IMPLEMENTATION OF PROGRAM TO FIND THE NOSE TIP FOR THE GIVEN INPUT IMAGE

K.B.SATHWIKA (211FA04008)
J.N.V.M.CHARAN (211FA04014)
D.VAMSI KRISHNA (211FA04026)
Sd. KARIMUN (211FA04038)

BATCH-03
2nd B.TECH,CSE BRANCH
VFSTR DEEMED TO BE UNIVERSTY

ABSTRACT:

In this we used pre-trained facial detection and pre-trained facial landmark to identify the Face region and nose tip respectively in person's whole body.

INTRODUCTION:

Pre-processing : Load the input image and convert it to grayscale. Apply any necessary image pre-processing Techniques, such as noise reduction or contrast enhancement to improve the quality of the image.

Face detection : utilize a face detection algorithm, such as Haar Cascade or a deep learning-based method,to detect the face in image.

Once the face is detected, extract the region of interest (ROI) corresponding to the face for further processing.

Facial landmarks detection:

Employ a facial landmarks detection algorithm, such as the shape predictor from dlib or a convolutional neural network (CNN)-based method, to identify key facial landmarks.

These landmarks typically include points corresponding to the eyes, nose, mouth, and other facial features.

Nose tip identification:

Locate the specific landmark corresponding to the tip of the nose among the detected facial landmarks.

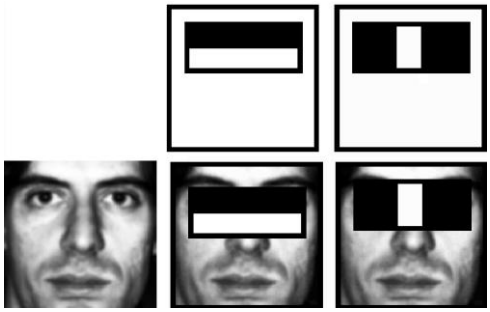
This landmark is usually denoted by an index or a specific label in the facial landmarks model.

Retrieve the coordinates of the nose tip landmark.

Refinement (optional):

Apply any necessary refinement techniques to improve the accuracy of the nose tip estimation.

For example, you can use geometric constraints or image processing techniques to validate the nose tip's position or further refine its coordinates.



Source:

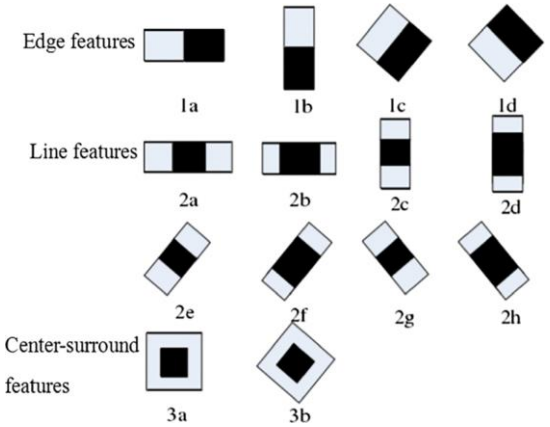
https://www.researchgate.net/profile/Subhashree-Subudhi/publication/303144203/figure/fig1/AS:565826378256385@1511914929667/Haar-like-feature-detector.png

ALGORITHM:

- Load the input image of a person's whole body and convert it to grayscale.
- Apply a face detection algorithm (such as Haar cascade classifier) to find the bounding box of the face region in the image.
- Crop the face region from the image and resize it to a fixed size (such as 64x64 pixels).
- Compute the multi-angle energy (ME) for each pixel in the face region, which is defined as the sum of absolute differences between the pixel value and its eight neighbours along different directions.
- Find the pixel with the maximum ME value, which is a candidate for the nose tip.
- Apply a coarse-to-fine strategy to refine the nose tip location by using a genetic algorithm that searches for the optimal ME value in a smaller neighbourhood around the candidate.
- Return the coordinates of the nose tip in the face region.

SOURCE CODE :

```
import cv2
import numpy as np

# Load image
img =
cv2.imread('C:\Users\hp\OneDrive\Pictures\Saved
Pictures\IMG_20191007_105550_986.jpg')

# Convert to grayscale
gray = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)

# Load Haar cascade for nose
nose_cascade =
cv2.CascadeClassifier('haarcascade_mcs_nose.xml')

# Detect nose tip
nose = nose_cascade.detectMultiScale(gray, 1.3, 5)

# Crop image around nose tip
for (x,y,w,h) in nose:
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]

# Display cropped image and nose tip coordinates
cv2.imshow('Cropped Image', roi_color)
print('Nose tip coordinates:', x+w//2, y+h//2)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



Source:

https://www.researchgate.net/publication/328490787/figure/fig3/AS:962379286327343@1606460510770/Feature-prototypes-of-haar-like-features.png

CONCLUSION:

To summarize, using a Haar cascade classifier for nose detection can be challenging due to the complexity and variability of nose shapes and appearances. Haar cascades are more commonly used for face detection, where they perform well. However, detecting specific facial features like the nose with a Haar cascade alone may provide accurate results.