WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Sketch Based Face Matching System

Saritha D[1],Prajwal T S[2],Dileep M D[3],Srinidhi B S[4],Varun Kumar B V[5]

Assistant Proffesor ,Computer Science and Engineering, Bahubali College of Engineering Shravanabelagola
Student,Computer Science and Engineering,Bahubali College of engineering,
Student,Computer Science and Engineering,Bahubali College of engineering,
Student,Computer Science and Engineering,Bahubali College of engineering,
Student,Computer Science and Engineering,Bahubali College of engineering.

ABSTRACT:

The system accepts facial sketches as input and compares them with a database of suspect photographs using advanced facial embedding techniques. It employs OpenCV's Haar cascade classifier to detect and crop facial regions from both sketches and real images, ensuring that only relevant facial features are processed. DeepFace's Facenet model is utilized to extract high-dimensional facial embeddings, capturing unique features of each face. These embeddings are then compared using cosine similarity to identify the most similar face from the database.

Introduction:

In forensic investigations, accurate and timely suspect identification is critical. One of the traditional techniques employed by law enforcement agencies is sketch-based suspect identification, where eyewitnesses provide descriptions that artists translate into facial sketches. However, manually comparing these sketches with large databases of suspect photographs is time-consuming, error-prone, and highly dependent on human expertise.

With the advancements in deep learning and computer vision, it is now possible to automate and enhance this process. This project, "Sketch-Based Face Matching for Forensic Investigation," aims to bridge the gap between hand-drawn sketches and real suspect images using a machine learning-based face recognition system. By employing the Facenet model from the DeepFace library, the system extracts facial features (embeddings) from both sketches and stored images, enabling efficient and accurate similarity comparisons.

The application integrates OpenCV for face detection, DeepFace for feature extraction, and cosine similarity for matching, all wrapped within a Python Flask web framework for ease of use. Authorized users such as forensic investigators can upload sketches via the web interface and retrieve the most similar suspect image from the database based on facial embeddings.

This system not only increases the speed and reliability of sketch-based identification but also provides an accessible, real-time tool for forensic professionals, thereby modernizing and supporting law enforcement operations in a meaningful way.

Methodology:

The development of the *Sketch-Based Face Matching for Forensic Investigation* system follows a systematic and modular approach that combines deep learning, image processing, and web technologies to enable accurate suspect identification based on sketches. The methodology comprises the following phases:

1. Data Collection and Preprocessing

- The system maintains a structured image database containing real face photographs of suspects.
- Facial sketches are provided as input through the web interface.
- *OpenCV's Haar Cascade Classifier* is used to detect and crop facial regions in both sketches and real images.

- All face images are resized and standardized to ensure uniform processing and reduce variation during feature extraction.

2. Feature Extraction Using Deep Learning

- *DeepFace* library with the *Facenet* model is employed to extract high-dimensional facial embeddings from both real images and sketches.
- These embeddings represent distinctive facial features in vector format, enabling precise similarity measurement.
- Embeddings for stored face images are precomputed and stored in memory for faster comparison during sketch matching.

3. Sketch-to-Image Matching Using Cosine Similarity

- Upon sketch upload, the system performs facial detection and extracts the sketch's embedding using the same model.
- It calculates the *cosine similarity* between the uploaded sketch's embedding and each stored image's embedding.
- The image with the *highest similarity score (lowest distance)* is selected as the best match.
- A similarity score is displayed along with the matching image to interpret the result (e.g., perfect match, very similar, or different).

4. Web-Based Implementation with Flask

- The application is implemented using *Flask*, a lightweight Python web framework.
- Forensic authorities can securely log in, upload sketches, and view matching results through a clean, user-friendly interface.
- The system provides visual feedback, displaying both the uploaded sketch and the matched real face side-by-side with similarity metrics.

5. Result Interpretation and Feedback

- The system offers guidelines to interpret similarity scores:
    - *0.0 → Perfect Match*
    - *0.2 – 0.4 → Very Similar*
    - *0.5 – 0.7 → Moderately Similar*
    - *> 0.7 → Different Faces*
- If no face is detected in the sketch, the system returns an appropriate error message to ensure meaningful feedback.

Objective:

1. *Develop an AI-Based Sketch-to-Image Matching System*
2. *Implement Deep Learning for Facial Feature Extraction*
3. *Enhance Forensic Investigations with Automated Face Detection*
4. *Provide a User-Friendly Web-Based Interface*

Results

Sketch-based face matching systems utilize advanced algorithms to compare facial sketches with existing databases, enabling identification or verification of individuals. These systems have applications in forensic investigations, security, and identity verification. By extracting facial features from sketches and matching them against databases, these systems can provide accurate results despite variations in sketch quality and style. Effective sketch-based face matching systems require robust algorithms and large, diverse databases to ensure accurate matches. Ongoing research focuses on improving algorithm accuracy and handling variability in sketches, enhancing the potential for real-world applications.

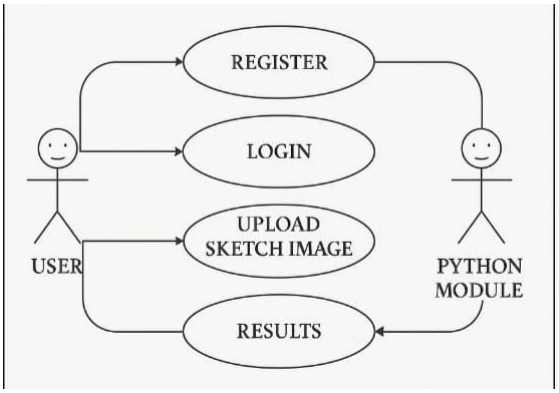

Fig 1 Block Diagram

Conclusion

The Sketch-Based Face Matching for Forensic Investigation project presents an innovative solution to a significant challenge in forensic investigations: matching facial sketches with real images. By integrating deep learning techniques like DeepFace and Facenet with cosine similarity for efficient facial recognition, the system successfully bridges the gap between abstract sketches and real-life images. The use of cosine similarity ensures an accurate and objective comparison of facial embeddings, making the system capable of identifying suspects even with imperfect or incomplete sketches.

REFERENCES:

List all the material used from various sources for making this project proposal

Research Papers:

1. *Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L.* (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1701-1708.
2. *Schroff, F., Kalenichenko, D., & Philbin, J.* (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815-823.
3. *Parkhi, O. M., Vedaldi, A., & Zisserman, A.* (2015). Deep Face Recognition. *British Machine Vision Conference (BMVC)*, 41.1-41.12.
4. *Viola, P., & Jones, M.* (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, 511-518.
5. *Jain, A. K., Ross, A., & Prabhakar, S.* (2004). An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 4-20.
6. *Kumar, R., Patil, A., & Chatterjee, S.* (2021). A Deep Learning-Based Face Recognition System Using OpenCV and TensorFlow. *International Journal of Computer Applications*, 183(32), 20-27.
7. *DeepFace Library* - Open Source Facial Recognition Framework. Available at: https://github.com/serengil/deepface
8. *Flask Framework* - Official Documentation. Available at: https://flask.palletsprojects.com/