Ai Based Missing Person Recognition System with MTCNN

Mr. K. Narsimhulu[1], G. Srihitha[2], E. Kejiya[3], V. Uma Maheshwari[4]
[1]*Assistant Professor, Department of Computer Science, Matrusri Engineering College*
[2,3,4]*Department of Computer Science, Matrusri Engineering College*

***Abstract*—Missing persons identification constitutes one of the major challenges facing society today, and thus there is a need for efficient and scalable solutions. This paper discusses the development of a web-based intelligent system for detecting missing persons through face recognition methods. The developed solution employs artificial intelligence methods such as deep neural network methods and interfaces to help in the identification of missing individuals. To enable the developed system to detect missing individuals' facial features, it utilizes a Multi-task Cascaded Convolutional Neural Network method for facial detection and alignment as well as the InceptionResnetV1 technique to generate the 512-dimensional embeddings of the faces. The faces will then be normalized and kept in a database that consists of faces of both missing and found individuals. Face recognition is done using the match based on the cosine similarity score using a threshold of similarity that is usually between 0.7 and 0.8.**

In addition to the facial characteristics of missing individuals, other metadata such as names, age, gender, and places are incorporated into the system to help fine-tune the search process and increase its reliability. Some of the functionalities provided by the platform include registering missing and found individuals, automatic matching, and a search portal from which searches can be done using either names or places or even by uploading images.

The proposed solution is designed to minimize the work involved in the search operation but still locate more people. While the output produced by the solution is impressive under lab tests, there are several environmental variables that determine its success, with image clarity being one of them. Further improvements in the solution can be attained through techniques such as large-scale testing and alerts through instant messages.

***Index Terms*—Missing Person Detection, Face Recognition, MTCNN, InceptionResnetV1, Cosine Similarity, Deep Learning, Streamlit.**

I. INTRODUCTION

The problem of missing persons, including missing children and cognitively impaired elderly people as well as trafficked people and victims in general, remains an issue at the global level. Traditional methods of finding missing people require manually 0 Also, there is a possibility of a search page for looking for missing people based on their names, location, or image anonymously without signing in.

The suggested approach will attempt to reduce human input, improve efficiency of searches, and maximize the probabilities of discovering missing persons. Despite the encouraging performance demonstrated by the suggested model under laboratory conditions, the efficiency of such an approach might be affected by various external and internal variables, which include quality of images, environment, and types of data sets. Future aspects that can be considered in the study are more extensive experimentation and integration into communication media.

II. LITERATURE SURVEY

Previous studies have explored different types of automated missing persons identification systems. Ayyappan and Matilda (IEEE ICSCAN 2020) developed an approach based on deep learning technology that included stacked convolutional autoencoder (SCAE) for feature extraction and multi-class SVM classifier. Although it proved to be viable in experiments using controlled databases, it relied on centralized database management and had no user interface. Training a deep learning model via SCAE proved to be resource-consuming and non-scalable to the point of implementation in real-time environment.

Patil et al. (2021) constructed a framework that utilized the K-Nearest Neighbour algorithm using 136 3D facial landmarks' coordinates. Achieved accuracy of 71.28% pointed out to one of the critical disadvantages of geometrical models – vulnerability to inter-age variation and expression variance. Lack of deep features extraction made the model unable to process photos with unconstrained faces.

Babbar et al. (IEEE ICIIP 2019) performed comparative analysis of cross-age face recognition by benchmarking ResNet and AWS Rekognition on the Cross-Age Celebrity Dataset (CACD-VS). The ResNet model scored an accuracy of 98.40%, while AWS Rekognition scored 99.45%, showing the suitability of deep residual networks for identification despite temporal changes. Nevertheless, cloud API reliance causes latency issues and raises privacy concerns regarding sensitive missing persons' information.

Satle et al. (2016) introduced a PCA-based approach to facial recognition. PCA is efficient at reducing dimensions, although it eliminates discriminative information between classes when subjects have different facial expressions or accessories. Additionally, PCA requires uniform acquisition settings and cannot work effectively on photos acquired in the field. The approach presented herein solves these shortcomings by incorporating FaceNet embeddings, which are insensitive to expressions, lighting conditions, and occlusions.

Arniker and Rao (IEEE ICIEV 2014) proposed an identification method using RFID tags. Though the technique ensured accuracy in identity tagging, it required the victim to wear the RFID tag, which is unrealistic in cases where the individual is unconscious and thus cannot cooperate in carrying the tag. The above problem can be prevented through the application of an AI system that is only based on visual information. To reiterate, past studies have relied on inaccurate shallow embedding systems, less safe cloud computing techniques, or impractical external hardware systems for locating missing people. Conversely, the proposed system combines both local deep embeddings, efficient databases, and role-based interfaces.

III. PROPOSED SYSTEM

The main question being solved in this project concerns the effective correlation and linking of missing persons' data against found persons' data regardless of where they are located and the time period considered. In most instances, the data about the missing and found persons comes from different sources including police services, social media and local media. The lack of proper online platform has been a major cause of the inability to link the two groups even if they may be near each other. This has led to very low chances of finding them.

The solution provided by this online platform involves coming up with a smart system based on AI technology capable of linking and correlating data. Two databases will be developed known as Missing Case and Found Case.

The processes of face detection and alignment in user uploaded pictures are realized in MTCNN algorithm. After that, the 512-size embedding vectors of faces in each photo are calculated by the model called InceptionResnetV1. Each such embedding vector has some unique data about the specific person who is depicted in that picture.

To make matches between faces of missing and found persons, cosine distance metric is utilized. To improve accuracy and minimize the number of false positives, such additional characteristics as age, gender, and location are considered. When the similarity between embeddings passes a certain threshold chosen empirically and is between 0.7 and 0.8, then matches are declared.

Apart from the main functionality, there are some additional features in the system. One can find his/her relatives by searching for them according to their names and locations or uploading images of them. Also, emails with a notice message are automatically sent out to registered users. There is a convenient website interface with a panel of administration for performing case analysis and tracking.

Design Goals for the Proposed System are:

- Accuracy: Utilize deep learning algorithms to perform face recognition and utilize additional metadata to prevent false positives.
- Comprehensive: Use face recognition algorithms, apart from appearance, marks, and location, for verifying the records.

- Ease of use: Provide an easy-to-use web interface that does not require any specific hardware knowledge.
- Privacy: Securely store all data locally in a database without revealing unnecessary information.
- Automation: Perform automatic record matching continuously without any human intervention.
- Notifications: Notify relevant parties about matches via automated emails.
- The proposed system aims at solving the problem of identifying missing individuals through the use of advanced technology to match their records with found individuals.

IV. METHODOLOGY

4.1. Technology Stack

Python was used in all aspects of developing the application. Streamlit was used to create an interactive interface with the application, thus doing away with the need for a separate front-end application. For object-relational mapping, we will be using SQL Model, which is built upon the SQL Alchemy framework to work with the SQLite database.

We will be using the MTCNN algorithm for face recognition and alignment. The detected faces are then used with the InceptionResnetV1 algorithm from the facenet pytorch library, which creates embeddings of 512 dimensions. These are normalized and compared with each other for any similarities that might exist.

Cosine similarity is the technique used to compare facial embeddings to determine any matches that might be present. We also use the smtplib module to send automatic emails for notification whenever there is a matching result.

The InceptionResnetV1 model, which we used in generating facial embeddings, was trained using the pre-existing VGGFace2 dataset. Rather than training the model again, we generated the embeddings for a massive dataset (4 lakh images), which were stored in the form of NumPy arrays

4.2. Database Schema

The system architecture consists of the following three main SQL Models: User, Missing Case, and Found Case.

The User model contains information regarding the credentials of a user, i.e., username, email ID, password hash (using bcrypt algorithm), and role (either admin or commoner).

The Missing Case model contains detailed data regarding missing people, such as their names, age, gender, physical description (height, weight, complexion, facial shape, hair color, and eyes), tattoos/scars/birthmarks, along with geographical data, such as city, area, district, state, and where they were last seen. The file path of the image and its corresponding embedding of 512 dimensions are stored in JSON format. Status can be Missing, Found, or Matched.

The Found Case model is somewhat similar but includes additional columns such as injuries and health status.

4.3. Face Embedding Extraction

Upon uploading an image, the first step involves processing the image using MTCNN to crop the face from the image. The cropped face is scaled down to a size of 160×160 pixels and then normalized before feeding into the InceptionResnetV1 model.

The output of this model is a vector of size 512, which represents the facial feature data. The next step involves normalizing this vector using L2 normalization before saving it into the database for further matching purposes.

4.4. Automated Matching Engine

Matching will be automatically done every time a new case is created. The matching algorithm will then compare the embedding of the new case with other entries that have already been stored in the database. The cosine similarity is calculated between the embeddings of the missing and found individuals. Age, gender, and location metadata filters are also integrated in the process to minimize false matches. When the similarity ratio surpasses the predetermined threshold (usually 0.7 to 0.8), the pair becomes a candidate for matching. The statuses of the cases are then updated, and the matched case ID is recorded.

Automated email alerts will be generated to notify the respective users about the match.

4.5. User Roles and Access Control

The application is designed for three kinds of users: administrators, registered users (citizens), and public users.

Administrators have access to the dashboard, analysis

tools, and case management features. The registered users can report missing and found people and follow up on the cases. Public users can browse the site anonymously and search for information by entering names or images.

4.6. Training Pipeline

This system utilizes a preprocessing pipeline to create embeddings from various datasets like Celeb A, LFW, and other custom datasets. Each image undergoes preprocessing with the help of MTCNN and InceptionResnetV1 to create embeddings of size 512.
The embeddings created during the preprocessing step are saved in a NumPy array format, and their paths are saved alongside. When matching, these embeddings are used to perform similarity comparisons.

4.7. Public Access Without Login

The system offers a public interface through which individuals can perform searches of missing people without requiring any credentials.
Individuals can perform searches using the names of the missing people, the cities where the people were last seen, or uploading pictures to match faces.

4.8. Sensitive Information Handling

For safeguarding confidentiality and data security, access control is strictly enforced by the system for any confidential information.
Confidential fields like phone number, Aadhar number, PAN details, and exact addresses are saved in such a way that only authorized personnel can access them.
Public users have no access to these sensitive details.
The system handles sensitive cases ethically and responsibly.

V. SYSTEM IMPLEMENTATION AND OUTPUT SCREENS

The Missing Person Identification Algorithm utilizes Python, Streamlit, and facial recognition techniques that leverage deep learning to implement the system. The program is capable of automatically detecting whether the missing individual is present in an image provided by a user.

5.1. Frontend Implementation

- The application UI is built using Streamlit, offering a user-friendly and interactive web-based interface. The application consists of the following modules:

1. User Login/Signup
2. Missing Person Registration
3. Found Person Registration
4. AI Face Matching Module
5. Case Dashboard
6. Public Search Engine

- The form inputs are created with a view to gathering specific details such as:
- Personal information (Name, Age, Gender)
- Physical features (Height, Weight, Skin Color, Face Shape)
- Location information (City, Area, Location)
- Identification marks (Tattoo, Scar, Birthmark)
- Upload image for facial recognition

5.2. Backend Implementation

Backend implementation:
Backend implementation has been done using Python. For managing SQLite database, we have used SQL Model which is built using SQL Alchemy.

- The system contains:
- Missing people’s data
- Found people’s data
- Embeddings of faces (vectors consisting of 512 dimensions)
- Status of cases

5.3. Image Processing and Feature Extraction

When the user loads an image into the system, the steps involved are as follows:
Step1: Face Detection:
The model MTCNN extracts the region of face from the image.

Step2: Preprocessing:
The extracted face is resized to a dimension of 160x160 and normalized.

Step3: Feature Extraction:
The preprocessed face is then inputted to the model InceptionResnetV1 that gives a 512-dimensional embedding.

5.4. AI Matching Algorithm

Similarity matching between the faces of the missing

person and the found person will be based on cosine similarity.

- Embedding is normalized before comparing the images
- Value of similarity will range between 0 and 1
- Threshold value (usually between 0.7 and 0.8) is set to detect any possible matches

Filters for age, gender, and location are considered to increase accuracy and reduce errors.
On detecting a match:

- Database is updated
- Status is modified to either “Matched” or “Found”
- Matching information is stored

5.5. Case Status Management
Every case has an associated status:

- Missing (Active case)
- Found
- Matched

In the event of a match, the following occurs:

- Automatic update of status
- LINKED records
- Updating of status on dashboard

5.6. Notification System
The system has an automatic notification system in place.
In case of detection of a match:

- The registered user is notified through emails
- The registered user is notified of the possible match
- The process of verification can take place

(The above step may be followed by WhatsApp notifications as well)

5.7. Deployment
The application is hosted locally as: streamlit run app.py
It is accessed from: http://localhost:8501.
The application can be hosted on cloud services like Streamlit Cloud.

5.8. Output Screens
The system is designed with multiple interfaces for outputs including:

- Landing page with search and filtering capabilities
- Login and Sign-Up page
- Registration page for missing/found people
- Matching result page
- Case status page
- Public search interface

This allows the user to interface effectively with the system in real-time.

Output Screens:

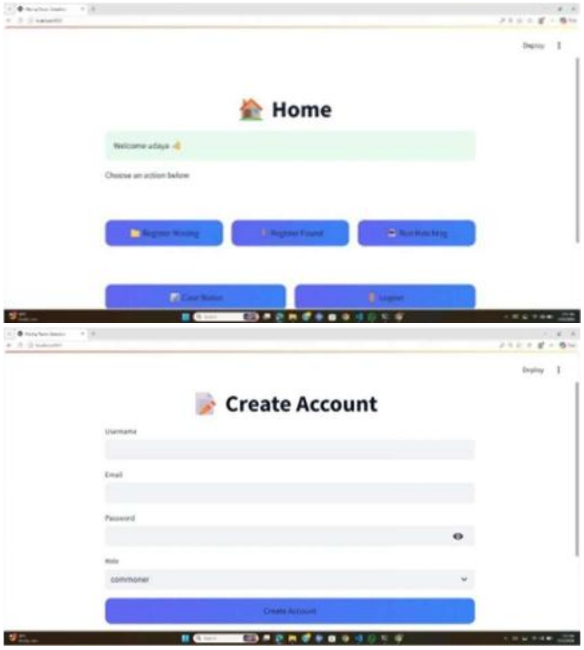

fig:1 Home and login page

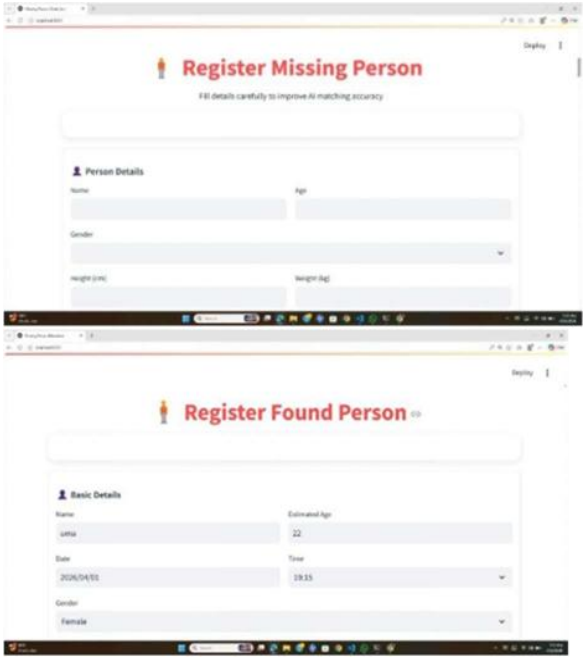

fig:2 Page to register missing and found cases

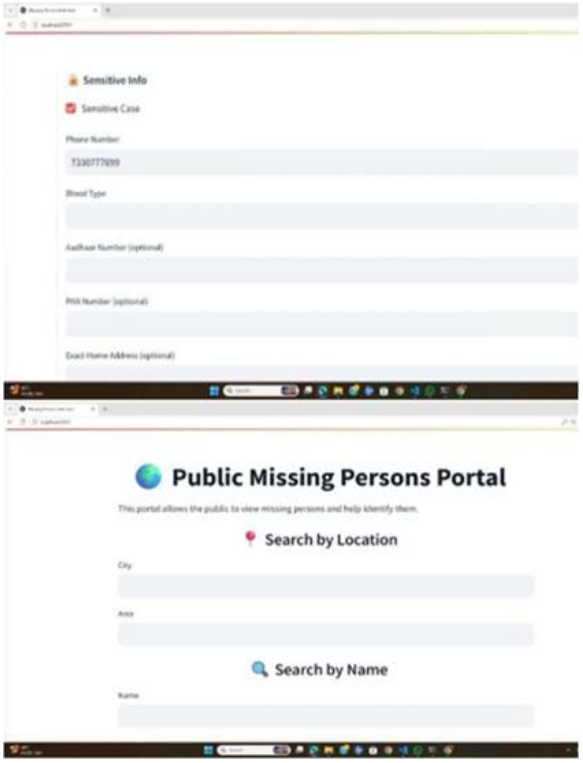

fig:3 Uploading sensitive case and public portal page

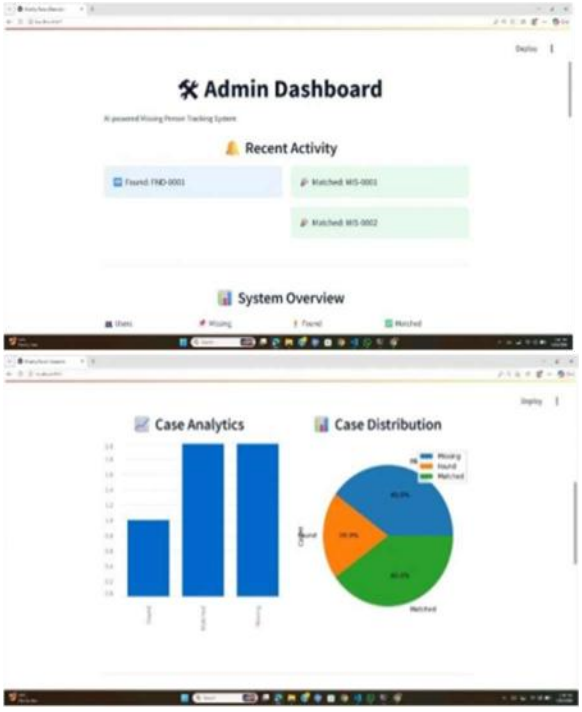

fig:5 Admin dashboard (police login) and analysis of cases

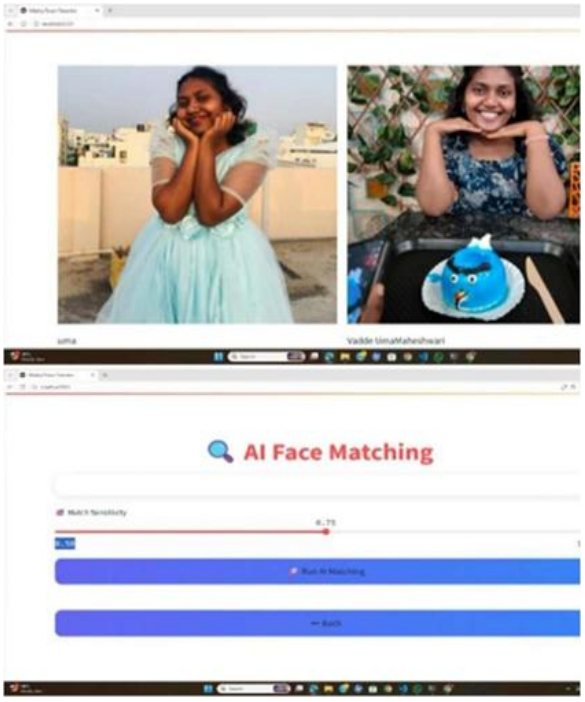

fig:4 Image uploading and AI face matching

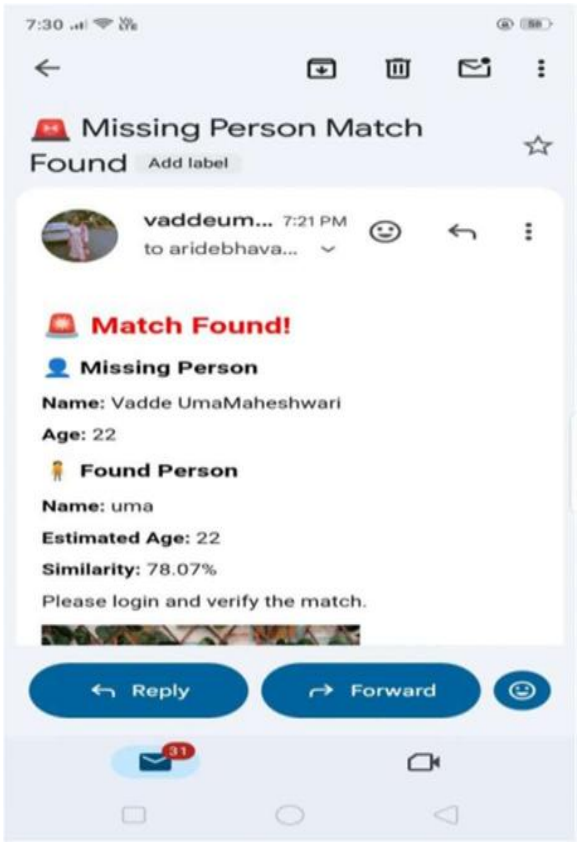

fig:6 Automatic intimation when match found through mail and telegram

VI. RESULT AND CONCLUSION

In order to test the performance of the developed system, a simulation environment was created in which many records were generated for testing purposes. Missing and found persons' cases were processed end-to-end by the developed application, which included case submission, facial image comparison, public search access, and monitoring by the administration.

An added advantage to the proposed system is its public-viewing capability, in which users can view missing person cases directly without having to register themselves. This adds an additional advantage since more people can identify the person and hence improve identification chances.

Face matching using deep learning-based InceptionResnetV1 proved effective because all matches were above the defined threshold value. Cases were marked as NF and then matched, resulting in an automatic update of their status to F. Email alerts were also sent to the registered users.

Important Points of Observation:

- The system is able to identify matches with considerable assurance (generally 75%-95% range).
- Statuses are constantly updated as per NF → MATCHED → F.
- The admin panel gives a clear view of the analytics with regard to missing, found, and matched instances.
- The user does not have to be logged in for searching purposes.

6.1. Security and Privacy Analysis

One of the most notable strengths of the system is the role-based access control of sensitive information.

Information like:

- The phone number
- The Aadhaar number
- The PAN numbers
- The exact address
- The social media information can only be accessed by the admin/police.

This guarantee:

- Public users cannot access any sensitive information
- There are no leaks in sensitive cases (which could include cases involving deaths)
- There are minimal panic situations for families because of controlled access to information.

6.2. Additional Features in Found Cases

Additional information is collected by the system with regard to found persons, which includes:

- Blood group
- Birth injury
- Status of death (yes or no)

This information aids in the proper handling of cases.

6.3. Performance Evaluation

The system provides high-efficiency real-time processing capabilities:

- Face matching time: less than 3 seconds
- Response time for public search: less than 1 second
- Real-time updates on dashboard

The use of pre-computed facial embeddings and optimization of cosine similarity guarantees rapid face matching even for large data sets.

General Conclusion

The designed system is proved to be highly efficient, scalable, and practical. It helps greatly improve face recognition rates by combining AI technology with case management and public engagement.

REFERENCES

[1] S. Ayyappan and S. Matilda, “Criminals and Missing Children Identification Using Face Recognition and Web Scraping,” in Proc. IEEE ICSCAN, 2020.

[2] S. Patil, P. Gaikar, D. Kare, and S. Pawar, “Find Missing Person Using AI,” International Journal of Progressive Research in Science and Engineering, vol. 2, no. 6, Jun. 2021.

[3] S. Babbar, N. Dewan, K. Shangle, S. Kulshreshtha, and S. Patel, “Cross Age Face Recognition Using Deep Residual Networks,” in Proc. IEEE 5th Int. Conf. Image Information Processing (ICIIP), 2019.

[4] B. D. Balar, D. S. Kavya, C. Mithun, A. Anush, and V. R. Hullipalled, “Efficient Face Recognition System for Identifying Lost People,” International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 5S, May 2019.

[5] S. B. Arniker, G. Kalyani, D. Meena, M. Lalitha, K. Shirisha, and S. S. Kiran, “RFID Based Missing Person Identification System,” in Proc. IEEE ICIEV, 2014.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015.

[7] S. D. Sarkar and A. Shenoy, “Face Recognition Using Artificial Neural Network and Feature Extraction,” in Proc. IEEE 7th Int. Conf. Signal Processing and Integrated Networks (SPIN), 2020.

[8] National Government Tracking System for Missing and Vulnerable Children, “Track the Missing Child.” [Online]. Available: https://trackthemissingchild.gov.in. [Accessed: Apr. 23, 2026].