DetectShush.py

```
import numpy as np
import cv2
import os
from os import listdir
from os.path import isfile, join
import sys

def detectShush(frame, location, ROI, cascade):
    mouths = cascade.detectMultiScale(ROI, 1.15, 3, 0, (20, 20))
    for (mx, my, mw, mh) in mouths:
        mx += location[0]
        my += location[1]
        cv2.rectangle(frame, (mx, my), (mx+mw, my+mh), (0, 0, 255), 2)
    return len(mouths) == 0

def detect(frame, faceCascade, mouthsCascade):
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

#    gray_frame = cv2.equalizeHist(gray_frame)
#    gray_frame = cv2.medianBlur(gray_frame, 5)

    faces = faceCascade.detectMultiScale(
                gray_frame, 1.15, 4, 0|cv2.CASCADE_SCALE_IMAGE, (40, 40))
    detected = 0
    for (x, y, w, h) in faces:
        # ROI for mouth
        x1 = x
        h2 = int(h/2)
        y1 = y + h2
        mouthROI = gray_frame[y1:y1+h2, x1:x1+w]

        if detectShush(frame, (x1, y1), mouthROI, mouthsCascade):
            detected += 1
            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        else:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
    return detected


def run_on_folder(cascade1, cascade2, folder):
    if(folder[-1] != "/"):
        folder = folder + "/"
    files =  [join(folder,f) for f in listdir(folder) if isfile(join(folder,f))]
    windowName = None
    totalCnt = 0
    for f in files:
        img = cv2.imread(f)
        if type(img) is np.ndarray:
            lCnt = detect(img, cascade1, cascade2)
            totalCnt += lCnt
            if windowName != None:
                cv2.destroyWindow(windowName)
```

```
53              windowName = f
54              cv2.namedWindow(windowName, cv2.WINDOW_AUTOSIZE)
55              cv2.imshow(windowName, img)
56              cv2.waitKey(0)
57      return totalCnt
58
59  def runonVideo(face_cascade, eyes_cascade):
60      videocapture = cv2.VideoCapture(0)
61      if not videocapture.isOpened():
62          print("Can't open default video camera!")
63          exit()
64
65      windowName = "Live Video"
66      showframe = True
67      while(showframe):
68          ret, frame = videocapture.read()
69
70          if not ret:
71              print("Can't capture frame")
72              break
73          detect(frame, face_cascade, eyes_cascade)
74          cv2.imshow(windowName, frame)
75          if cv2.waitKey(30) >= 0:
76              showframe = False
77
78      videocapture.release()
79      cv2.destroyAllWindows()
80
81
82  if __name__ == "__main__":
83      # check command line arguments: nothing or a folderpath
84      if len(sys.argv) != 1 and len(sys.argv) != 2:
85          print(sys.argv[0] + ": got " + len(sys.argv) - 1 +
86          "arguments. Expecting 0 or 1:[image-folder]")
87          exit()
88
89      # load pretrained cascades
90      face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalf
91      mouth_cascade = cv2.CascadeClassifier('Mouth.xml')
92
93      if(len(sys.argv) == 2): # one argument
94          folderName = sys.argv[1]
95          detections = run_on_folder(face_cascade, mouth_cascade, folderName)
96          print("Total of ", detections, "detections")
97      else: # no arguments
98          runonVideo(face_cascade, mouth_cascade)
```