory(basic_operation(option = "numbers"), typ = "n")
```

```
    elif input_menu == 3:
        exit()
    else:
        print("You selected wrong option :(")
        print("Please use the above options....")
        selected_menu()

def menu():
    """
        DOCSTR:: Design the menu of program!
    """

    main_menu = """
    1- [Alphabets for AFG Sign Langauge]
    2- [Numbers for AFG Sign Langauge]
    3- [Exit the program]
    """
    print(main_menu)
    selected_menu()

menu()
```

```
    1- [Alphabets for AFG Sign Langauge]
    2- [Numbers for AFG Sign Langauge]
    3- [Exit the program]

Selecte the menu, [Just Number]: 1

Enter root direcoty:
```