Robots for presentation scoring

Eye Contact Part

Taher Almoussali

17/10/2022

Contents

1 Pseudocodes

1. The First way: detect eye contact by using Deep eye contact model

DeepEC : Deep neural network trained to detect eye contact from facial image

Algorithm 1 detect_eye_contact

Require:

(1)*videoPath*: Path where the video is stored

(2)*modelWeight*: Path where weights of DeepEC model is stored

Ensure:

(1)*finalResult*: it will be 1 if video with eye contact or 0 if not

start algorithm
y = list stores 0 or 1 for each frame
confidence = list stores confidence of eye contact for each frame
k = list stores confidence then more 0.5
$model = modelStatic(modelWeight)$ ▷ load model weights
confidenceThresh = 0.9
TotalConfThresh = 0.75
TotalScoreThresh = 0.85
while videoCapture is opened **do**

    $frame \leftarrow readFrame$
    $boundedBox \leftarrow FaceDetectionModel(frame)$
    $face \leftarrow faceCrop(boundedBox)$
    $image \leftarrow face.resize(224, 224)$
    $output \leftarrow DeepECModel(image)$
    $confidence \leftarrow confidence(output)$
    if $confidence \geq confidenceThresh$ **then**
        $y \leftarrow 1$
    else
        $y \leftarrow 0$
    end if
    if $confidence > 0.5$ **then**
        $k \leftarrow confidence$
    end if
end while
$totalConf \leftarrow size(k)/size(y)$
$totalScore \leftarrow average(k)$
if $totalScore > TotalScoreThresh \wedge totalConf > TotalConfThresh$ **then**
    $finalResult \leftarrow 1$
else
    $finalResult \leftarrow 0$
end if
return $finalResult$
end algorithm

2.The second way: detect eye contact by using detect position of iris center

Algorithm 2 iris_position_in_eye

Require:

(1)*irisCenter*: iris center point of eye in frame
(2)*rightPoint*: Farthest to the right of the eye in frame
(3)*leftPoint*: Farthest to the left of the eye in frame
(4)*upPoint*: The highest point in the eye in frame
(5)*downPoint*: lowest point in the eye in frame

Ensure:

(1)*irisPosition*: The position of center of the iris in relation to the eye

start algorithm
rightThresh = 0.42
leftThresh = 0.57
upThresh = 0.25
downThresh = 0.58
$ratioHorizontal \leftarrow centerToRightDistance/leftToRightDistance$
$ratioVertical \leftarrow centerToUpDistance/DownToUpDistance$
if $ratioHorizontal > rightThresh \wedge ratioHorizontal \leq leftThresh$ **then**
    if $ratioVertical > upThreshandratioVertical \leq downThresh$ **then**
        $irisPosition \leftarrow center$
    end if
end if
if $ratioVertical > upThreshandratioVertical \leq downThresh$ **then**
    if $ratioHorizontal > rightThresh \wedge ratioHorizontal \leq leftThresh$ **then**
        $irisPosition \leftarrow center$
    end if
end if
    return *irisPosition*
end algorithm

Algorithm 3 detect_eye_contact_unsupervised

Require:

(1)*videoPath*: Path where the video is stored

Ensure:

(1)*finalResult*: it will be 1 if video with eye contact or 0 if not

start algorithm
AllScores = list stores 0 or 1 for each frame
decisionThresh = 0.5
while videoCapture is opened **do**

    $frame \leftarrow readFrame$
    $boundedBox \leftarrow FaceDetectionModel(frame)$
    $face \leftarrow faceCrop(boundedBox)$
    $landmarks \leftarrow faceMesh(face)$
    $irisPosition \leftarrow irisPositionInEye(landmarks)$
    if $irisPosition == center$ **then**
        $score \leftarrow 1$
    else
        $score \leftarrow 0$
    end if
    $AllScores \leftarrow score$
end while
if $average(AllScores) > decisionThresh$ **then**
    $finalResult \leftarrow 1$
else
    $finalResult \leftarrow 0$
end if
    return $finalResult$
end algorithm

2 Configuration

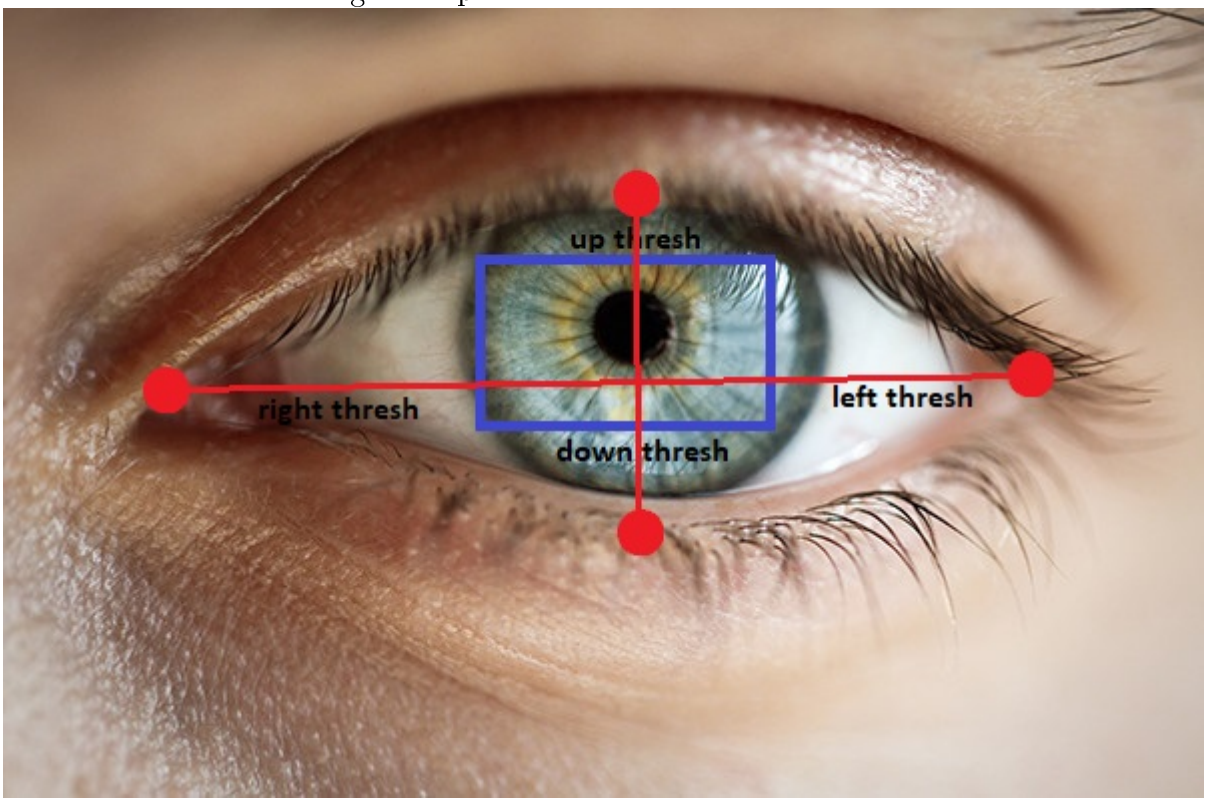
Table 1: detect eye contact

confidenceThresh = 0.9	TotalConfThresh = 0.75	TotalScoreThresh = 0.85

Table 2: detect eye contact unsupervised

decisionThresh = 0.5	leftThresh = 0.57	downThresh = 0.58
rightThresh = 0.42	upThresh = 0.25	

Figure 1: picture to illustrate



Note: (y true) has prudocted by manual labeling tool isn't handle frame by frame but it take each 15 frame as one block

(y product) handle frame by frame but we take dominant label in each 15 frame to match size of (y true) with size (y product)

3 Results

DeepEC = the first way
Heuristic = the second way

Figure 2: Accuracy

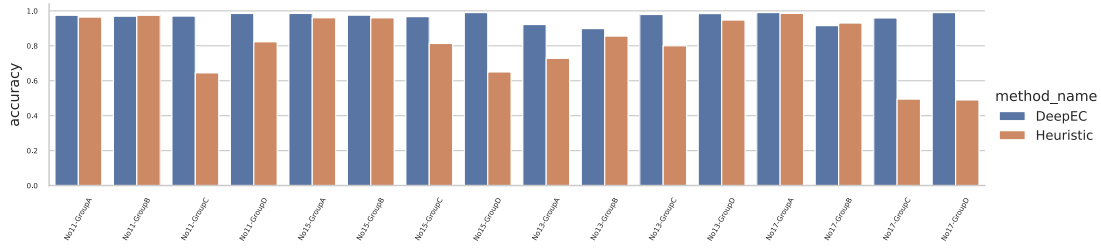
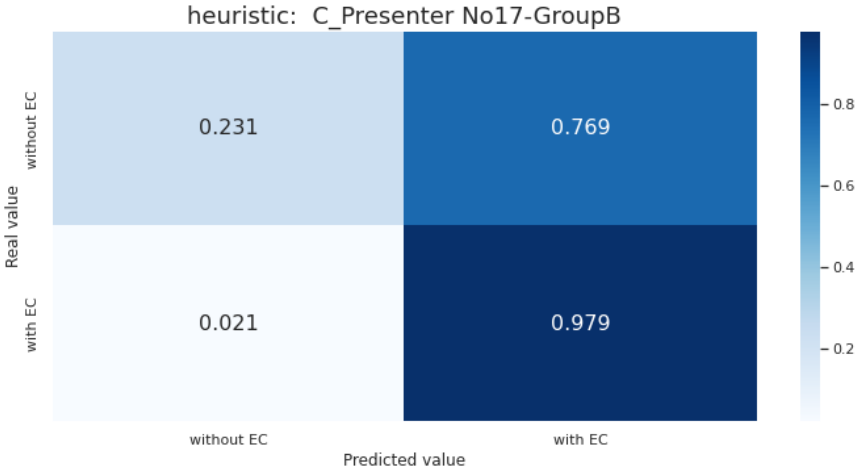


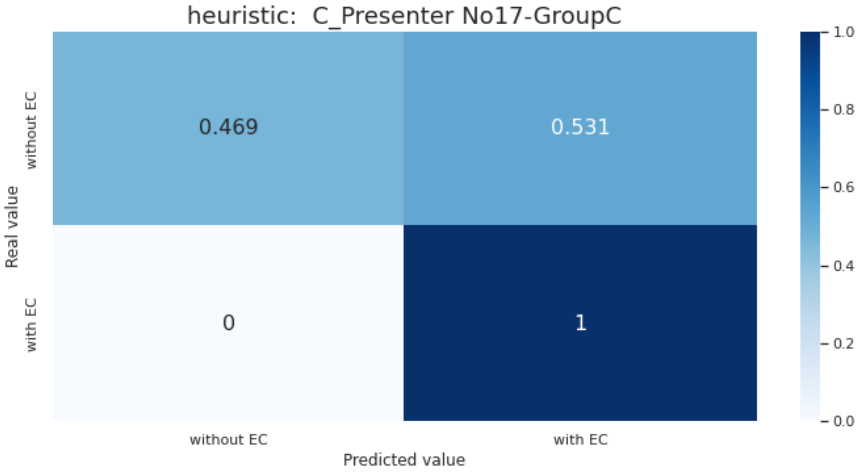
Figure 3: Confusion Matrix



accuracy score on test : 0.9303482587064676

	precision	recall	f1-score	support
0	0.43	0.23	0.30	13
1	0.95	0.98	0.96	188
accuracy			0.93	201
macro avg	0.69	0.60	0.63	201
weighted avg	0.91	0.93	0.92	201

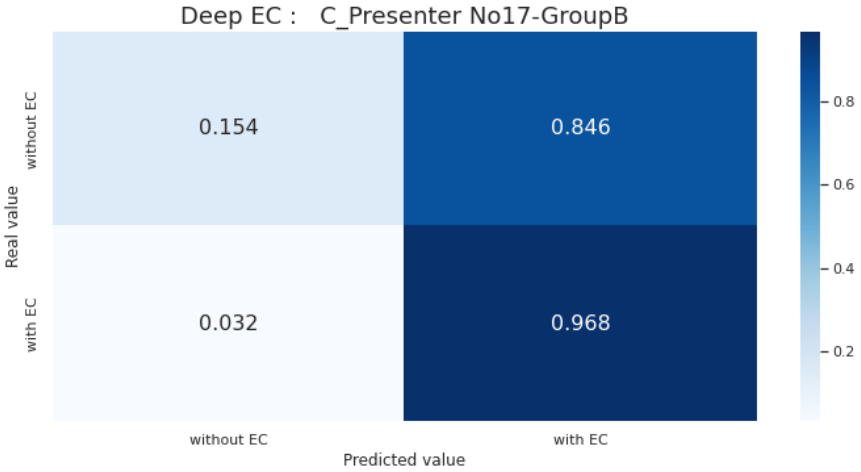
Figure 4: Confusion Matrix



accuracy score on test : 0.4946808510638298

	precision	recall	f1-score	support
0	1.00	0.47	0.64	179
1	0.09	1.00	0.16	9
accuracy			0.49	188
macro avg	0.54	0.73	0.40	188
weighted avg	0.96	0.49	0.62	188

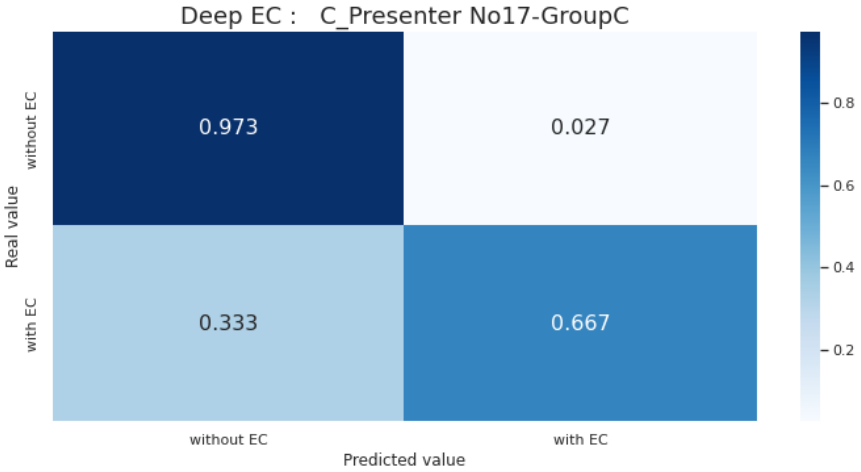
Figure 5: Confusion Matrix



accuracy score on test : 0.9154228855721394

	precision	recall	f1-score	support
0	0.25	0.15	0.19	13
1	0.94	0.97	0.96	188
accuracy			0.92	201
macro avg	0.60	0.56	0.57	201
weighted avg	0.90	0.92	0.91	201

Figure 6: Confusion Matrix



accuracy score on test : 0.9593908629441624

	precision	recall	f1-score	support
0	0.98	0.97	0.98	188
1	0.55	0.67	0.60	9
accuracy			0.96	197
macro avg	0.76	0.82	0.79	197
weighted avg	0.96	0.96	0.96	197