Eye Squint detection using Facial Landmarks in videos

Jagadish Kumaran Jayagopal

Abstract

Eye squinting or narrowing eyes is one of the body languages that may indicate displeasure, uncertainty or evaluation. Eye Squinting involves lowering of the brows and narrowing eyes to a certain extent. This can happen in a fraction of a second. It is said that squinting happens due to emotional or physical pain. Squinting was one of the prominent features observed during online exam videos whenever the questions were hard. The present study developed a squint detection model using 2D facial landmarks. 5 videos of totally 7.5 minutes recorded while the subject took an online exam was considered in this study. Eyes and eyebrow landmarks were detected from each frame of the video using the Face-Alignment facial landmark detector [2]. Features such as eyebrow closeness, eyebrow height from upper eyelid and eye openness served as inputs to a Deep Neural Network (DNN) model. A classification accuracy of 93.35%, sensitivity of 97.32% and specificity of 89.46% were achieved using the DNN model. The results suggest that eyebrow closeness, eyebrow height from upper eyelid and eye openness are useful features for the detection of squint in videos.

1. Introduction

Eye squinting involves narrowing of both eyes and furrowing of eyebrows [1] as shown in Figure 1. It is action 44 in the Facial Action Coding System (FACS) [6]. Eye squinting happens when a person is confused or curious [5]. It can also indicate discomfort, stress, anger or focused reading (such as trying to read something while there is bright light), *etc*.

In this paper, we detect eye squinting in the following way. For each image frame in the video, we first obtain the facial landmarks for the face in the frame. (There are totally 68 facial landmarks as shown in Figure 3. The method to extract frames from a video and the method to extract facial landmarks from a video are shown in Figure 5 and 6 respectively in the Appendix A.) We then define three features based on the facial landmarks:

1. *Eyebrow height*: Consider the 68 facial landmarks as illustrated in Figure 3. For $i = 1, 2, \cdots, 68$, let (x_i, y_i) denote the coordinates of the ith facial landmark in the image. For facial landmarks corresponding to eyes and eyebrows, we also give them alternative names as shown in Figure 2. The mapping from those names to their corresponding indices of facial landmarks is shown in Table 2. We define eyebrow height ($\mathcal{H}_L$ for left eye and $\mathcal{H}_R$ for right eye) as the below formula:

$$\mathcal{H}_L = \frac{||e_{1l} - p_{1l}|| + ||e_{2l} - p_{2l}|| + ||e_{3l} - p_{3l}||}{3||p_{4l} - p_{4r}||}$$

, where

$$||e_{1l} - p_{1l}|| = \sqrt{(x_{21} - x_{39})^2 + (y_{21} - y_{39})^2}$$

,

$$||e_{2l} - p_{2l}|| = \sqrt{(x_{20} - x_{38})^2 + (y_{20} - y_{38})^2}$$

,

$$||e_{3l} - p_{3l}|| = \sqrt{(x_{19} - x_{37})^2 + (y_{19} - y_{37})^2}$$

$$||p_{4l} - p_{4r}|| = \sqrt{(x_{45} - x_{36})^2 + (y_{45} - y_{36})^2}$$

$$\mathcal{H}_R = \frac{||e_{1r} - p_{1r}|| + ||e_{2r} - p_{2r}|| + ||e_{3r} - p_{3r}||}{3||p_{4l} - p_{4r}||}$$

, where

$$||e_{1r} - p_{1r}|| = \sqrt{(x_{22} - x_{42})^2 + (y_{22} - y_{42})^2}$$

,

$$||e_{2r} - p_{2r}|| = \sqrt{(x_{23} - x_{43})^2 + (y_{23} - y_{43})^2}$$

,

$$||e_{3r} - p_{3r}|| = \sqrt{(x_{24} - x_{44})^2 + (y_{24} - y_{44})^2}$$

Note that both $\mathcal{H}_L$ and $\mathcal{H}_R$ are normalized by the term $||p_{4l} - p_{4r}||$, which measures the span of the two eye corners.

We define *eyebrow height* $\mathcal{H}$ as the average of $\mathcal{H}_L$ and $\mathcal{H}_R$:

$$\mathcal{H} = \frac{\mathcal{H}_L + \mathcal{H}_R}{2}$$

2. *Eyebrow closeness*: We define the *eyebrow closeness* $\mathcal{C}$ as

$$\begin{aligned} \mathcal{C} &= ||e_{1l} - e_{1r}|| \\ &= \sqrt{(x_{22} - x_{21})^2 + (y_{22} - y_{21})^2} \end{aligned}$$

3. *Eye openness*: We define *eye openness* for the left eye $\mathcal{O}_L$ and for the right eye $\mathcal{O}_R$ as

$$\mathcal{O}_L = \frac{||p_{2l} - p_{6l}|| + ||p_{3l} - p_{5l}||}{2||p_{1l} - p_{4l}||}$$

, where

$$||p_{2l} - p_{6l}|| = \sqrt{(x_{38} - x_{40})^2 + (y_{38} - y_{40})^2}$$

$$||p_{3l} - p_{5l}|| = \sqrt{(x_{37} - x_{41})^2 + (y_{37} - y_{41})^2}$$

$$||p_{1l} - p_{4l}|| = \sqrt{(x_{45} - x_{42})^2 + (y_{45} - y_{42})^2}$$

$$\mathcal{O}_R = \frac{||p_{2r} - p_{6r}|| + ||p_{3r} - p_{5r}||}{2||p_{1r} - p_{4r}||}$$

, where

$$||p_{2r} - p_{6r}|| = \sqrt{(x_{43} - x_{47})^2 + (y_{43} - y_{47})^2}$$

$$||p_{3r} - p_{5r}|| = \sqrt{(x_{44} - x_{46})^2 + (y_{44} - y_{46})^2}$$

$$||p_{1r} - p_{4r}|| = \sqrt{(x_{45} - x_{42})^2 + (y_{45} - y_{42})^2}$$

We define $\mathcal{O}$ as the average of $\mathcal{O}_L$ and $\mathcal{O}_R$:

$$\mathcal{O} = \frac{\mathcal{O}_L + \mathcal{O}_R}{2}$$

We then train two models that detect eye squinting for a frame in the video – an SVM model, and an XGBoost classifier – as follows. Given the ith frame F_i, we consider the 6 frames preceding F_i and the 6 frames following F_i. That is, we consider the following 13 frames:$F_{i-6}, F_{i-5}, \cdots F_{i-1}, F_i, F_{i+1}, \cdots, F_{i+5}, F_{i+6}$
For $i - 6 \leq j \leq i + 6$, let α_j, β_j and γ_j denote the *eyebrow height*, *eyebrow closeness* and *eye openness* for the jth frame, respectively. Let $l_i \in \{0, 1\}$ denote the label of the ith frame, where $l_i = 1$ if the eyes are squinting in the ith frame, and $l_i = 0$ otherwise. Let $M_i = [(\alpha_{i-6}, \beta_{i-6}, \gamma_{i-6}), (\alpha_{i-5}, \beta_{i-5}, \gamma_{i-5}), \cdots,$ $(\alpha_i, \beta_i, \gamma_i), (\alpha_{i+1}, \beta_{i+1}, \gamma_{i+1}), \cdots, (\alpha_{i+6}, \beta_{i+6}, \gamma_{i+6})]$ be the 13×3 matrix that contains the 39 features of the 13 frames around the ith frame.

$$M = \begin{bmatrix} \alpha_{i-6} & \beta_{i-6} & \gamma_{i-6} \\ \alpha_{i-5} & \beta_{i-5} & \gamma_{i-5} \\ \vdots & \vdots & \vdots \\ \alpha_i & \beta_i & \gamma_i \\ \alpha_{i+1} & \beta_{i+1} & \gamma_{i+1} \\ \vdots & \vdots & \vdots \\ \alpha_{i+6} & \beta_{i+6} & \gamma_{i+6} \end{bmatrix}$$

For the DNN model, we let M_i be their input, and let l_i be their target output. (To simplify the training process for the model, we impose the constraint that the labels l_{i-6}, l_{i-5},$\cdots$, l_i, l_{i+1},$\cdots$, l_{i+6} have the same value. This constraint is not imposed during testing.) We used 4856 such samples to train the DNN model. A classification accuracy of 93.35%, sensitivity of 97.32% and specificity of 89.46% were achieved using the DNN model.

There has been a number of related works on detecting facial actions in the Facial Action Coding System. The action units and related references which detected them are given in Table 1.

Table 1: Facial action units and references

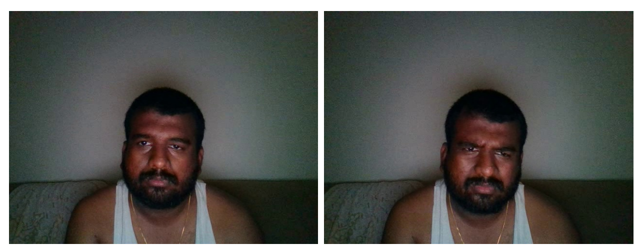
Action unit (AU)	References	AU name
AU1	[25], [26], [8]	Inner Brow Raiser
AU2	[25], [26], [8]	Outer Brow Raiser
AU4	[27], [14], [11], [17]	Brow Lowerer
AU5	[9]	Upper Lid Raiser
AU6	[28], [4], [3]	Cheek Raiser
AU7	[15], [22], [10]	Lid Tightener
AU9	[9], [16], [12]	Nose Wrinkler
AU10	[7]	Upper Lip Raiser
AU11	[7]	Nasolabial Deepener
AU12	[19], [18], [21]	Lip Corner Puller
AU13	[21]	Cheek Puffer
AU14	[26], [27], [21]	Dimpler
AU15	[26], [28], [4]	Lip Corner Depressor
AU16	[21]	Lower Lip Depressor
AU17	[19], [21], [26]	Chin Raiser
AU18	[21]	Lip Puckerer
AU20	[9], [7], [21]	Lip stretcher
AU22	[21]	Lip Funneler
AU23	[15], [22], [26]	Lip Tightener
AU24	[15], [3], [26]	Lip Pressor
AU25	[10], [9], [21]	Lips part
AU26	[14], [10], [21]	Jaw Drop
AU27	[7], [21]	Mouth Stretch
AU28	[10], [11], [21]	Lip Suck
AU34	[21]	
AU41	[23], [24]	Lid droop
AU42	[23], [24]	Slit
AU43	[7], [21], [23]	Eyes Closed
AU44	[21]	Squint
AU45	[10], [11], [3]	Blink
AU46	[13]	Wink

2. Proposed Method

Eye squinting is a facial action where the eyebrows are lowered and the eyes are narrowed. The degree of squinting differs every time the subject squints. The time the squinting lasts also varies every time a subject squints. Similar to [20] in this study facial landmark detector [2] is used to confine the eyes, eyebrows and their shapes. Features such as eye aspect ratio [20], eyebrows' closeness and height of eyebrow from the upper eyelid are computed using the detected landmarks in each frame. It is observed that eye-squint lasts at least the time that eye-blink lasts. So in this study a deep neural network which takes a big temporal window of a frame as input is trained [20].

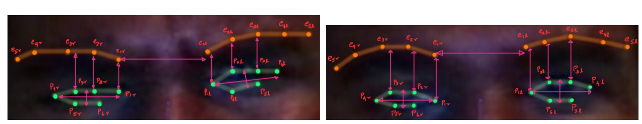
3. Description of features

The eye and eyebrow landmarks are detected for every frame in a video. It was experimentally observed that using only 2D facial landmarks as features to detect eye squint using DNN model was not successful. Eye squint involves lowering of eyebrows along with certain degree of narrowing eye. During lowering of eyebrows, the eyebrows come close to each other. It is also observed that the vertical distance between the eyebrows and upper eye lids decreases while the lowering of eyebrows occur. Therefore it was decided to use $\mathcal{C}$ and $\mathcal{H}$ as features along with $\mathcal{O}$ to detect squint. In the calculation of $\mathcal{H}$, we have used only three landmark points of eye and eyebrow that are close to the nose as shown in Figure 2. This is because, it was observed that the portion of the eyebrow (e_{4l}, e_{5l}, e_{4r}, e_{5r}) that is away from the nose do not move much and so the height of the those two points from the upper eyelid don't change significantly during squinting.



(a) Normal eyes (NE) (b) Squinted eyes (SE)

Figure 1: Normal and Squinted eyes



(a) NE with landmarks (b) SE with landmarks

Figure 2: Normal and Squinted eyes with landmarks

4. Model Architecture

In this model, the features were passed on to a Dense layer with 200 neurons and 'ReLU' activation. One more

Figure 3: An illustration of the 68 facial landmarks for a human face.

dense layer with 100 neurons and 'ReLU' activation was added after the first Dense layer. Finally a dense layer with 1 neuron and 'Sigmoid' activation was added. Initially the model was compiled with Adam optimizer. Since this is a binary classification problem, binary cross-entropy is used as loss function that the model attempts to minimize and binary-accuracy is used as the metrics to be evaluated by the model during training and testing. After hyper-parameter tuning, the number of epochs and batch size were fixed as 500 and 15 respectively. The model architecture is shown in Figure 4.

4.1. Shapes of Tensors

- Dense Layer I:
  Input Tensor Shape: (39,) – (Batch size, input dimension)
  //Model will take as input arrays of shape (*, 39)

  Output Tensor Shape: (None, 200) – (Batch size, number of neurons)
- Dense Layer II:
  Input Tensor Shape: (None, 200) – (Batch size, number of neurons)
  Output Tensor Shape: (None, 100) – (Batch size, number of neurons)
- Dense Layer III:
  Input Tensor Shape: (None, 100) – (Batch size, number of neurons)
  Output Tensor Shape: (None, 1) – (Batch size, number of neurons)

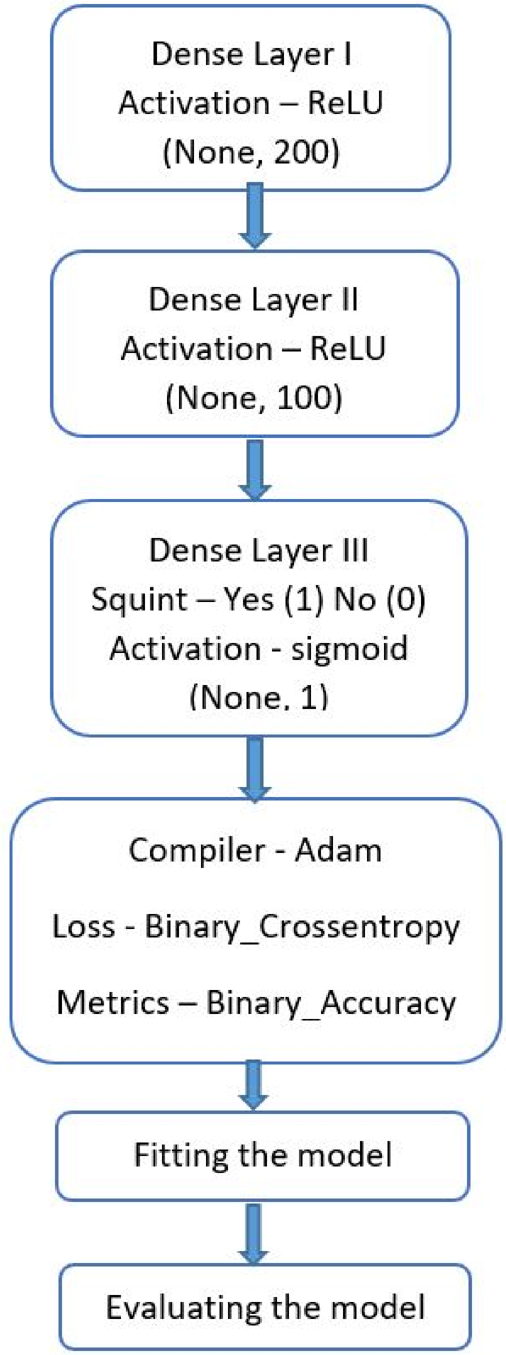


Figure 4: Deep Neural Network Model Architecture.

5. Classification

During the data collection, it was noted that whenever the subject squints, its presence is seen in at least four frames of the video. Also, blink occurs while the subject had been squinting for a certain amount of time. [20] had experimentally found that for each frame, using a 13-dimensional feature obtained by concatenating $\mathcal{O}$s of its ± 6 neighboring frames have notable impact in blink detection. This concept has been adopted in this study. Here for each frame, a 39-dimensional feature was computed ,considering ± 6 neighboring frames for each of the three features. In this study, a Deep Neural Network model was used to implement the squint detection and its performance was evaluated. Similar to [20], ground-truth squints with $'1'$ labels were considered as positive samples and frames with 'no squint' label whose ± 6 neighboring frames also had no squint $'0'$ labels were considered as negative samples during training. It was experimentally observed that the selection of negative samples per the above method helps the classifier perform better while testing. During testing a 39-dimensional feature was computed like in [20].

Alternative facial landmark labels for eyes and eyebrows	Index of facial landmarks (Between 1 and 68)
e_{1l}	21
e_{2l}	20
e_{3l}	19
e_{4l}	18
e_{5l}	17
p_{1l}	39
p_{2l}	38
p_{3l}	37
p_{4l}	36
p_{5l}	41
p_{6l}	40
e_{1r}	22
e_{2r}	23
e_{3r}	24
e_{4r}	25
e_{5r}	26
p_{1r}	42
p_{2r}	43
p_{3r}	44
p_{4r}	45
p_{5r}	46
p_{6r}	47

Table 2: Alternative facial landmark labels for eyes and eyebrows and their corresponding indices

5.1. Classification metrics

In this study we have used metrics such as accuracy, sensitivity and specificity to evaluate the classification models. We will first see the definition of the terms used in estimating the metrics below:

- True Positives (TP): The total number of accurate predictions that were 'positive' or '1'. In our study, this is the total number of samples correctly predicted as squint.
- False Positives (FP): The total number of inaccurate predictions that were 'positive' or '1'. In our study, this is the total number of samples incorrectly predicted as squint.
- True Negatives (TN): The total number of accurate predictions that were 'negative' or '0'. In our study, this is the total number of samples correctly predicted as "not squint".
- False Positives (FN): The total number of inaccurate predictions that were 'negative' or '0'. In our study, this is the total number of samples incorrectly predicted as "not squint".

Now we will define the classification metrics such as accuracy, sensitivity and specificity and present their mathematical expressions.

- Accuracy: It is a measure of the fraction of all instances that are correctly classified. It is the ratio of the number of correct classifications to the total number of predictions.

  $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Sensitivity: It is the proportion of instances that were predicted as positive and are positive of all the instances that actually are positive. It is the ratio of the number of correct positive classifications to the total number of true positives.

  $$Sensitivity = \frac{TP}{TP + FN}$$

- Specificity: It is a measure of the fraction of all instances that are correctly classified. It is the ratio of the number of correct negative classifications to the total number of true negatives.

  $$Specificity = \frac{TN}{TN + FP}$$

6. Experiments

The training data-set contains 4856 frames taken from 4 videos. The test data-set contains 2343 frames taken from one video. Facial landmarks for the eyes were estimated using face-alignment model [2]. These are 2D landmarks.

Metrics	DNN
Accuracy	93.35 %
Sensitivity	97.32 %
Specificity	89.46 %

Table 3: Classification metrics

In this study we have considered the landmarks of eyes and eyebrows. There are 22 landmarks which describe the eyes and eyebrows. The classification was performed using the DNN model. Initially 2D facial landmarks themselves were used as features for each frame in the video. But the DNN model was not even fitting on the training data. So the manually computed features such as $\mathcal{O}$, $\mathcal{H}$ and $\mathcal{C}$ for each frame in the video were used as inputs to the DNN model and this approach was successful. The classification metrics of DNN model is shown in Table 3.

6.1. Alternate approach

In the current approach, we have used manually calculated features as input to the deep neural network. An alternate approach is to use the euclidean distance between each of these landmarks with the other 21 landmarks as features for each frame in the video. I am currently working on preparing the data for this approach and will update the classification metrics for this approach as soon as I evaluate the model.

References

[1] Andrew Billings. Self-monitoring in the treatment of tics: A single-subject analysis. *Journal of Behavior Therapy and Experimental Psychiatry*, 9(4):339–342, 1978.

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[3] Wei-Yi Chang, Shih-Huan Hsu, and Jen-Hsien Chien. Fatauva-net: An integrated deep learning framework for facial attribute recognition, action unit detection, and valence-arousal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–25, 2017.

[4] Wen-Sheng Chu, Fernando De la Torre, and Jeffrey F Cohn. Learning spatial and temporal cues for multi-label facial action unit detection. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 25–32. IEEE, 2017.

[5] Sidney D'Mello, Tanner Jackson, Scotty Craig, Brent Morgan, P Chipman, Holly White, Natalie Person, Barry Kort, R El Kaliouby, Rosalind Picard, et al. Autotutor detects and responds to learners affective and cognitive states. In *Workshop on emotional and cognitive issues at the international conference on intelligent tutoring systems*, pages 306–308, 2008.

[6] Paul Ekman and Wallace V Friesen. *Manual for the facial action coding system*. Consulting Psychologists Press, 1978.

[7] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Multi-conditional latent variable model for joint facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3792–3800, 2015.

[8] Itir Onal Ertugrul, Laszlo A Jeni, and Jeffrey F Cohn. Pattnet: Patch-attentive deep network for action unit detection.

[9] Sayan Ghosh, Eugene Laksana, Stefan Scherer, and Louis-Philippe Morency. A multi-label convolutional neural network approach to cross-domain action unit detection. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 609–615. IEEE, 2015.

[10] Amogh Gudi, H Emrah Tasli, Tim M Den Uyl, and Andreas Maroulis. Deep learning based facs action unit occurrence and intensity estimation. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–5. IEEE, 2015.

[11] Shashank Jaiswal and Michel Valstar. Deep learning the dynamic appearance and shape of facial action units. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–8. IEEE, 2016.

[12] Shreyank Jyoti, Garima Sharma, and Abhinav Dhall. Expression empowered residen network for facial action unit detection. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[13] Sander Koelstra and Maja Pantic. Non-rigid registration using free-form deformations for recognition of facial actions and their temporal dynamics. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–8. IEEE, 2008.

[14] Mihee Lee, Vladimir Pavlovic, Maja Pantic, et al. Fast and effective adaptation of facial action unit detection deep model. *arXiv preprint arXiv:1909.12158*, 2019.

[15] Wei Li, Farnaz Abtahi, Zhigang Zhu, and Lijun Yin. Eac-net: A region-based deep enhancing and cropping approach for facial action unit detection. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 103–110. IEEE, 2017.

[16] Yong Li, Jiabei Zeng, Shiguang Shan, and Xilin Chen. Self-supervised representation learning from videos for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10924–10933, 2019.

[17] Jingjing Liu, Bo Liu, Shaoting Zhang, Fei Yang, Peng Yang, Dimitris N Metaxas, and Carol Neidle. Recognizing eyebrow and periodic head gestures using crfs for non-manual grammatical marker detection in asl. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2013.

[18] Peng Liu, Zheng Zhang, Huiyuan Yang, and Lijun Yin. Multi-modality empowered network for facial action unit detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2175–2184. IEEE, 2019.

[19] Zhiwen Shao, Jianfei Cai, Tat-Jen Cham, Xuequan Lu, and Lizhuang Ma. Weakly-supervised unconstrained action unit detection via feature disentanglement. *arXiv preprint arXiv:1903.10143*, 2019.

[20] Tereza Soukupova and Jan Cech. Eye blink detection using facial landmarks. In *21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia*, 2016.

[21] Bilal Taha, Munawar Hayat, Stefano Berretti, and Naoufel Werghi. Learned 3d shape representations using fused geometrically augmented images: Application to facial expression and action unit detection. *arXiv preprint arXiv:1904.04297*, 2019.

[22] Chuangao Tang, Wenming Zheng, Jingwei Yan, Qiang Li, Yang Li, Tong Zhang, and Zhen Cui. View-independent facial action unit detection. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 878–882. IEEE, 2017.

[23] Ying-li Tian, Takeo Kanade, and Jeffrey F Cohn. Eye-state action unit detection by gabor wavelets. In *International Conference on Multimodal Interfaces*, pages 143–150. Springer, 2000.

[24] Ying-li Tian, Takeo Kanade, and Jeffrey F Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 229–234. IEEE, 2002.

[25] Cheng-Hao Tu, Chih-Yuan Yang, and Jane Yung-jen Hsu. Idennet: Identity-aware facial action unit detection. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[26] Chu Wang, Jiabei Zeng, Shiguang Shan, and Xilin Chen. Multi-task learning of emotion recognition and facial action unit detection with adaptively weights sharing network. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 56–60. IEEE, 2019.

[27] Le Yang, Itir Onal Ertugrul, Jeffrey F Cohn, Zakia Hammal, Dongmei Jiang, and Hichem Sahli. Facs3d-net: 3d convolution based spatiotemporal representation for action unit detection.

[28] Kaili Zhao, Wen-Sheng Chu, and Honggang Zhang. Deep region and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3391–3399, 2016.

A. Appendix I

In this appendix, we show how to get images from a video, and how to get facial landmarks from an image.

The method of getting images from a video is straightforward. Basically, we read images from the video one by one, and save them as separate files. The code is shown in Figure 5.

The facial landmarks can be obtained by multiple software packages. The package we use here is face-alignment, which can find facial landmarks for each image that contains faces. The code is shown in Figure 6.

```
Python 3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 19:07:31)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import face_alignment
>>> from skimage import io
>>> import pandas as pd
>>> import numpy as np
>>> fa = face_alignment.FaceAlignment(face_alignment.LandmarksType._2D, flip_input=False)
>>> preds = fa.get_landmarks_from_directory('/mnt/sde/jagadish/vid1/')
100%|██████████████████████████████| 710/710 [00:46<00:00, 15.36it/s]
>>> df = pd.DataFrame.from_dict(data=preds,orient='index')
>>> df.to_csv('data_t.csv', mode='a', index=True)
```

Figure 5: Command line code to extract facial landmarks from videos

```
import cv2 as cv
count = 0
videoFile = '/home/jagadish/20191016214708.avi'
cap = cv.VideoCapture(videoFile)  #capturing the video from the
given path

while(cap.isOpened()):
  ret, frame = cap.read()
  if (ret != True):
      break

  filename ="frame\%d.jpg" % count

  cv.imwrite(filename, frame)
```

Figure 6: Command line code to extract frames from videos