Face Recognition Implementation as an Attendance Feature on Web-Based Video Conference Application

Robin 1*, Fransiskus Hermanto 2, Wenripin Chandra 3
1,2,3Universitas Pelita Harapan, Kampus Medan, Indonesia
1robin.huang@lecturer.uph.edu, 2th80050@student.uph.edu, 3wenripin@lecturer.uph.edu

ABSTRACT

Online meeting by video conference has been used extensively in business, government, education and many more. One important aspect in doing online meeting is making sure the attendees recorded for administration purposes. In order to make attendance record, the hosts often have to spend some time to do attendance record because most of video conference applications do not have automatic attendance feature. To ease the meeting host in creating attendance record, we utilized computer vision to do attendee face recognition. In order to create valid attendance record, it is very crucial to make sure that identity of the attendees recognized correctly. We employ face-api.js in doing face recognition. face-api.js is a Javascript API (application programming interface) for face detection, face recognition and face landmark detection. This API support several face detection, face recognition and face expression recognition models. We use facial landmarks and combine it with MobileNets to do face recognition. We implement implement all of these into a web application to facilitate the usage in real life situation. The web-based video conference application has the capability to record the identity of the face owner and generate attendees’ records automatically. The testing shows that face-api.js can successfully recognize registered faces under fair video conference condition and unsuccessful only when faces covered by another object.

INTRODUCTION

Online meeting by video conference has been used extensively in business, government, education and many more. One important aspect in doing online meeting is making sure the attendees recorded. Although video conference applications have feature to record attendees, the hosts often have to spend some time to do attendance record because most of these applications do not have automatic attendance feature. The most common way to record attendance are either by taking screenshot, downloading attendance list or calling attendees’ names one by one to verify if they attend the meeting. In other words, the host must manually labor to know attendees of his meeting.

To rectify this situation, we propose using face recognition to the attendees faces to make automatic meeting attendance record. We utilized computer vision to do recognition and then record the identity of the face owner. We implement this into a web application to facilitate the usage in real life situation.

LITERATURE REVIEW

Computer vision is an automatic analysis process of images and videos by computer to gain information about the world. This field was inspired by human vision system (HVS) (Dawson-Howe, 2014). In computer vision, programmer can think as if the program can duplicate the HVS. However, the main difficulty is The main difficulty is that the programmer does not understand what the human vision system is doing most of the time.

Computer vision has been widely implemented in industry, in particular enabling automatic inspection of manufactured goods at each stage in the production line, for example: inspection of the printed circuit board to ensure that the tracks and components are positioned correctly, checking the quality of label printing, checking bottles to ensure they are filled properly, and so on. Computer vision can also be used along biometrics to do identification.

Biometrics are basically physical or behavioral characteristics of humans that can be used in the digital identification process. Biometric identification, or biometrics, refers to the identification of an individual based on characteristics that distinguish him (Bolle, Connell, Pankanti, Ratha, & Senior, 2004).
Six biometrics that are often used in today's authentication systems can be seen in Table 1.

Table 1. The Biometrics Used for Digital Authentications

<table>
<thead>
<tr>
<th>Physiologies</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>Signature</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>Voice</td>
</tr>
<tr>
<td>Hand geometries</td>
<td></td>
</tr>
<tr>
<td>Iris</td>
<td></td>
</tr>
</tbody>
</table>

Face recognition is a problem of visual pattern recognition. Humans recognize visual patterns over time, and obtain visual information through eye. Information recognized by the brain as meaningful concepts. For computer, whether it is an image or a video, it is a matrix of many pixels. The machine has to figure out what concept a particular piece of data represents. This is classification problem in visual model recognition. For facial recognition, it is mandatory to distinguish whose face belongs to in the piece of data (Li, Mu, Li, & Peng, 2020).

Around 1950, people began to learn how to make machines recognize faces. In 1964, the research application of face recognition officially began, mainly using face geometry to perform recognition.

In the initial stage, there are two algorithms used:

1. Principal Component Analysis (PCA)
   PCA is the most widely used data dimensionality reduction algorithm. In face recognition, PCA implements facial feature extraction. PCA is usually used to preprocess data before other analyses. In dealing with data with more dimensions, PCA can eliminate redundant information and noise, maintain important characteristics in a data, greatly reduce dimensionality, increase speed in data processing, and save a lot of time and cost.

2. Linear Discriminate Analysis (LDA)
   For facial recognition datasets that use labels, you can use linear discriminate analysis (LDA). LDA is used to classify faces. PCA requires the data variance after dimension reduction to be as large as possible so that the data can be divided as widely as possible, while LDA requires the variance within the same data group category after projection to be as small as possible, and the variance between groups to be as large as possible.

At the artificial features and classifier stage, there are three methods used:

1. Support Vector Machine (SVM)
   SVM is an algorithm specifically intended for small sample, high-throughput facial recognition problems. SVM is a classifier developed from common imaging algorithm. In face recognition, we can use extracted facial features and SVM to search for hyperplanes to distinguish different faces.

2. Adaboost
   Adaboost algorithm originally proposed by Schapire. This algorithm is used for facial recognition. Boosting algorithms can increase the accuracy of various algorithms for machine learning. There are two problems for face recognition in the boosting algorithm. The first is how to organize the training set, and the other is how to combine weak classifiers so that they can form a strong classifier. Adaboost has improvement for these problems, and has been recognized as an effective and practical boosting algorithm in facial recognition. Adaboost uses weighted training data, rather than randomly selected samples to focus on relatively difficult training data samples.

3. Small samples
   The problem of small samples refers to the fact that the number of training samples for face recognition is too small, which causes most face recognition algorithms fail to achieve ideal performance in recognition, effectively store image information, maintaining relationship between samples, and improve the effect of face recognition in the future.

Face recognition is very important wherever identification is needed. It can be applied in various fields. One of such things is video conference.

Video conference is the two-way or multipoint real-time reception and transmission of audio and video signals by people in different locations for real-time communication (Johnson, 2005). Video conference is conducted using applications such as Zoom, Google Meets, Microsoft Teams, and so on. In video conference session, it is often necessary to have attendance record for administration or other purposes. It's too bad that there is no features to automatically recognize attendees to create attendance record in these applications.

METHOD

We use facial landmarks (Amato, Falchi, Gennaro, & Vairo, 2018) and combine it with MobileNets model (Howard, et al., 2017) to do face recognition. We implement this face recognition into a web-based application using face-api.js library. The functional analysis for this application is done by use case as shown in Figure 1.
RESULT

To use the application, we need computer with high-definition camera, web browser and face-api.js library. In order to be recognized, the faces must be registered.

We do accuracy testing using two samples with various conditions. We use maximum descriptor distance between 0.3 to 0.6. We choose this test condition to mimic real life video conference situation. The face samples can be seen in Table 2.

Table 2. Registered Face Images

<table>
<thead>
<tr>
<th>Sample 1</th>
<th>Sample 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Sample 1" /></td>
<td><img src="image2.png" alt="Sample 2" /></td>
</tr>
</tbody>
</table>

When run for the first time, application needs to load the face recognition mode which is huge. This caused the web browser took long time to execute the recognition process.

The results of the testing can be seen on table 3 and 4. The green dots and frames around the faces indicate the successful recognition while the red ones indicate the unsuccessful attempts.

Table 3. Accuracy Testing Result for 1st Sample

<table>
<thead>
<tr>
<th>No.</th>
<th>Condition</th>
<th>Face Image</th>
<th>Descriptor distance value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Normal</td>
<td><img src="image3.png" alt="Face Image" /></td>
<td>0.41 – 0.45</td>
</tr>
</tbody>
</table>
2. Wearing Mask

3. Low Light

4. Wearing mask and low light

5. Further from camera

6. Looking down

7. Looping up

0.45 – 0.49

0.46 – 0.48

0.49 - 0.57

0.39 - 0.45

0.47 – 0.58

0.45 – 0.51
8. Looding beside

0.48 – 0.60

9. Closing eyes

0.41 – 0.43

10. Wearing glasses

0.47 – 0.53

11. Wearing glasses and mask

0.47 – 0.53

12. Closing face using a hand

0.57 – 0.63
<table>
<thead>
<tr>
<th>No.</th>
<th>Condition</th>
<th>Face Image</th>
<th>Descriptor distance value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Normal</td>
<td><img src="normal_face.png" alt="Image" /></td>
<td>0.30 – 0.33</td>
</tr>
<tr>
<td>2.</td>
<td>Wearing Mask</td>
<td><img src="mask_face.png" alt="Image" /></td>
<td>0.39 – 0.43</td>
</tr>
<tr>
<td>3.</td>
<td>Low Light</td>
<td><img src="low_light_face.png" alt="Image" /></td>
<td>0.38 – 0.44</td>
</tr>
<tr>
<td>4.</td>
<td>Wearing mask and low light</td>
<td><img src="mask_low_light_face.png" alt="Image" /></td>
<td>0.59 – 0.68</td>
</tr>
<tr>
<td>5.</td>
<td>Further from camera</td>
<td><img src="further_face.png" alt="Image" /></td>
<td>0.38 – 0.40</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Percentage</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Looking down</td>
<td>0.33 – 0.41</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Looping up</td>
<td>0.48 – 0.51</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Looding beside</td>
<td>0.45 – 0.49</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Closing eyes</td>
<td>0.32 – 0.35</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Wearing glasses</td>
<td>0.35 – 0.40</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Wearing glasses and mask</td>
<td>0.47 – 0.57</td>
<td></td>
</tr>
</tbody>
</table>
The testing shows that application can’t recognize faces in low light condition and when faces hidden behind other objects. We also see the tendency of unsuccessful recognition when value of descriptor distance greater than 0.6.

**DISCUSSIONS**

The testing shows that face-api.js can successfully recognize registered faces under fair video conference condition. The unsuccessful recognition in this test happened when the face was covered by another object such as mask or hand.

**CONCLUSION**

Implementation of face-api.js library in web-based video conference application shows that this feature can be used to facilitate creation of attendance record during video conference session. This should ease the conference host.

**REFERENCES**


