

SuspectSearch: AI-Powered Criminal Identification System

*This is a Senior Project Report paper

Batyrkhan Abukhanov
dept. Computer Science
Nazarbayev University
Astana, Kazakhstan

Tamerlan Abaideldinov
dept. Computer Science
Nazarbayev University
Astana, Kazakhstan

Maxim Tsoy
dept. Computer Science
Nazarbayev University
Astana, Kazakhstan

Bibarys Gilazh
dept. Computer Science
Nazarbayev University
Astana, Kazakhstan

batyrkhan.abukhanov@nu.edu.kz tamerlan.abaideldinov@nu.edu.kz maxim.tsoy@nu.edu.kz bibarys.gilazh@nu.edu.kz

Abstract—The traditional approach to criminal investigation, taken in the context of rapidly developing technological advancements, presents itself to be rather outdated. One of the most crucial parts of identifying a suspect is criminal sketching. The modern process requires manual sketch craftsmanship from assets based on the description of the suspect in order to approximate the appearance of the culprit. This method is highly limited and burdensome, it requires a considerable amount of time and resources that could be used in other areas of law enforcement.

The proposed system utilizes machine learning techniques to approach this problem and suggests an alternative way of identifying criminals. The system analyzes photos uploaded to the database and evaluates pre-defined metrics to synthesize a vector for a person that will be used to describe the appearance in abstract terms. The system then takes prompts from the intended users, which could be victims or witnesses, to synthesize a new vector for each prompt and will find the most corresponding photos from the database.

The results of testing the product indicate decent accuracy and usability and propose a direct search from the database of people instead of a traditional empirical search from a manual approximated sketch. This method will remove the necessity for the time-consuming manual sketching process and further search for an imprecise image of a suspect. A simple and intuitive in-use interface will not require any special training of the police staff in order to integrate the product into the current system of law enforcement.

The project might be limited as of today, but given enough time and resources, several additions will drastically improve the effectiveness of criminal investigation and contribute to the redistribution of the police force on other important tasks.

Index Terms— *Facial Metrics, Criminal Identification, Machine Learning, KNN, Law Enforcement, Identikit*

I. INTRODUCTION

Today, the traditional method of criminal identification is commonly utilized in police departments of the Republic of Kazakhstan. This process begins with the detailed description of a potential suspect including facial attributes and distinctive features given by the witness of the crime. The professional artist or drag-and-drop software application converts verbal descriptions to a sketch that will be used by police officers to identify a suspect [9].

This procedure is considered to be time-consuming and ineffective as it heavily relies on the competence of the artist and the human ability to recall the facial features of the suspect. According to Tyng et al., an individual capability to recall details can be affected by stress or trauma, thus it could obstruct the process of criminal identification, particularly when relying on the sketches created by hand [7].

Our main goal is to create a possibility to search for criminals by their actual pictures which is more efficient than the identikit tool. The proposed algorithm extracts seven facial metrics, encompassing measurements of the left and right eyes, eyebrows, nose length, nose height, and lips size. Subsequently, these metrics undergo normalization based on their respective sizes or areas. The normalized metrics are then assembled into a seven-dimensional feature array, which serves as input for our search algorithm. Employing the K-Nearest Neighbors (KNN) method, the search algorithm identifies the closest matches from the user's provided description, which is also transformed into a seven-dimensional normalized feature array.

As the platform of our solution, we decided to use a web application, since it's fast, simple for users, and does not require anything but a browser. The witness's testimony is planned to be first recorded on the website. Afterward, it will be processed on the server, which is connected to the convicted criminals database. This database could be populated by administrators who have access to our algorithms.

The statistics indicated by Professor Chukmaitov state that approximately 50% of individuals who commit a crime are likely to re-offend - a fundamental concept in criminal justice called recidivism. Of those who have completed their first prison term, 40.4% commit new crimes within 1 year after release, 21.3% - from 1 to 2 years inclusive, 13.4% - from 2 to 3, and 75.1% of these individuals re-offend within 3 years of their release [3]. Given the statistical likelihood of recidivism, our platform serves as an important tool in criminal identification by enabling police departments to match new crime data with potential suspects who have previous records. This functionality increases the effectiveness of investigation

procedures and public safety by automated recognition.

This paper will explain the foundation of the established SuspectSearch web-based system. Also, it will describe the implementation of the Machine Learning algorithm, processes behind the textual prompt analysis, as well as integration of the search method and software parts. Lastly, the evaluation of the proposed system will be included to define future work.

II. BACKGROUND AND RELATED WORK

Throughout the development of the system, it was essential to understand the current methods of criminal identification utilized within the Republic of Kazakhstan. For this reason, it was decided to refer to Vasylichikov [9] and his methods of expert research of suspects. The main technique is based on the creation of sketches by compiling different facial features based on witness descriptions. While being widely used by law enforcement agencies, this method has limitations regarding the accuracy of the resulting image. This insight was valuable since it emphasized the necessity of our solution where the efficiency of the algorithm is increased due to the use of real photographed criminal data.

The justification of our platform also relied on the statistical evidence of Chukmaitov [3], who studied the social pattern of recidivism. By analyzing the paper, we concluded that our platform should be able to accurately recognize repeated offenders. This is possible to implement owing to the fact that criminal data including photographs and biological data is routinely stored by governmental organizations.

As stated by Tyng et al [8], the psychological state of a witness can remarkably influence the investigation procedure. The source elaborates on how stress and trauma might affect the ability to recall specific facial components, especially in the hand-drawn or compiled sketch. Singh et al [7] conducted further research on the impact of emotional state on the accuracy of facial recognition and recalling crime situation details. Their experiment involving 60 participants divided into high and low-emotional state subgroups provided additional evidence supporting the theory that individuals experiencing stress are less capable of identifying specific facial components. This research was important in establishing our platform, leading us away from the sketches towards using actual criminal photographs.

Under the authority of Vasylichikov, while there are existing computer algorithms utilized in person identification, they are primarily aimed toward matching the facial features of a corpse with the photographs of a living individual. [9] Even though there is not a single project that is exactly like ours, we did thorough research to find related works that could help solve the separate parts of our project. The research analysis on the topic of face recognition and evaluation has yielded several techniques on face segmentation which our team found useful in the implementation of the system. We have developed strategies based on some of the above-mentioned techniques which included Felzenszwalb segmentation, which is a superpixel-based segmentation using the graph approach [5]. This algorithm uses a graph with pixels as nodes and

the similarity between them as edges. It runs at an almost linear time depending on the number of edges, which is highly efficient considering the amount of data that will be processed in the real-world implementation of the project. Moreover, in the lack of sufficient standardized data that will resemble the intended dataset, the Felzenszwalb segmentation did not require training which brings more benefits to its use.

With the discovery of a suitable dataset, we found a state-of-the-art technique that will allow for more accurate detection of the parts of the face - Instance segmentation [6]. Instance segmentation described in this paper combines Semantic segmentation with Object detection and allows for easier manipulation with the labeled classes of segments found in the picture. This method requires training and is computationally more expensive to use than Felzenszwalb segmentation. However, it provides qualitatively better results in segmenting an image and computing the yielded segments for further use.

Research on Natural Language Processing and Computer Vision techniques contributed to the majority of time spent on this project. Among various discovered approaches, most of the time was spent on learning and adapting the BERT model for text processing [4]. From the paper on this architecture, the base model pre-trained to understand the English language was discovered and utilized with utmost attention. Although the model was effective on various tasks, the system's current design did not leave options for further consideration.

A big leap and major hint for text processing were discovered in word embeddings, described in the paper "Enriching Word Vectors with Subword Information" [2]. Information from this paper gave a gist on word embeddings and provided a model trained on a large corpus of texts, which opened a new approach to the problem and was implemented as the final solution.

III. PROJECT APPROACH

A. Overview

Overall, the project consists of two major functionalities. The first functionality is an extraction of features from the picture. As indicated in Figure 1 below, administrators have access to import new criminals into the database. Once the upload button is pressed it starts the Feature extraction algorithm. This algorithm identifies seven facial metrics, which are nose length, nose size, right and left eyebrow sizes, right and left eye sizes, and mouth size. Then it calculates the areas or sizes of the above-mentioned metrics. Finally, these metrics are normalized and stored as 7-dimensional feature arrays (e.g. [0.7, 0.3, 0.5, 0.5, 0.7, 0.7, 0.4]) in the database. These 7-dimensional feature arrays are then used in the search algorithm.

The second main function is a search algorithm. The workflow is depicted in Figure 2. As indicated in Figure 2, policemen will have access to run a search through criminals' databases. There are two possibilities for inserting facial descriptions, textual and manual with sliders. If the textual description is used, then it runs the NLP algorithm automatically to convert text into a 7-dimensional feature array. Manual

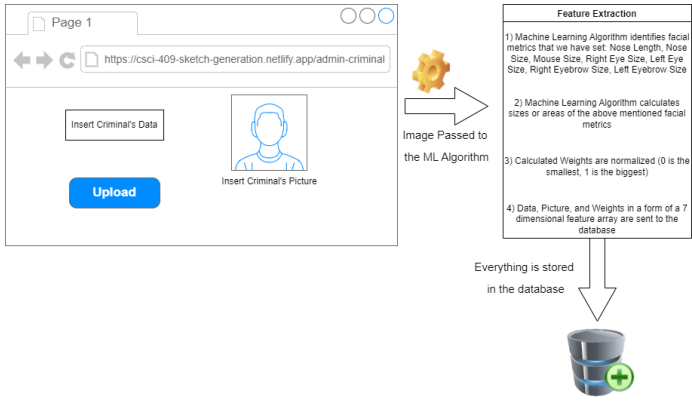


Fig. 1. Workflow: Insert

search does not require NLP because the backend receives normalized values of each metric directly from the frontend. Then, the search algorithm searches for the closest matches from the database and returns them to the user.

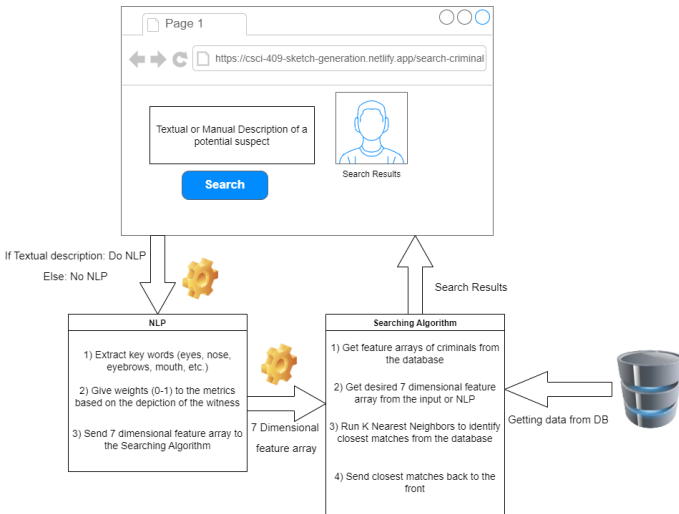


Fig. 2. Workflow: Search

B. Machine Learning

The core functionality of facial metrics estimation and evaluation was done using a machine learning approach. The facial feature extraction process could be divided into two parts: inferring the face landmarks on the provided photo and computing the desired metrics from the mesh of landmarks. A deep learning model extracts crude facial features and marks the landmarks on the photo based on a template that the model was trained on. The output of 468 landmarks is demonstrated in Figure 3.

The next step in evaluating the facial features is calculating the metrics of the desired areas. This step involves the calculation of the selected subsets of landmarks and the length/area they are referring to. Output at this stage does not represent the absolute values in physical units of measurement but rather the abstract value relative to the overall area of the face. The



Fig. 3. Facial Landmarks (Image retrieved from CFD)

computation process is comprised mainly of working with the coordinates of landmarks and computing the distance between them or the area that the points enclose. For the latter one, the Shoelace formula, presented below, was proven to be efficient due to variance in the shape and areas of the detected polygons.

The Shoelace formula is given by:

$$A = \frac{1}{2} \left| \sum_{i=1}^{n-1} (x_i y_{i+1} + x_n y_1) - \sum_{i=1}^{n-1} (x_{i+1} y_i - x_1 y_n) \right|$$

$$A = \frac{1}{2} \left| (x_1 y_2 + x_2 y_3 + \dots + x_{n-1} y_n + x_n y_1) - (x_2 y_1 + x_3 y_2 + \dots + x_n y_{n-1} + x_1 y_n) \right|$$

where $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ are the coordinates of the vertices of the polygon.

C. Natural Language Processing

The Natural Language Processing technique was implemented to extract feature vectors from the textual description of the suspect, and the output was used to match the criminals with their feature vectors extracted from photos. The main idea behind the vectorization and information extraction from a crude text description was to use the pre-defined array of keywords that indicate a certain degree of sentiment description of size and then compute the cosine similarity of the inputted words with the array to find the most suitable analog and correlated sense of size. The pre-defined array was arbitrarily chosen based on the common feeling and frequent use context in daily life (e.g. the word "large" is bigger than "big" but smaller than "gigantic", resulting in ['big', 'large', 'gigantic']). The word embedding with a FastText model was used in word vectorization to utilize them in cosine similarity measurement further. The semantic similarity of words was computed by measuring the cosine between their vectors:

$$\text{cosine similarity}(A, B) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}}$$

where:

- A_i and B_i are components of vectors \mathbf{A} and \mathbf{B} respectively.
- n is the number of dimensions of the vectors.

The main algorithm was looking for the keywords in the prompt. Then, after filtering the adjectives, the description was computed into a vector with fixed positions that correspond to the same metrics used in the vectors from the photos.

D. Search Algorithm

First of all, values retrieved from the facial feature extraction algorithm are normalized. For the purpose of normalization, the Linear Scaling technique was used:

$$x_{\text{scaled}} = \frac{(x - x_{\min})}{x_{\max} - x_{\min}}$$

where:

- x is the original data point.
- x_{\min} and x_{\max} are the minimum and maximum values of the original data, respectively.

Then, these normalized values are concatenated into the single 7-dimensional feature array (numpy array). Each person in the database has a field that stores a normalized feature array. When a user inputs a description, the system will receive a new 7-dimensional array that will be used as a desired value. Then, our system runs the K-nearest neighbors algorithm to predict and find the closest matches from the database. The K-nearest neighbor algorithm utilizes the Euclidean distance formula given below to identify the distances between feature arrays in a space:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

where:

- p and q are points in n -dimensional space.
- $d(p, q)$ denotes the Euclidean distance between these points.

E. Frontend

Regarding the project approach of the Frontend part a hosted, deployed website was created. The site provides a user interface for interacting with the system. For accessibility purposes, the website has a modern design, light/dark themes toggler, responsive layout, and smooth transitions with animations.

The site comprises the following pages:

- Authorization page
- Plain text-based search
- Manual search
- A page for criminals and users uploading and editing
- Home and contacts pages

The website is integrated with the server and database through a REST API. It was developed using Webpack (for compilation purposes), React (the main UI library), and Netlify (the hosting service).

The website currently has two types of users: admins and policemen. The admins have the permission to search for criminals (Figure 4). On the page below, the size of the face on the right (eyes, nose, mouth) corresponds to the values entered on the left.

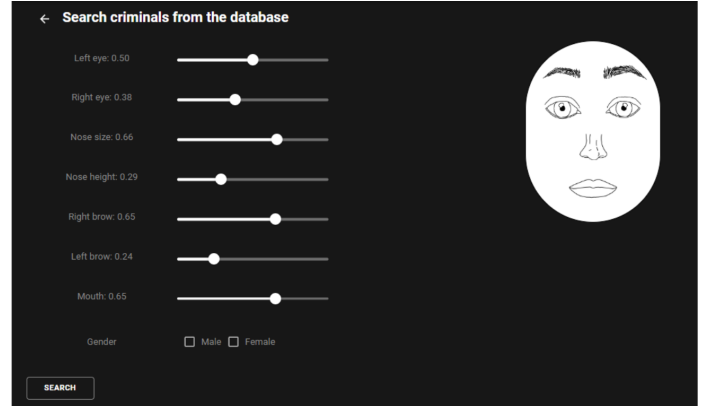


Fig. 4. Searching

The admins have the same permissions as policemen, and additionally, they could modify the list of all criminals and all users (Figure 6):

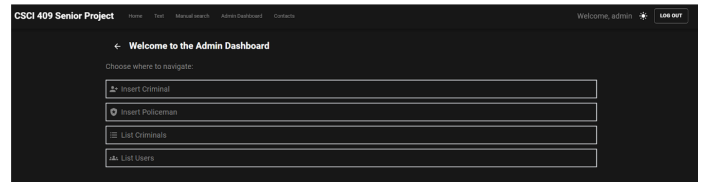


Fig. 5. Admin Dashboard

F. User Guide

The user guide is an important part of the system as it provides visual feedback, thus enhancing the user experience and accuracy during the manual input of suspect information. The user can identify the potential suspect by intuitively applying the slider and altering the size, width, and height of the facial features.

Visual Interface Design: The visual interface illustrates the person's face with facial components that can be altered by sliders (Figure 4). The custom components of the suspect face were drawn via the Krita graphical program for this purpose. By applying the slider to the left the facial characteristics decrease, conversely, moving the slider to the right results in increased parameters. This pattern is recognizable from other software applications, therefore it should be intuitive for the user.

Normalized Metrics: The metrics for the size, height, and width of the facial components are normalized to simplify the

input and alteration procedures. The values range from 0 to 1 and are considered to be a straightforward way to quantify the adjustment. The movement of the sliders changes the metrics of a facial component, leading to real-time visual feedback that enhances the accuracy of the suspect identification process.

Currently, the system supports basic shapes of facial attributes to demonstrate the manual input functionality. Later, it is planned to add more components to develop the investigation procedure.

G. Backend

For the backend part, the Django project was created and added to the GitHub repository. As we all are working on the same project it was necessary to use a version control tool. That way we can work simultaneously without slowing each other down. Moreover, the backend was deployed to the external server (PythonAnywhere).

CRUD (Create, Read, Update, Delete) functionality was implemented in Django using the REST API framework to connect with the frontend. Administrators have full access to add/delete/edit users and criminals in the database. Whereas, policemen can only read from the database.

The Authorization and Authentication functionalities were created to ensure the secure management of the system. For this purpose, it was decided to implement three main user groups - Policeman, Moderator, and Administrator, while considering the related permissions and responsibilities. Moreover, the Authentication process was planned to support Individual Identification Number (IIN) to enhance realism and security, by aligning with the governmental platforms. This number issued to every citizen of the Republic of Kazakhstan consists of 12 unique digits, commonly utilized in state organizations as it ensures user identity and permissions.

Roles and permissions:

- **Policeman:** This group is considered to be the primary user role. Its main responsibility is to input information about the suspect and manage the process of criminal identification.
- **Moderator:** The user group with additional permissions and responsibilities. They are in charge of inserting criminal information into the database and managing the Policemen. This user group has a separate page that requires permission from the system to use its functionalities.
- **Administrator:** The highest-level user role with all permissions of the platform. It has the functionality to manage all user groups including altering permissions, user roles, and the database. The Django administrator page was utilized as a control board for this group.

The Django framework supports only basic authorization by providing an email or username, that is not suitable for the system. Therefore, it was agreed to implement the custom authorization module which would utilize the IIN.

It was decided to create the authentication process based on the inheritance aspect of Object-Oriented Programming by identifying the custom primary user and related user groups.

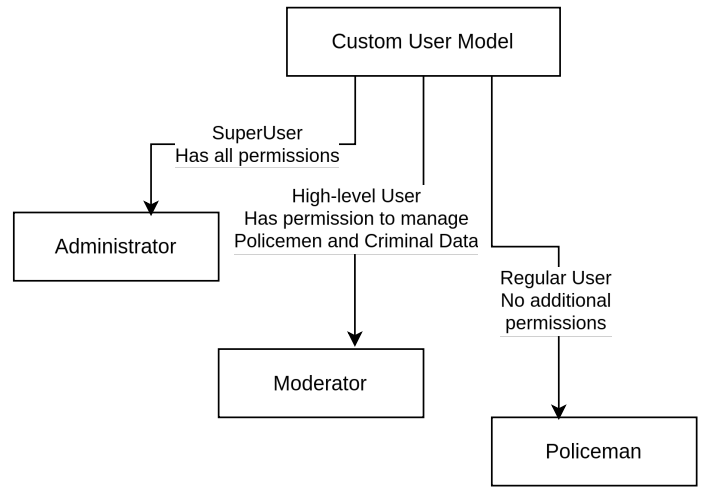


Fig. 6. User Roles

The user role and its permissions would be identified by the functions of this class.

H. Database

The database of the system is written in MySQL. It is deployed to the external private server in the PythonAnywhere cloud service. This server is protected by an SSH key. In other words, no one can externally access our database and manipulate personal data. The database was populated by the usage of a custom script. However, we did not create the images of people (dataset) by ourselves. Access to these pictures was given by the Chicago Face Database, University of Chicago.

This project consists of 5 separate tables: Criminals-Data, CriminalsImage, MaxMin, CustomUser, and AdminLog. CriminalsData and CriminalsImage have a one-to-one connection with each other as shown in the entity relationship diagram in Figure 7. These two tables are used to store criminals' data and the weights of their facial metrics. The CriminalsImage table has also a very important field - normalized_feature. This is a JSON field that stores normalized values of each previously identified field in the same table, except for iin_id and id.

In addition to the two previous tables, there is a separate MaxMin table that is isolated and is not connected to anything. It stores the maximum and minimum values of each facial metric used in our system. These values are instrumental in normalizing the sizes of facial parts and generating the normalized seven-dimensional feature array.

As this system is supposed to be used by law enforcement, it was crucial to manage permissions and access to it. Therefore, we have created a CustomUser table which is distinct from the default Django User table. This table stores the data of users (law enforcement in our case) as indicated in the entity relationship diagram in Figure 8. Most importantly it distinguishes between policemen and administrators as they have different permissions to the functionality of the application.

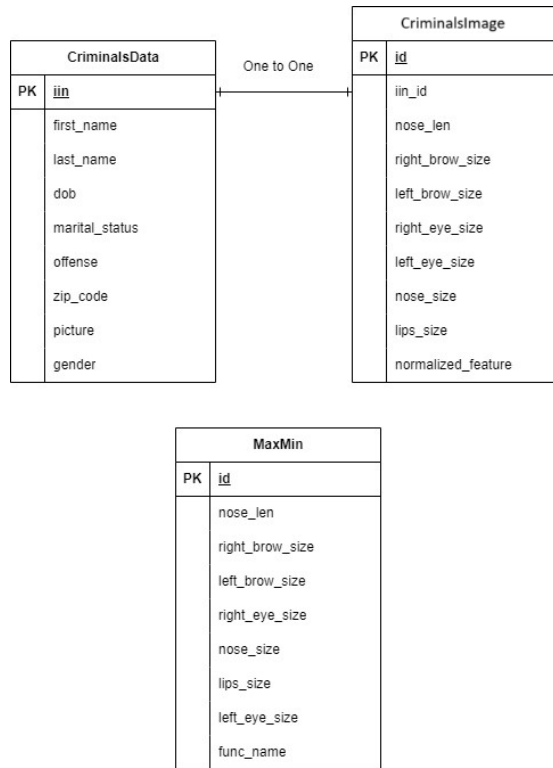


Fig. 7. Entity Relationship Diagram: Crimals Data

For example, if the role of a user is ‘policeman’ they will have access to just search for criminals. Whereas, administrators have full access to add/edit/delete both criminals and users from the system. In addition, the CustomUser table is linked to Django’s AdminLog table which allows tracking users’ activity in the system.

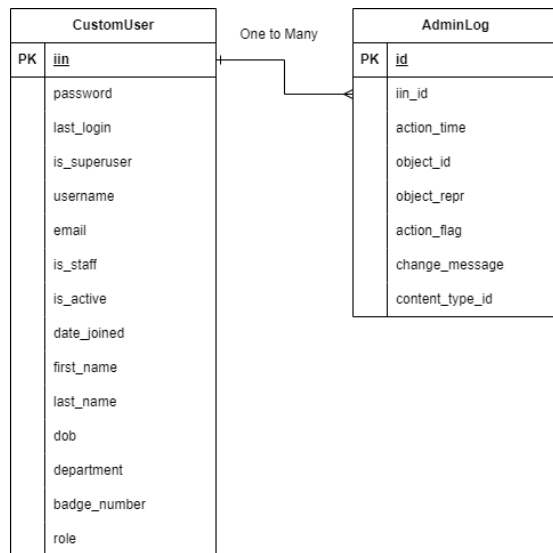


Fig. 8. Entity Relationship Diagram: Custom User

IV. PROJECT EXECUTION

A. Extracting Facial Metrics

A lot of research was done in terms of finding the right solution to our problem. First of all, we started with the Facial Keypoints Detection. We found a dataset (in Kaggle) with 15 key points on the face (see Figure 9) [1]. Then, we implemented a deep learning neural network using a pre-trained (on Imagenet dataset) Resnet50 model which demonstrated a good result (92% accuracy). However, using 15 key points is not enough for our project. We will need at least 2-3 times more key points on the face. Moreover, we could not adapt it to our solution. It is quite complicated to describe facial features (like beard, hair, eyes, etc.) This model would be more useful for detecting facial expressions or emotions, but not for describing the facial features of a person. Therefore, this method failed. Example of the data with 15 key points in Figure 9.

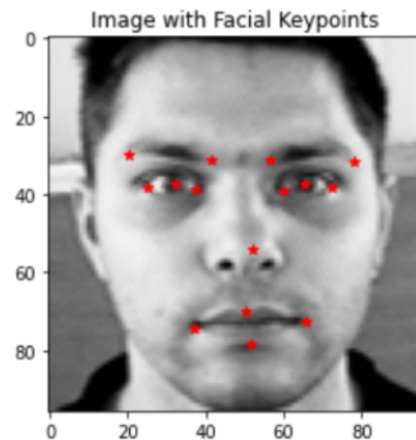


Fig. 9. Facial Keypoints Detection (Image was retrieved from public Kaggle Dataset)

Another challenge encountered during the development of a strategy for face evaluation was finding the proper dataset. The main criterion for the dataset was the format of the image, as it needed to resemble the photos on an actual ID of citizens that is standardized by size and neutral in emotion. The research indicated a short number of suitable datasets, among which the CFD - Chicago Face Database (<https://www.chicagofaces.org/>) - was the most viable option. After access to the database was granted, we were able to continue our research with enough data to work on. A sample from CFD can be seen in Figure 10.

The next approach we tried was to utilize segmentation algorithms to differentiate facial structures and classify them. Although this approach was intended to be effective due to the simple time and space complexity, the resulting segmented areas needed to be revised for further evaluation. The Felzenszwalb Segmentation was the most accurate among many segmentation models and algorithms. The algorithm converts an image into a graph and then tries to group the areas depending on a hyperparameter setting and the similarity of neighboring



Fig. 10. A sample from CFD (Image retrieved from Chicago Face Database. <https://www.chicagofaces.org/>)

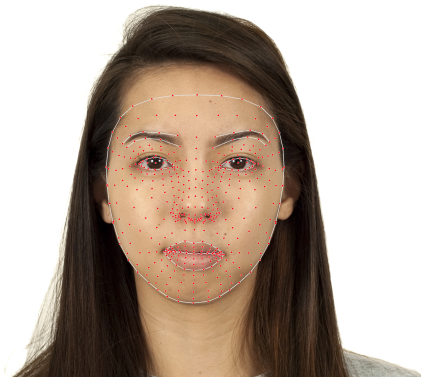


Fig. 12. Landmarks Detection (Image retrieved from Chicago Face Database. <https://www.chicagofaces.org/>)

pixels. An example of the Felzenszwalb Segmentation can be seen in Figure 11.



Fig. 11. Felzenszwalb Segmentation (Image retrieved from Chicago Face Database. <https://www.chicagofaces.org/>)

After thorough research, the final decision was to base our solution on face landmark detection with an increased number of the latter. The final model detects 468 landmarks and creates a mesh for better guidance and estimation. This solution yielded crucial information for further use and was proven sufficient for recognition. An example of the final landmarks detection model can be seen in Figure 12.

B. Natural Language Processing

The first approach to text processing was subdividing the task into a pipeline, where the initial part will summarize the prompt extractively while maintaining the crucial information for detection, and the final part will evaluate the prompt by making a custom sentiment analysis of subparts of the text. Although the summarization task did not present any challenges, the importance of such was still a question, as the loss of information was inevitable, and the length of the raw input was undefined. The main challenge in the process was to analyze the adjectives in the text, define which parameter they were referring to, and resolve the sentiment

value they presented in the given context. This required a large custom dataset and fine-tuning the AI model, which was, at the time, chosen to be a base BERT (Bidirectional Encoder Representations from Transformers). The architecture of BERT [4] can be seen in Figure 13.

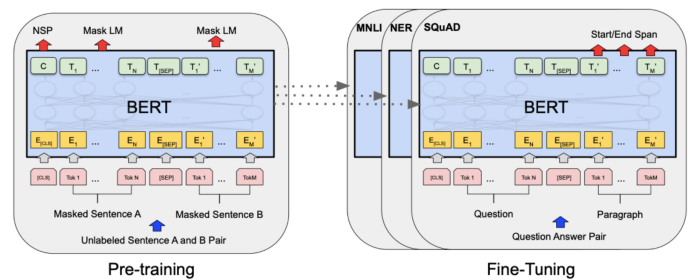


Fig. 13. BERT Model (Image retrieved from <https://arxiv.org/abs/1810.04805>)

The lack of the dataset and specificity of the task in the context of limited time left us with the other option of defining semantic value via inferring it from the word embedding model with pre-defined ranged sets of adjectives.

C. Search Algorithm

The search algorithm was changed several times during the two semesters. Initially, it was planned to search simply one metric at a time. In other words, we tried to first eliminate all people whose eyes do not correspond to the inserted data. Then, eliminate others whose nose sizes do not match. And do the same with all seven metrics. However, we quickly recognized that it is not the most optimal solution and could cause uncertainties.

The next approach was creating 7-dimensional feature arrays and calculating their Euclidean distances. Surprisingly, that simple and old algorithm demonstrated quite accurate results. However, we did not stop at that solution and decided to try and compare different approaches.

After the Euclidean distance, we implemented the cosine similarity algorithm. However, this approach was the weakest, demonstrating low accuracy during testing. The cosine

similarity was compared with linear regression and the K-nearest neighbor algorithm. Even though linear regression has shown fairly good results, the K-nearest neighbor was better at solving our problem. So, our final decision was to use KNN in order to search for the closest matches in the database.

D. Frontend

Initially, the visual layout was constructed for the frontend part. The skeleton of the website had no logic, but afterward, it was connected to the server. Subsequently, the automatic deployment integrated with GitHub was established for the purpose of faster debugging.

Some bugs were occurring during the development, but our group found them during the testing and they were quickly addressed. These bugs were primarily related to logic and layout.

During the last two semesters, we changed the layout of the website, modified its requirements, and fixed the bugs. Initially, the prototype of the website had a different color scheme and layout model. Additionally, the final version has improved design, and accessibility. Also, the website was minimized and optimized for the deployment. The design was modified during the semester from Figure 15 to Figure 14.

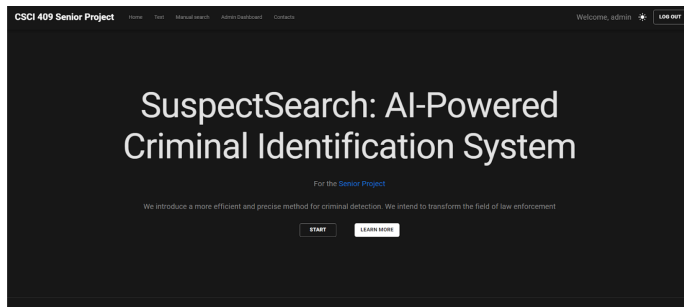


Fig. 14. Current Design

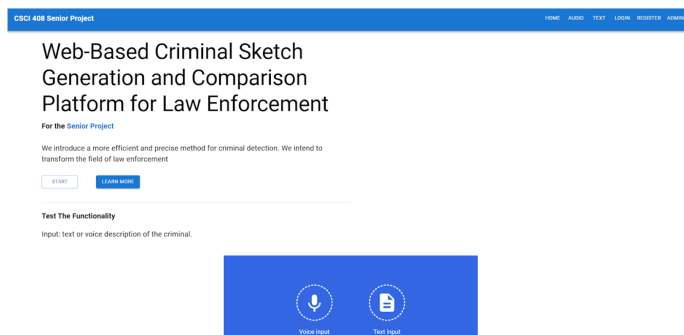


Fig. 15. Initial Design

During the semester the frontend part was tested, and the layout was improved. Additionally, a significant part of the time was spent on the architecture of the application, so that future minor fixes could be addressed quickly.

E. Database

The database of the project has undergone slight changes during the development of the product. The initial version of the database and tables can be seen in Figures 16 and 17. Initially, it was planned that we would have two separate tables for policeman and admin users that are connected to the Logs table. In addition, according to the initial solution to the problem, the Image Features included fields like beard, hair, mustache, and eyebrows distance.

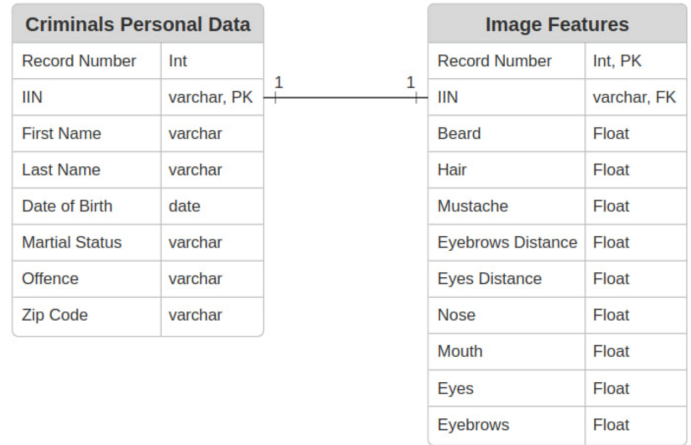


Fig. 16. Initial ERD: Criminals Data

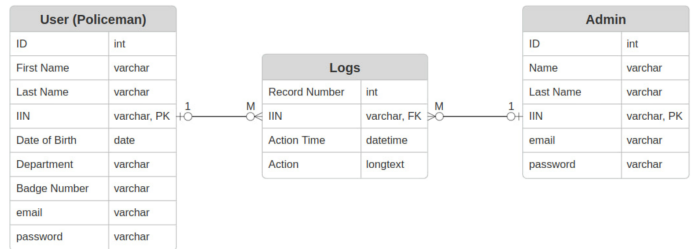


Fig. 17. Initial ERD: Users

At the end of the development of the product, the final database became as in Figures 7 and 8. First of all, the fields of the CriminalsData table have changed according to the metrics that are retrieved from the machine learning algorithm. Also, the MaxMin table was added. This table stores the maximum and minimum values of all 7 metrics. These maximum and minimum values are necessary to normalize the seven metrics that are used in the search algorithm. In addition, two separate tables (Policeman and Admin) were merged into a single CustomUser table. Roles and permission are defined by the 'role', 'is_staff', and 'is_superuser' fields of the CustomUser table. This table is connected to the Django AdminLog table by the 'user_iin' field.

During the development of the product we also faced a problem of a low response time from the server side. This problem arose because of the free TiDB Cloud Server that we were using to host our database at the beginning. TiDB is a free

AWS cloud server that was useful for deploying our testing database. Then, we bought a server from PythonAnywhere and deployed our backend and database there. The response time of the server from PythonAnywhere is much faster than the response time of the TiDB cloud server.

V. EVALUATION

To ascertain the effectiveness of our system post-implementation, we employed various evaluation methods. We segmented the assessment into two main components: Facial Metrics Detection and Searching Algorithm.

A. Manual Tests

To validate the accuracy of the Facial Metrics Detection and calculation algorithm, we conducted a series of manual tests. We selected diverse images featuring individuals with varying facial features and analyzed the extracted metrics. Subsequently, we cross-referenced these metrics with existing data to ensure their correctness. Below you can see the results of the testing. In the calculated result below you will see 7 metrics in each case. All these metrics are normalized where 0 means the smallest size and 1 means the biggest size.

In Figure 18, we can see a man characterized by a big nose and mouth size. We would depict his eyes as a bit bigger than the average size. His eyebrows appear to be of standard size, with the left eyebrow appearing slightly larger than the right. The results of our algorithm correspond to the given picture with very high accuracy. Specifically, the system identified his mouth size as 0.77, nose size as 0.71, nose length as 0.60, and the sizes of the right and left eyes as 0.61 and 0.5 respectively. Also, the algorithm accurately captures the asymmetry in eyebrow size. From Figure 18 it can be seen that the person has a bigger left eyebrow than the right one. The results of the algorithm perfectly align with the characteristics given in Figure 18 (the left brow size is 0.7 and the right brow size is 0.57)



Successfully Loaded

CLEAR PICTURE

The calculated result:

Nose_length: 0.6041322314049585
 Right brow size: 0.5661375661375664
 Left brow size: 0.7000000000000002
 Left eye size: 0.4855255647068417
 Right eye size: 0.6129811210302941
 Nose size: 0.718978964783739
 Lips size: 0.7691773727935829

Fig. 18. Test 1

The next case was used to specifically test the accuracy of identifying eye sizes and lips size. In Figure 19, we can see

a woman with asymmetric eyes and eyebrows. Her eyes and lips are huge compared to the people in Figures 20 and 21. The system accurately calculated these metrics. It can be seen from Figure 19 that the algorithm calculated the size of the right eye as 1 and the left eye size as 0.77. Keeping in mind that in our system results are normalized from 0 to 1 where 1 is the biggest possible size, she has huge eyes. Moreover, it can be observed that the size of her mouth was identified as 0.81 which closely corresponds to the image provided. The asymmetry of her eyes and eyebrows is also captured in the results, with recorded values of 0.80 for the right brow size and 0.7 for the left brow size.



Successfully Loaded

CLEAR PICTURE

The calculated result:

Nose_length: 0.6198347107438016
 Right brow size: 0.8042328042328046
 Left brow size: 0.7000000000000002
 Left eye size: 0.7671858435945527
 Right eye size: 1
 Nose size: 0.8026471283384542
 Lips size: 0.8109918684854228

Fig. 19. Test 2

For the next case, we selected an individual with small eyes and a long nose length. The results of the algorithm can be observed in Figure 20. According to the system, the size of the woman's eyes is around 0.33 which is relatively low compared to the previous two cases. Furthermore, other facial features of the woman are shown by the algorithm to be of average size, with values around 0.5.



Successfully Loaded

CLEAR PICTURE

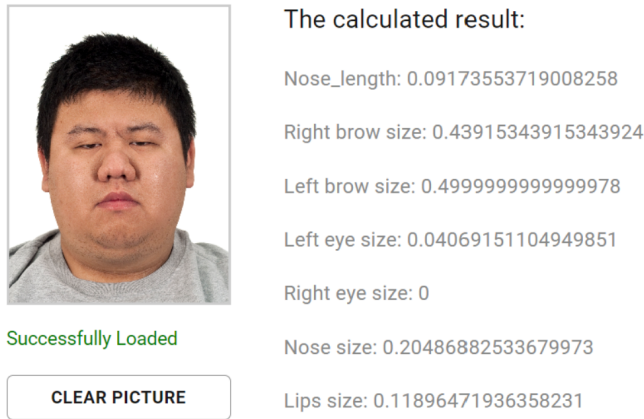
The calculated result:

Nose_length: 0.7024793388429751
 Right brow size: 0.507936507936508
 Left brow size: 0.6500000000000001
 Left eye size: 0.3394764739460165
 Right eye size: 0.3210888336016391
 Nose size: 0.5155282439139685
 Lips size: 0.4290861411666189

Fig. 20. Test 3

The last test case showed that there could be some anomalies in the results. For example, in Figure 21 we can see an

individual with a big nose size and a small. Extremely small eyes are captured well in the results with sizes of around 0. However, the nose size given in the results does not quite correspond to the picture that we see in Figure 21.



The calculated result:

Nose_length: 0.09173553719008258

Right brow size: 0.43915343915343924

Left brow size: 0.49999999999999978

Left eye size: 0.04069151104949851

Right eye size: 0

Nose size: 0.20486882533679973

Lips size: 0.11896471936358231

Fig. 21. Test 4

B. Questionnaire

The overall performance and the search results were tested initially by us. However, we concluded that our thoughts could be biased as we possess information about the dataset and feature metrics identification algorithm. Therefore, the system was given to anonymous users, and a questionnaire was implemented afterward. The most reasonable results of the questionnaire can be found in Appendix A (Figures 22, 23, 24).

According to the feedback received from the questionnaire, the overall performance of the system was great. It could identify people that approximately correspond with the given description. The majority of the respondents believe that this system could be used as an alternative or a complement to a current identikit system.

However, there is still room for improvement that can be seen from the users' feedback. Most of them agree that provided 7 metrics by the system (eyes, eyebrows, nose length, nose size, and mouth size) are not enough to identify a person. According to their thoughts, usually people memorize others according to their hair color, mustache, beard, scars, maybe some tattoos, etc. We agree in that regard, the system is not fully complete. Given the short period of time, we could manage to integrate only 7 metrics, but there is a possibility to include all the above-mentioned features as well.

In addition, some answers from the questionnaire indicate that the description of facial features depends on the perception of the witness. To avoid this problem we included a user guide with the scales of each metric that is observed in Figure 4. It does not scale precisely but still gives an approximate understanding of how the system perceives the sizes of different metrics.

Finally, there were complaints about the textual description functionality of the system. According to the respondents, sometimes the accuracy of the functionality could be critically low. Furthermore, some of them are convinced that we should clarify what kind of description should users insert. In other words, it would be a good idea to include a user guide or a prompt as an example.

Overall, the high accuracy of our own tests and positive user feedback can serve as proof that we solved the problem of improving the identikit system with minor complications. However, we should keep in mind that there is still a huge room for improvement. Once these improvements are done, the system can be used in the real world by law enforcement.

VI. CONCLUSION AND FUTURE WORK

The final system has proven to be an effective tool for improving the traditional approach of criminal investigation and criminal sketching. Although the metrics are purely empirical, they still give a gist of the overall information and the reference points for further investigation and strictly enhance the work of the intended users. The presented product will, however, need more extensions and improvements in order to increase the impact it potentially has.

Regarding the facial features, we limited the system to only seven metrics that were arbitrarily chosen as the crucial ones. The refinement by extension of the number of the aforementioned metrics will further enrich the description and precision of the system, allowing for more flexible and accurate results of the search. Possible additions might include facial hair evaluation, accessory detection (e.g. earrings, piercing), ink detection (tattoo), race/gender detection, and so on.

Considering improvements on the Natural Language Processing part, a whole new Large Language Model might be used with a sufficient dataset, computing resources, and budget for deployment. The crude semantic similarity of words could be replaced with a single Large trained model that will yield the results directly as a vector.

Overall, the system has numerous ways of enriching its use via additional functionality, such as speech recognition, which will allow the system to gather audial descriptions, process incoming voicemail, or work as an additional tool during the interrogation of victims/witnesses.

VII. APPENDIX A

Questionnaire results:

- 1) Did the system successfully identify individuals closely resembling the description provided (Figure 22)?
- 2) Do you believe this system could serve as a viable alternative to or complement the identikit system currently utilized by law enforcement agencies (Figure 23)?
- 3) In order to enhance the efficiency and effectiveness of the system, are there any specific improvements or additions you would recommend to optimize its functionality and user experience (Figure 24)?

First, I imagined a person with certain types of facial features. Then it was really easy to input those features using the manual search, where I could adjust facial characteristics using the slider. The result, which depicted the list of people according to my criteria was surprisingly accurate. I could see in them the person, whom I'd imagined. In contrast, the textual description search posed challenges in articulating desired features effectively. Overall, while the manual search function excelled, the textual description search fell short in precision.

By selecting the necessary features, I obtained results that closely resembled the imaginary suspect I had envisioned. I can say without a doubt, the system demonstrated its capability by accurately identifying individuals who closely resembled the description provided. This showcases its effectiveness in pinpointing suspects based on specific traits, contributing significantly to law enforcement efforts

Yes, I had no flaws during the usage of the website

Yes, it provided close results, however in some cases it did not identify a person

Mostly yes, i did clearly see the identification results when I changed size of eyes, however, other face marks were not clearly visible for me. Maybe perception of face firstly pays attention to the eyes instead of mouth or nose

Fig. 22. Results 1

I believe this system has the potential to be a valuable addition to the toolkit of law enforcement agencies. Its functionality, which allows to retrieve people, based on witness descriptions proves that it could serve as a promising complement to the traditional identikit method.

I am confident that this system could indeed serve as a valuable complement to the identikit method employed by law enforcement agencies. Its ability to retrieve individuals based on witness descriptions demonstrates promise in augmenting traditional identikit practices.

In my opinion yes, the ML algorithms work fine, and the website implements the idea of the identikit system

I think it has a big potential, because of automatization that might save the time of policemen

I dont think current system can replace identikit system as it just simply lacks parameters for search, even though they seem to work fine.

Yes, this is a viable option, although the system needs to be developed a lot, the idea and the implementation of the project look promising

Fig. 23. Results 2

VIII. ACKNOWLEDGMENT

Special thanks to the University of Chicago for providing access to their Chicago Face Database. The dataset of people retrieved from this database played a crucial role in implementing this project.

REFERENCES

- [1] Facial Keypoints Detection — kaggle.com. <https://www.kaggle.com/competitions/facial-keypoints-detection/overview>. [Accessed 19-04-2024].
- [2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.
- [3] D. S. Chukmaitov. Recidivism of crimes as the most important criterion for the effectiveness of criminal law influence — repository.kazguu.kz. <http://repository.kazguu.kz/handle/123456789/324>, 2014. [Accessed 19-04-2024].
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [5] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004.
- [6] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.
- [7] S Mittal, T Singh, YK Arya, and MK Tiwari. Effect of emotional arousal on eyewitness memory. *Indian Journal of Social Science Researchers*, 10(1):24–30, 2013.
- [8] Chai M Tyng, Hafeez U Amin, Mohamad NM Saad, and Aamir S Malik. The influences of emotion on learning and memory. *Frontiers in psychology*, 8:235933, 2017.

One area for improvement is to expand the search parameters to include additional identifying features beyond eyes, brows, mouth, and nose. Incorporating options for scars, tattoos, beard, or other distinctive marks. I believe that it would enhance the system's accuracy in identifying individuals.

I'd improve the textual search functionality of the system. My experience with this feature was marked by confusion, primarily due to the lack of clear guides or hints provided. Implementing intuitive prompts or assistance within the textual search interface would tackle user confusion and enhance the search process. By offering clearer guidance and support, users can navigate the textual search function more effectively, ultimately improving the overall usability and satisfaction of the system.

Probably optimize the algorithms and use more data.

It might be efficient to increase the number of suspects in a database

The "drawing" that changes with the change of parameters may be could be more realistic in terms of change of nose parameters (i don't think that's how nose is sized). + when I see person I remember it not by size of mouth or nose (of course if they are not ginormous) but if they have scars, beard, mustache, may be even hair color\style or something. but not definitely size of nose and mouth

Fig. 24. Results 3

- [9] V. V. Vasylichkov. The methodology of expert research of victims, suspects, other persons, and corpses for personality identification — online.zakon.kz. https://online.zakon.kz/Document/?doc_id=34266912pos=4;-88pos=4;-88, 2016. [Accessed 19-04-2024].