# Capstone Project 2021/22

**Project Title:** Develop your web with biometric identification

**Autumn term supervisor:** Javier Andreu-Perez

**Spring term supervisor**: Junhua Li

**Summer term supervisor**: Alba Garcia

**Second accessor**: Manoj Thakur

**Project Description:**

In this project the student will develop a web where their users are identified using biometrics rather than user and passwords. These can be done by mean of computer vision techniques such as facial recognition, voice or keystroke identification. The student can use any web programming languages for this endeavor PHP, Node.JS, python or java.

You are free to use any web technologies or libraries to support this project. You can do some research about what is already out there.

## Project objectives

The main objective of the project was the creation and development of a website where the users are identified using biometrics rather than username and password.

## Implementation plan:

The system will be written mainly using JavaScript and Node.js.

An open-source facial recognition system created by Vincent Muhler Face-api.js will be used to perform the user face identification process. Data about users and their biometrics will be stored in a database.

1. Write a blank, basic website.
2. Connect website to a local server.
3. Add login ability to the page
4. Add register ability to the page
5. Implement the face recognition system.
6. Create user login page / window that will ask user for an access to the web camera and perform user identification process.
7. Create a database.
8. Create user registration page / window that will ask user for an access to the web camera, ask user for their personal data, take user’s biometric measurements, and save data to database.
9. Add logout ability to the page.
10. Add ability for a user to change their personal data.
11. Add ability for a user to delete their account/personal data.
12. Design the website to be user-friendly and easy to use.

## challenge week

**My challenge:** Create formulation of project objectives, management, implementation plan, requirements, and user stories. Start coding.

What will the project achieve?

Development of a secure website where the users are identified using biometrics (?*two step biometric?*) rather that username and password. The system will be written mainly using JavaScript and Node.js. An open-source facial recognition system created by Vincent Muhler Face-api.js will be used to perform the user face identification process. Data about users and their biometrics will be stored in a database.

What are the risks?

* incorrect user identification and gaining access to the data by unauthorized / unregistered users.
* denials of service
* no web camera
* blurry video from a camera
* system incorrectly identifies an impostor as an authorized user
* spoof attacks at the user interface
* template database leakage

**Implementation plan:**

1. Write basic website.
2. Connect website to a local server.
3. Write user friendly page.
4. Create a database.
5. Implement the face recognition system.
6. Create user login page / window that will ask user for an access to the web camera and perform user identification process.
7. Create user registration page / window that will ask user for an access to the web camera, ask user for their personal data, take user’s biometric measurements, and save data to database.
8. Add login ability to the page
9. Add logout ability to the page.
10. Add ability for a user to change their personal data.
11. Add ability for a user to delete their account/personal data.

## To be considered:

* Issues of biometrics and how to prevent them (e.g. someone try to login with a video of an authorized user)

### Requirements for website with biometric identification

Development of a secure website where the users are identified using biometrics (facial recognition) rather than username and passwords, using web camera and face recognition system.

Functional Requirements:

1. System allows users to register, log in, delete personal data, change personal data, and log out.
2. System able to access web camera.
3. System able to carry on a facial recognition from a video from a live web camera.
4. System able to recognize users with as little error margin as possible.
5. System able to store secured data about users in database.
6. System able to read secured data about users from a database.

### User stories:

* New user > registration with biometric > system remembers their data
  + System opens web camera
  + Takes 3 pictures, asks user to look straight to camera, right and left
  + Asks to enter their name, email address, password in case face recognition is impossible
  + Save data to the database
  + Log in the user
* Registered user > log in > system authenticate them
  + System opens web camera
  + Carry on face recognition with live stream video from camera
  + Check if the database contains data of the user
  + If user exists in the database, log in
  + If user is not in the database, return a message
* Registered user > change personal information > keep data secure
* Authenticated user > log out
  + End the session
* Registered user > delete an account > data removed

## Face recognition system

**Pipeline:**

1. Face detection
2. Face alignment
3. Feature extraction
4. Face matching

To might be used:

* OpenFace
* Python
* JavaScript
* Node.js
* Face API
* Face-api.js
* MTCNN (5 facial landmarks: eyes, nose, mouth corners)
* OpenCV.js (popular one, for object detection, face detection and object tracking, image resizing, color model transformation, similarity transformation, too heavy for website)
* Face-api.js (optimized for web browser side recognition)

Faces of users in most papers were stored as images, why not store them as results of measurements, 128 vector?

Best to use deep learning techniques AI

## Database Storage

The matched features of respective individuals need to be maintained in order to avoid repetitive registration.

All info should be secured.

To reduce the risk of the data being breached, it should be encrypted when transferring over the network. The issue with encryption is deciding where encryption keys will be stored and who will be trusted with access.

Keep in mind however, that even if the system is breached and the biometrics “harvested”, without the secret and proprietary algorithm, the biometric data cannot be interpreted.

### Pros of biometric:

* No need to remember anything
* Works with the camera and microphone on any modern computing device
* Faster, simpler, and more secure than one-time passwords, which are compromised as soon as a fraudster compromises your email or your device
* Immune to social engineering; can’t be sold on the Dark Web
* Prevents credential sharing in the workplace
* Usable across communication channels
* Blocks common fraud vectors, like SIM swaps and device spoofing

## Background RESEARCH

|  |  |
| --- | --- |
| Query | ("biometric identification" OR biometrics OR "biometric authentication") AND ("facial recognition" OR "facial identification" OR "face recognition" OR face OR recognition OR facial) AND (website OR web OR websites OR site OR sites) AND (Node.js OR java OR python OR javascript OR PHP OR js) |
| Visualization + share link | Graphical user interface, application  Description automatically generated  https://app.2dsearch.com/new-query/611e42b7fa16660004cf130b |
| Google Scholar | Searched on 19/08/2021: 10,600 results |
| ACM | Searched on 19/08/2021: 520 results |
| Web of Science | Searched on 19/08/2021: 9 results |
| CiteSeer | Searched on 19/08/2021: 0 results |
| Bing | Searched on 19/08/2021: 4 220 000 results |
| IEEE | Searched on 19/08/2021: 24 results |

New query

“user” AND “biometric” AND “authentication”

“user” “biometric” "database" – for storing biometric data in database

## PAPERS Review

**Title:** Web based Biometric Validation Using Biological Identities: An Elaborate Survey

**Link:** <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8480939>

* Secured storage
* “As a result the held back models need fortification in holding the delicate information to avert any sort of seepage.”

**Title**: Cloud Based Face Recognition for Google Glass

**Link**: <https://dl.acm.org/doi/pdf/10.1145/3194452.3194479>

* Cloud safe storage
* One of a way for facial recognition is by doing comparison of designated facial features from the picture and a face record database
* **Steps to implementation** (open the camera, system try to detect face, when face detected surround it with square box, take a picture, send image to server to recognize person)
* Eigenface

“Face recognition system plays a vital role in several applications such as person identification, face tracking, video surveillance and human computer interaction. Algorithm for effective face and facial characteristic detection are required for applying to those tasks.”

“Face recognition is done by comparing digital image captured by camera or from video frame with the existing image in database.”

Algorithms for face recognition:

* Dr. Bonsor used shape and size of nose, eyes, jaw, cheeks and their relative position in the algorithm. [20].
* Principal Component Analysis via eigen faces [21],
* Elastic Bunch Graph Matching that uses Fisher face algorithm [22].
* face feature is extracted by Local Binary Pattern Histogram (LBPH) [6].
* as“Lambda “,
* “Google’s Picasa”,
* “Apple’s iPhoto”.
* There is a three-dimensional (3D) face recognition trend with improved accuracies because it data is pose variant and consist of more facial surface information [23]
* A comprehensive pose invariant automatic system was presented by, Prof. Loannis [24]. According to design, registration of 3D facial scan was done by using composite alignment technique. As compared to 2D systems, 3D face recognition system achieves higher accuracy. But 3D system is much expensive alongside computational delay due to which has less practical implementation. Furthermore, 3D face recognition requires subject’s cooperation to capture data from 3 dimensions that delays the process.
* The author [25] used OpenCV for face detection due to open source and highly supportive behavior for Eigen faces, Fisher faces and Local Binary Patterns Histograms (LBPH). Eigen faces [26], [27] demonstrates illumination-robust property reliable under multiple illuminations. Eigen faces approaches also support to update face recognizer due to which it lessens the running delays and memory space.

“**Usability**: System shall be user friendly that any type of user shall learn to operate the system after a short training. The interface by which user interact should easy to navigate throughout the system.

**Reliability**: System shall be reliable in its best capacity to detect face and recognize the person in it.

**Availability**: System availability is the time when the application must be available for use.

**Recovery**: The system shall be able to recover or heal up when an error occurs. If it does not heal then it should at least display an error message.

**Performance**: System should be designed such that it does not need much time to load pages. Develop each Interface page to be fast-loading. All images, graphics and multimedia should be optimized to the appropriate size and quality to ensure that they load within a few seconds.

**Supportability**: System should be compatible for viewing across a wide range of visitor platforms and browsers. It should be compatible with following browser versions: Safari, Internet Explorer 10, Firefox 35, and Chrome.”

**Title**: A Face Recognition System for Assistive Robots

**Link**: <https://dl.acm.org/doi/pdf/10.1145/3378184.3378225>

“It is typically split into simpler sub-problems or steps, namely: **face detection**, **feature extraction**, and **face classification**. Indeed, a popular pipeline for FR during years incorporated the Viola Jones classifier [29] for face detection, and eigenspace based methods to project the detected faces onto a low-dimensional space, where recognition is carried out by checking their distance to known faces”

“the creation of huge public datasets [3, 10, 14], new loss functions applicable to FR [6, 17, 25, 30], or the use of multiple network architectures [11, 26, 27]. Such advances have steadily increased the state of the art performance in public benchmark datasets, e.g. improving accuracy from a 97.35% to 99.83% in just 3 years on the LFW repository [6, 12, 28]. Nevertheless, which algorithm to use in each step of the FR pipeline strongly depends on the requirements of the target application.”

open-source implementations, including MTCNN [32], OpenCV’s DNN [2], and OpenPose [4] for face detection, and InsightFace [6] and Facenet [3] for feature extraction. For evaluating the formers we have resorted to FDDB [13], while for the latters we made use of LFW [12], the de facto benchmark dataset for face recognition. Accordingly to the reported results, we propose a FR system implemented in the Robot Operating System (ROS) [22], ready to be used by assistive robots, and test it in a real robotic platform. Finally, we have made public our contribution to the robotics community at <https://github.com/samuelbaltanas/face_recognition_ros>.

FACE RECOGNITION PIPELINE:

1. a face detector (which serves to extract all human faces from the initial image),
2. face alignment (using fiducial key points to normalize pose variability),
3. a feature extractor (which extract discriminative facial information for comparison purposes)
4. final classification step (comparing a detected individual to a dataset of known identities).

Graphical user interface, application

Description automatically generated

**Detailed description of each pipeline step for face recognition system** [**pdf**](Papers/3378184.3378225.pdf)

**Title**: On the Effectiveness of Low-Cost Face Recognition with Deep Learning

**Link**: <https://dl.acm.org/doi/pdf/10.1145/3437120.3437275>

“deep learning approaches are a reliable solution for face recognition even with low-cost apparatus”

Used OpenCV library

“The systems take as input video or an image from a live feed camera. Then the face detections procedure follows. The system is trying to find a match, of the face that was spotted, in the database. In this stage of the facial recognition process there are different approaches as we mentioned above. In our case we are using deep metric learning algorithms.”

“The deep metric learning that we used in our work implements the following method: it assigns a valued feature vector to the image. This process is done using the dlib library, created by Davis King”

“The facial recognition network provides as an output a **128-d vector** with real valued numbers. This **vector is then used to quantify the input face**. The **training of this type of network is done using triplets**. To be more specific, the deep learning network is trained based on three face images. Two of the images belong to the person for which we want to train the network and the third one needs to be from a random person. Then, the neural network generates the 128-d vectors, one for each image, and afterwards the network for the images that belong to the same person, tweaks the neural network weights in order to make the distance of the vectors closer, metric wise. With this process we have successfully quantify the face of its person in our database and these vectors that we have created can be used for reference”

Title: OpenFace: an open source facial behavior analysis toolkit

Link: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7477553>

Title: Web Front-End Realtime Face Recognition Based on TF.JS

Link: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8965963>

“Face-api.js is developed by Vincent Muhler. Based on TF.JS, it provides a lot of out-of-box models and APIs for face recognition[24]. It’s optimized for web environment and easy to use.”

Workflow:

Firstly, the workflow requires a video stream as an input. And it will be loaded via a HTML <video> tag. As the video starts to play, the workflow loops to process frames.

To capture a frame, we utilize HTML5 Canvas API. The canvas is a web-based equivalent to Java’s java.awt.Graphics2D, which enables developers to render and manipulate 2D shapes, text and images on browser-side. Also, they have similar APIs, such as drawSometing() for drawing objects and fillSometing() for fill objects. And here we use drawImage(obj, x, y, w, h), where obj can be an image or a video. If a video is passed to drawImage, it will draw the immediate frame of the video. Also, we can resize the frame to a smaller size by specifying parameter w and h and here we choose 640x360. Compared to common video resolution 1920x1080, this significantly reduces pixels passed to CNNs, which ensures speed of the process.

Afterwards, we run SSD-MobileNetV1 model provided by face-api.js to detect faces on the captured frame, where SSD is the object detector described before. Originally SSD uses VGG16 as its backbone network, but here VGG16 is replaced by MobileNetV1, another CNN backbone network developed by Google. Compared to VGG16, MobileNetV1[25] is optimized for mobile devices by employing a trick called depthwise separable convolution, which reduces amount of computation.

Then the face alignment is applied to the detected faces. Here we use a tiny yet accurate 68-point face landmark predictor provided by face-api.js. The whole model is ~200kb, which is suitable for web environment. Afterwards, we align predicted 68-point landmarks and the standard landmarks. Then we have aligned faces for face recognition.

To do face recognition, a ResNet-34 like architecture is implemented to compute a face descriptor. The face descriptor is a vector of 128 values, representing a face in the 128 dimensional vector space. Also, the euclidean metric of different vectors represents their similarity. Then we can match each face shown in the video frame with pre-calculated fingerprints of each person, using naive KNN(k-nearest neighbor) algorithm.

Last, we visualize the results on the screen, utilizing the Canvas API, again. We put the original video frame on the background and the labeled bounding box of each detected face on the foreground. Then, it loops back to capture the next frame if the video is still playing, or the workflow comes to an end.

Link: <https://d1wqtxts1xzle7.cloudfront.net/55078028/Face_Recognition-survey-with-cover-page-v2.pdf?Expires=1630579157&Signature=LC3vExPmObtGT9KR~c9iQSmBOjavmIIKntG8VDD9FrHO6DN44ye2o8xxvJN2U31nY0BBQYLeX981gtBatxNyQUDynLfimCNUdUuSi2zbue3AhtwFYE5y8uimanMqCZBfLRXIvZU9fO83yechp~fzqCt0tyf4wYR1xBWHRt36o45ZW0K1AHM6hMUmOgC2Wmu3XdFJF47rZQFwy5dQDCs1Gn94LWTuPK2uL-T41E~9sZb4wuI5QwxRHlgJVqg7KcAYESTumHonNT7UPVysrOh3LZoyCLV5Tfl3f6II3wAvcx0gNLwvJ7fh9wLtz-HH2kbSQfnznxt9WZf~ArfZDSCl2g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>

Overall about face recognition systems, compares different methods of face recognition.

Title: Biometric Authentication: System Security and User Privacy

Link:<http://biometrics.cse.msu.edu/Publications/SecureBiometrics/JainNandakumar_BiometricAuthenticationSystemSecurityUserPrivacy_IEEEComputer2012.pdf>

Brief: vulnerabilities of biometric systems, safety, attacks, biometric template security requirements

“biometric recognition is a natural and more reliable mechanism for ensuring that only legitimate or authorized users are able to enter a facility, access a computer system, or cross international borders. Biometric systems also offer unique advantages such as deterrence against repudiation and the ability to detect whether an individual has multiple identity cards (for example, passports) under different names. Thus, biometric systems **impart higher levels of security when appropriately integrated into applications requiring user authentication**.”

“biometric system **can be circumvented by a skillful impostor given the right circumstances and plenty of time and resources**. Mitigating such concerns is essential to gaining public confidence and acceptance of biometric technology”

**BIOMETRIC SYSTEM OPERATION**

* records a sample of a user’s biometric trait using an appropriate sensor—for example, a camera for the face— during enrollment
* extracts salient characteristics, such as fingerprint minutiae, from the biometric sample using a software algorithm called a feature extractor.
* The system stores these extracted features as a template in a database along with other identifiers such as a name or an identification number.

“To be **authenticated**, the user presents another biometric sample to the sensor. Features extracted from this sample constitute the query, which the system then compares to the template of the claimed identity via a biometric matcher. The matcher returns a match score representing the degree of similarity between the template and the query. The system accepts the identity claim only if the match score is above a predefined threshold”

“A biometric system is vulnerable to denials of service (system doesn’t recognize a legitimate user) and intrusions (system incorrectly identifies an impostor as an authorized user), which can be caused by both intrinsic limitations and adversary attacks.”

“An adversary can circumvent a biometric system by coercing or colluding with insiders, exploiting their negligence (failure to properly log out of a system after completing a transaction), or fraudulently manipulating the procedures of enrollment and exception processing, originally designed to help authorized users”

Fail through direct attacks on the user interface (sensor), the feature extractor and matcher modules, the interconnections between the modules, and the template database

“Two major vulnerabilities that specifically deserve attention in the context of biometric authentication are **spoof attacks** at the user interface and **template database leakage**.”

“numerous liveness detection techniques—for example, verifying the physiological properties of human fingers or observing involuntary human actions such as blinking of the eye—to ensure that the biometric trait captured by a sensor indeed comes from a live person”

“it isn’t possible to replace stolen templates with new ones because biometric traits are irrevocable”

**Acceptable tradeoff among three requirements:**

**Noninvertibility**. It must be computationally hard to recover the biometric features from the stored template. This prevents the adversary from replaying the biometric features gleaned from the template or creating physical spoofs of the biometric trait.

**Discriminability**. The template protection scheme shouldn’t degrade the biometric system’s authentication accuracy.

**Revocability**. It should be possible to create multiple secure templates from the same biometric data that aren’t linkable to that data. This property not only enables the biometric system to revoke and reissue new biometric templates if the database is compromised, but it also prevents cross-matching across databases, thereby preserving the user’s privacy.

Diagram

Description automatically generated

**Researchers have proposed two main approaches for generating a secure sketch: fuzzy commitment and fuzzy vault.** Fuzzy commitment can be used to protect biometric templates that are represented as fixed-length binary strings. The fuzzy vault is useful for protecting templates that are represented as a set of points

Biometric Cryptosystems: Issues and Challenges

Link: <Papers/Biometric_cryptosystems_issues_and_challenges.pdf>