EYE DETECTION

**OBJECTIVE**

The Objective of our project is to detect the position of human eyes from a given set of random input images of human face .

**METHODOLOGY**

At first we generate all possible Eye Candidates purely based

on the fact of corner points. Corners in image are detected using FAST algorithm.

Once we get set of Eye Candidates, we pass it to Pattern Recognition Neural Network which

recognizes whether the given Eye Candidate is Left Eye, Right Eye or Not an Eye.

Algorithm

% DESCRIPTION: Algorithm to detect Left and Right Eye.

% INPUT: RGB image

% OUTPUT: X and Y coordinates of Left and Right Eye.

--------------------------------------------------------------------------------------------------------------------------

1. IF trained Pattern Recognition Network does not exist,

Then,

a. Prepare training data.

b. Create a Pattern Recognition Network.

c. Train Pattern Recognition Network using prepared training data.

ELSE,

a. Load trained Pattern Recognition Network.

ENDIF

2. Find width and height of given input RGB image.

3. Transform image from RGB color space to LAB color space.

Reason:

Eye detection is efficient in LAB color model compare to RGM color model.

4. Use 'L' component of LAB image for further processing, where 'L' stands for lightness

contrast.

Reason:

As Eye dot corresponds to brightest spot in face image, hence using 'L' component in

further algorithm returns better results.

5. Detect corner points in image using FAST algorithm.

6. Pick first 50 strongest corner points as possible Eye Candidates.

7. FOR every Eye Candidate (as specified by 'EyeCandidates'),

a. Get X and Y coordinates of Eye Candidate under inspection.

b. Crop proportional region surrounding Eye Candidate location,

Where,

=>X1: X coordinate of top-left corner of image segment to be cropped,i.e. X - (Width/10)

=>Y1: X coordinate of top-left corner of image segment to be cropped, i.e. Y - (Height/20) =>Width/5: Width of image segment to be cropped =>Height/10: Height of image segment to be croppedc. Preprocess cropped Eye Candidate image segment.d. Feed preprocessed Eye Candidate image segment to Pattern Recognition Networkand predict whether it is Left Eye, Right Eye or Not an Eye.e. Find which output class among 3 (i.e. Left Eye, Right Eye and Not an Eye) hashighest probability and accordingly record corresponding output class.f. Store predicted Left Eye and Right Eye probabilities for Eye Candidate imagesegment under inspection.

ENDFOR

Note: At the end of step, there can be multiple Left Eye or Right Eye output class

predictions. So, next step will pick exact Left Eye and Right Eye candidate among many

possible candidates.

8. Locate Left Eye using below steps,

a. First check which Eye Candidate predicted as a ‘Left Eye’ has highest Left Eye

probability and mark it as final expected Left Eye Candidate.

b. IF no Left Eye Candidate is found,

Then,

I. Pick an Eye Candidate having highest Left Eye probability among all

possible Eye Candidates (even if it is not predicted as ‘Left Eye’) and mark

it as final expected Left Eye Candidate.

ENDIF

c. IF still no Left Eye Candidate is found,

Then,

I. Set default location for Left Eye Candidate.

ENDIF

9. Locate Right Eye using below steps,

a. First check which Eye Candidate predicted as a ‘Right Eye’ has highest Right Eye

probability and mark it as final expected Right Eye Candidate.

b. IF no Right Eye Candidate is found,

Then,

II. Pick an Eye Candidate having highest Right Eye probability among all

possible Eye Candidates (even if it is not predicted as ‘Right Eye’) and

mark it as final expected Right Eye Candidate.

ENDIF

c. IF still no Right Eye Candidate is found,

Then,

II. Set default location for Right Eye Candidate.

ENDIF

10. Set X and Y coordinates for Left Eye and Right Eye using locations found as described

Above.

11. Stop.

**FUNCTIONS USED**

Image Preprocessing for Neural Network

Training data and Eye Candidate segments are preprocessed before feeding it to Neural

Network.

Preprocessing steps are as below:

1. Transform image from RGB color space into HSV color space.

2. Use 'V' component for further processing.

Reason:

a. 'V' component of HSV image provides a chromatic notion of the intensity

of the color, shortly brightness of color.

b. Eye dot in face image corresponds to bright spot due to light reflecting

from it.

c. Hence, using 'V' component for further analysis ensures better results.

3. Normalize 'V' component of HSV image.

4. Resize 'V' component of HSV image to standard size i.e. 200x100.

Pattern Recognition Neural Network

This algorithm uses Pattern Recognition Neural Network which recognizes whether the

given Eye Candidate segment is Left Eye, Right Eye or Not an Eye. Pattern Recognition

Neural Network is trained using training data for Left Eye, Right Eye and Not an Eye.

Neural Network Architecture

Pattern Recognition Neural Network used in this algorithm has one Input Layer (input size

[20000]), five Hidden Layers (20, 40, 80, 40, 20 activation units respectively) and one output

layer (output class [3]).

**Splitting training set:**

Training data used by this architecture is split into Training, Validation and Testing sets as

below:

*PatternNet.divideFcn = 'dividerand';*

*PatternNet.divideMode = 'sample';*

*PatternNet.divideParam.trainRatio = 75/100;*

*PatternNet.divideParam.valRatio = 20/100;*

*PatternNet.divideParam.testRatio = 5/100;*

**Training function:**

Here, we are using Scaled Conjugate Gradient backpropagation.

*PatternNet.trainFcn = 'trainscg';*

**Performance check:**

For checking performance of neural network, crossentropy technique is used.

**Output**

As our work is not completed yet, we haven’t generated any output. Hopefully after the completion of the coding work, we will get our expected output image which will show the position of eyes or detect the eyes from a given set of images of human face.

**ATTACHED CODE**

The code we have completed so far is attached with this report.