International Journal on Recent Researches In Science, Engineering & Technology
(Division of Computer Science and Engineering)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. It is an absolutely free (No processing charges, No publishing charges etc) Journal Indexed in JIR, DIIF and SJIF.
Research Paper
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 4, Issue 12, December 2016.

JIR IF : 2.54
DIIF IF : 1.46
SJIF IF : 1.329

Applications of Biometrics and Facial Recognition in the Remote Learning Environment

Dr. N.Sandya Rani
Department of Computer Science and Engineering,Shadan College of Engineering and Technology HYD,T.S,INDIA

Abstract

As cameras have become an critical aspect in present day computer systems and cellular devices, and the capability of these cameras in computer systems and mobile devices are ever increasing, facial awareness is turning into an without difficulty reachable functionality for many laptop applications. Another place that has benefited from the quick advancement of technology, together with skills of computers as nicely as the pace and bandwidth of the Internet, is distance learning. Online schooling and coaching are nowadays an emerging and prosperous region of business. More and greater ordinary greater academic institutes are offering on-line classes, joining their pure on line counterparts and competitors. In this paper, we first talk about frequent algorithms and methods used in facial recognition. We then existing methods to integrating facial recognition into Web purposes and explore how facial recognition may also be employed in distance getting to know surroundings to improve the effectiveness and effectivity of distance learning. At last we describe practical methods to integrate facial cognizance modules into current learning administration systems.

Keywords: Distance Learning, Learning Management Systems, Facial Recognition, Biometrics,

I. Introduction

Facial awareness is turning into a more and greater famous and on hand biometric method to enhance computer applications. Government agencies and banks have been actively integrating facial consciousness into their systems. The TSA is working on a Registered Traveler software the use of biometrics in order to conduct safety screenings in a faster trend [1]. Several banks nationwide are looking for approaches to implement facial focus into ATMs [2]. Another place that has been through a speedy growing due to advancement in technologies is distance learning. According to the 13th annual record of the nation of online getting to know in US Higher education, an on line record titled Tracking Online Education in the United States, there have been in whole of 5.8 million distance studying college students in the US in Fall 2014, amongst which 2.85 million had been taking all publications on-line and the other 2.97 million have been taking part of their route load on-line [3]. In this paper, we strive to research and discover methodologies to put in force facial focus systems and

examine how to efficiently deploy facial awareness in a distance getting to know environment or learning administration systems. We first inspect how facial attention is implemented, in concept and in practice. Then we focal point on Web purposes and explore how cutting-edge Web technologies have advanced to allow the integration of facial recognition into cutting-edge Web applications. At final we describe numerous facial consciousness elements that ought to be used in distance learning environment, and discuss practical strategies to integrate facial awareness modules into famous gaining knowledge of management systems.

2. Methods of Facial Recognition

There are in truth 4 principal facial recognition methods in use today: function analysis, neural network, eigenfaces and computerized face processing [4]. In the first method, feature analysis, each character has unique facial traits that can be measured. These characteristics include, but now not be constrained to, shape, spacing, texture, size and width. Each human face has approximately 80+ of these unique features named nodal points. These nodal points create a numerical code known as a face print. The Face print can be used to extract a matching picture template from a database [5]. The 2nd method, the neural network primarily based method, works in a comparable fashion to the way neurons work in the brain. The neural community is broken down into enter and output nodes, with nodes in between permitting for paths of communication. A particular node route is chosen primarily based on an algorithm. The neural community is deployed in facial consciousness through the use of the entire facial region. This contrasts with characteristic evaluation that solely uses nodal points. The neural network uses two techniques to understand a facial image. First, all faces have differing shapes, facets and dimensions. Principle factor analysis narrows the range of pics a face will be in contrast in opposition to with the aid of ruling out a particular team of pictures due to differing features or dimensions. In the 2nd phase, Back Propagation Neural Network will function the authentic awareness of an picture to the face [6]. The 0.33 facial cognizance technique in use is eigenfaces. The gain of the eigenface approach over other strategies is its velocity in the facial consciousness process. Eigenfaces can be surely described as a blurred facial image. The blurred picture still permits for the recognition of distinguished facial features. These points include the nose, eyes and mouth. Eigenfaces are generated mathematically through likelihood principle the usage of matrices and vectors. The disadvantage of this technique is its dependency on the identical lights and stage views in order to recognize faces against snap shots accurately. The use of Eigenfeatures has been developed to overcome this disadvantage. Eigenfeatures measure the distance between facial features, and then compare distances towards one some other [7].

The fourth facial cognizance approach is automatic facial processing. This method, like Eigenfaces, is established on the same lights and facial positions. However, this issue can be remedied as well, making automated facial recognition exceedingly accurate. A set of photographs for one person is stored inside the system. These units of snap shots fluctuate in facial expression, route and position of the face, lights and so on. Each individual image is called a probe image. From here, each probe picture is molded collectively to create an average image. The common photo is used for the facial comparison and can consequently increase accuracy with the aid of nearly fifty percent [8]. Newer facial cognizance techniques have additionally has been made on hand in latest years. In the past, facial recognition techniques have been restrained to 2D images, which have been sensitive to insufficient facts points due to low lighting, partial images, indirect snap shots etc..Newer strategies have been brought primarily based on three-d photographs to address these troubles and furnish higher accuracy. Another progress in facial focus algorithm and strategies is from deep learning. According to [19], the recent development in facial focus is normally due to two factors: 1). give up to quit gaining knowledge of the use of CNN (Convolutional Neural Networks); 2). large scale of education datasets handy now due to the increasing use of facial consciousness based applications. Recent complete survey on facial attention techniques can be seen in [9][10][11][19][20].

3. Acquiring the Facial Images

No be counted what facial consciousness method is used in your application, the face pix have to be acquired thru a kind of cameras. Fortunately, digital cameras have become extensively available due to development in technology and, especially, cellular computing. In the circumstance of distance learning, the focal point is how stay face pix may be received through Web browsers, or in other phrases that are more involved for Web developers, how to acquire face pix thru Web APIs. In history, the techniques to achieve the face pictures via Web applied sciences have varied. A frequent strategy would have been the use of web based plug-in like Flash or Silver mild in order to gain the face snap shots and then operate face detection, contrast and recognition. Fortunately, the state of affairs has changed and the techniques have been via standardization due to HTML5. With HTML5 and JavaScript, builders can achieve face pix with pure web code barring any plug-in. This offers less dependency and better integration with different technologies such as facial consciousness for Web utility development. One approach to gain a screenshot from the web camera in HTML is via a modified input element. The syntax for this input tag can also seem to be like below. . The above input issue will tie an photograph screenshot from your internet camera into a file enter tag to be uploaded to the Web server site. Support for this method is presently mobile centric and all cell browsers have certain amounts of help for this method. However, solely Chrome having help for it on desktop, which makes this approach unsuitable for a standard distance learning surroundings as desktop computer systems are very frequent in distance gaining knowledge of environment. The extra appropriate approach, and the one used in our purposes is the usage of the get User Media () API constructed into JavaScript. The gain of this approach, as well as the previous method when supported, is that it does no longer require a plug-in to be used so all people with a present day browser will be in a position to use your Web utility and function facial recognition easily. Using this approach, the Web utility will first do a characteristic take a look at to see if get User Media () API is on hand to the browser environment. In the first release of get User Media () API, the get User Media () approach belongs to JavaScript object window. Navigator. The JavaScript code for doing this checking would be similar to the code below.

```
1
2  function hasGetUserMedia() {
3          //hide the diffence of getUserMedia prefix among browsers
4          return (navigator.getUserMedia ||
5                  navigator.webkitGetUserMedia ||
6                  navigator.mozGetUserMedia ||
7                  navigator.msGetUserMedia);
8  }
```

Fig. 1: Check gets User Media hold up with window. Navigator Object

The revised get User Media() API makes use of navigator. Media Devices object [12]. When invoked, the Media Devices. Get User Media () method would immediate person for permission to use video and/or audio devices. A Promise object might also be again relying on the user's response. The JavaScript code can also look like beneath in which my Constraints is a Media Stream Constraints object in get User Media () API. This object specifies the sorts of media of request and necessities for every type.

```
navigator.mediaDevices.getUserMedia(myConstraints).then(function(mediaStream) {
  /* process the stream */
}).catch(function(err) {
  /* handle the error */
});
```

Fig. 2: Draw on of the Current get User Media () API

Since assist for get User Media() varies from browser to browser, we adopt the JavaScript code as proven in [12] to supply flexibility so that both older and modern-day browsers are supported. The JavaScript code as adopted from [12] is shown in Fig. 3

```
// Older browsers might not implement mediaDevices at all, so we set an empty object first
if (navigator.mediaDevices === undefined) {
  navigator.mediaDevices = {};
}

// Some browsers partially implement mediaDevices. We can't just assign an object
// with getUserMedia as it would overwrite existing properties.
// Here, we will just add the getUserMedia property if it's missing.
if (navigator.mediaDevices.getUserMedia === undefined) {
  navigator.mediaDevices.getUserMedia = function(constraints) {

    // First get ahold of the legacy getUserMedia, if present
    var getUserMedia = (navigator.getUserMedia ||
      navigator.webkitGetUserMedia ||
      navigator.mozGetUserMedia);

    // Some browsers just don't implement it - return a rejected promise with an error
    // to keep a consistent interface
    if (!getUserMedia) {
      return Promise.reject(new Error('getUserMedia is not implemented in this browser'));
    }

    // Otherwise, wrap the call to the old navigator.getUserMedia with a Promise
    return new Promise(function(resolve, reject) {
      getUserMedia.call(navigator, constraints, resolve, reject);
    });
  }
}

navigator.mediaDevices.getUserMedia({ audio: true, video: true })
.then(function(stream) {
  var video = document.querySelector('video');
  // Older browsers may not have srcObject
  video.src = window.URL.createObjectURL(stream);
  video.onloadedmetadata = function(e) {
    video.play();
  };
})
.catch(function(err) {
  console.log(err.name + ": " + err.message);
});
```

Fig. 3: Determining Cross-Browser Compatibility

Paste your text right here and click "Next" to watch this article rewriter do it's thing. Have no textual content to check? Have no textual content to check? Click "Select Samples". In the HTML5 code, the canvas and video tags are used to display the feed of the web digital camera as properly as to manipulate the pix from the frames.

To obtain the photo we would put an empty picture tag with a button below it that reads “Take Screenshot”. Thekey HTML aspects would be:

<video autoplay="" style="display: none"></video>

<canvas style="display: none;"></canvas>

<imgsrc="" id="screenshot">

When the user clicks on the button, a hidden video tag will be activated primarily based on JavaScript tournament listeners that are connected to the button to procedure this. The frames from this video tag will be passed to a canvas tag that can manipulate every frame as a picture to add effects or to store the photograph and use it someplace else on the page.

In our application, the first frame is taken as a photo and surpassed from the canvas issue to the photograph element. The user can then figure out to use it for facial cognizance or retake some other screenshot. The chosen photo would then be processed by using the server to detect the face and to authenticate or deny the user.

4. Detecting and Recognizing Faces

The Facial Recognition API from Lambda Labs is used in our software for the facial identification and cognizance component. This API may additionally be used for free with a limited range of detects and recognition of face, which is excellent for our prototype software modules. In addition to being capable to perform face detection and recognition, this API may also also perform facial function extraction, (of the eyes, nose, and mouth) and gender classification. The downloaded patron library can be used in languages such as Python, Objective C, PHP, and Java.

The API uses an album which can be stuffed with a restrained wide variety of pix for focus and another limited set of pix for detection. The album has to be skilled to recognize one label with a couple of pictures related with it. Pictures positioned into the album can be either URLs or files. Each photograph has to be tagged, labeling whom the picture represents.

The more snap shots are in the album for each single individual, the more correct detection and attention it will have. PHP was adopted as the programming language to improve our facial detection and cognizance module, with Lambda Labs Face Recognition and Detection API as its supporting package. The facial cognizance is used to confirm that the photograph the person uploads suits a person's saved template. Part of the PHP code is proven in Fig. 4

```php
<?php
require_once("mashape/MashapeClient.php");


class FaceRecognition {
    const PUBLIC_DNS = "lambda-face-recognition.p.mashape.com";
    private $authHandlers;

    function __construct($publicKey, $privateKey) {
        $this->authHandlers = array();
        $this->authHandlers[] = new MashapeAuthentication($publicKey, $privateKey);
```

```
}
public function createAlbum($album) {
    $parameters = array(
            "album" => $album);

    $response = HttpClient::doRequest(
            HttpMethod::POST,
            "https://" . self::PUBLIC_DNS . "/album",
            $parameters,
            $this->authHandlers,
            ContentType::FORM,
            true);
    return $response;
}
public function detect($files = null, $urls = null) {
    $parameters = array(
            "files" => (($files == null) ? null : ('@' . $files)),
            "urls" => $urls);

    $response = HttpClient::doRequest(
            HttpMethod::POST,
            "https://" . self::PUBLIC_DNS . "/detect",
            $parameters,
            $this->authHandlers,
            ContentType::MULTIPART,
            true);
```

Fig. 4: Eemploying Facial Recognition in PHP

6. Conclusion

Face identification and attention have grown to be a good deal extra accessible utility component as cellular gadgets and computer systems are equipped with cameras. It is turning into a natural and secure biometrics feature for user authentication. The new HTML5 and get User Media () API from W3 consortium have empowered Web applications to request permission from users and to use their Web cameras, which therefore can employ facial cognizance primarily based services. In this paper, we first survey well-known techniques and algorithm for facial recognition. Then we describe our exercise and experiments in integrating facial awareness carrier into a Web-based distance mastering environment, which make use of a combination of strategies such as HTML5 features, get User Media() JavaScript API, and Face Detection and Recognition API from Lambda Labs. We additionally describe how facial recognition modules and tools may additionally be integrated into modern mastering administration systems the usage of open and standardized programming paradigm and interfaces.

REFERENCES

[1] Blanch-Hartigan, Danielle, Susan A. Andrzejewski, and Krista M. Hill. "12 Training people to be interpersonally accurate." The social psychology of perceiving others accurately (2016): 253.
[2] Liao, Weidong, and Chad Vanorsdale. "Facial Recognition and Its Applications in Distance Learning Environment." Journal of Computer Science 4.2 (2016): 1-14.
[3] Allen, I. Elaine, and Jeff Seaman. "Online Report Card: Tracking Online Education in the United States." Babson Survey Research Group (2016).
[4] http://onlinelearningconsortium.org/read/online-report-card-trackingonline-education-united-states-2015/. (October 15, 2016)

[5] Liao, Weidong, and Chad Vanorsdale. "Facial Recognition and Its Applications in Distance Learning Environment." Journal of Computer Science 4.2 (2016): 1-14.
[6] Carcagnì, Pierluigi, et al. "Facial expression recognition and histograms of oriented gradients: a comprehensive study." SpringerPlus 4.1 (2015): 645.
[7] Latha, P., L. Ganesan, and S. Annadurai. "Face recognition using neural networks." Signal Processing: An International Journal (SPIJ) 3.5 (2009): 153-160.
[8] Turk, Matthew A., and Alex P. Pentland. "Face recognition using eigenfaces." Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE, 1991.
[9] Jenkins, Rob, and A. Mike Burton. "100% accuracy in automatic face recognition." Science 319.5862 (2008): 435-435.
[10] Sofi, Sameer Sadiq, and Rafi Ahmad Khan. "A Review of Face Recognition Techniques." Digital Image Processing 8.4 (2016): 117-120.
[11] Jafri, Rabia, and Hamid R. Arabnia. "A survey of face recognition techniques." Jips 5.2 (2009): 41-68.
[12] Zhao, Wenyi, et al. "Face recognition: A literature survey." ACM computing surveys (CSUR) 35.4 (2003): 399-458.
[13] Liao, Weidong, and Chad Vanorsdale. "Facial Recognition and Its Applications in Distance Learning Environment." Journal of Computer Science 4.2 (2016): 1-14.
[14] Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015.
[15] Heberling, Michael. "Maintaining academic integrity in online education." Online Journal of Distance Learning Administration 5.2 (2002).
[16] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.
[17] Farmer, James, and Ian Dolphin. "Sakai: eLearning and more." EUNIS 2005-Leadership and Strategy in a Cyber-Infrastructure World (2005).
[18] Rivera, Luis Felipe Zapata, and Maria M. Larrondo Petrie. "Models of collaborative remote laboratories and integration with learning environments." International Journal of Online Engineering (iJOE) 12.09 (2016): 14-21.
[19] Huertas, Francisco, and Antonio Navarro. "Integration Mechanisms in e-learning Platforms." International Journal of Computer Information Systems and Industrial Management Applications 5 (2013): 714-721.
[20] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.
[21] Patel, Vishal M., et al. "Visual domain adaptation: A survey of recent advances." IEEE signal processing magazine 32.3 (2015): 53-69.