**Missing Child Identification System Using Deep Learning and KNN**

A PROJECT REPORT

Submitted

*in the partial fulfillment of the requirements for the award of the degree of*

## BACHELOR OF TECHNOLOGY

in

## COMPUTER SCIENCE AND ENGINEERING

by

## J. PRAVALIKA (19B81A0531)

## G. PRANEETH (19B81A0530)

## I. SHIVANI REDDY (19B81A0543)

Under the guidance of

## Dr. RAM BABU

Associate Professor, CSE Department



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

**(*An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana- 501 510

## October 2022

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey our profound sense of gratitude to our principal **Dr.K. Ramamohan Reddy**, Vice-Principal **Prof. L. C. Siva Reddy**, **Dr. A. Vani Vathsala**, Head of CSE Department, CVR College Of Engineering, for having been kind enough to arrange necessary facilities for executing the project in the college.

We deem it a pleasure to acknowledge our sense of gratitude to our project guide **Dr. Ram Babu** under whom we have carried out the project work. His incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish a deep sense of gratitude and heartful thanks to the management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

## J. PRAVALIKA (19B81A0531)

## G. PRANEETH (19B81A10530)

## I. SHIVANI REDDY (19B81A0543)

**DECLARATION**

We the undersigned solemnly declare that the project report titled ‘**Missing Child Identification System Using Deep Learning and KNN**’ is based on our own work carried out during the course of our study under the supervision of Dr. Ram Babu, CSE Dept., CVR College of Engineering.

We assert the statements made and conclusions are drawn are an outcome of our research work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or in the any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.

# CERTIFICATE

This is to certify that the project titled **‘Missing Child Identification System Using Deep Learning and KNN’** is being submitted by **J. PRAVALIKA (19B81A0531), G.PRANEETH (19B81A0530)**, **I. SHIVANI REDDY (19B81A0543)** in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering to the CVR College of Engineering, is a record bonafide work carried out by them under my guidance and supervision during the year 2020-2021. The results embodied in this project work have not been submitted to any other University or Institution for the award of any degree or diploma.

Signature of the project guide, Signature of the HOD,

## Dr. Ram Babu Dr. A. Vani Vathsala

Associate Professor Head of Department (CSE)

CSE Department CVR College of Engineering

# ABSTRACT

A novel use of deep learning methodology is proposed in this project for identifying the missing child that are reported from the photos of various of children available, by using face recognition. The public can click the photographs of suspicious child and upload them into a portal with landmarks as well as remarks. This photo will be compared automatically with the photos of the registered missing child cases present in the repository. The input child image is classified with photo having best match and that photo will be selected from the database consisting of missing children. A deep learning model is trained in order to identify the missing child correctly from the database provided with missing child cases, with the help of image uploaded by the public. The Convolutional Neural Network (CNN) is a highly effective technique in deep learning for image-based applications which is adopted in this project for face recognition. Using these images, face descriptors are extracted with the help of a pre-trained CNN model having VGG-Face deep architecture. Our algorithm uses convolution network which has as a high-level feature extractor when compared to normal deep learning applications. Child recognition is done based on the trained KNN classifier. Choosing the best performing CNN model for face recognition, VGG-Face and proper training of it results in a deep learning model invariant to noise, illumination, contrast, occlusion, image pose and age of the child and it outperforms earlier methods in face recognition based missing child identification.

# TABLE OF CONTENTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Table of Contents** | | | |
|  |  | | | Page No. |
|  |  | List of Figures | | vii |
|  |  | List of Abbreviations | | ix |
| 1 |  |  | **Introduction** | 1 |
|  | 1.1 |  | Introduction | 1 |
|  | 1.2 |  | Motivation | 3 |
|  | 1.3 |  | Problem statement | 3 |
|  | 1.4 |  | Project Objectives | 4 |
|  | 1.5 |  | Project Report Organization | 5 |
| 2 |  |  | **Literature Survey** | 6 |
|  | 2.1 |  | Literature Survey | 6 |
| 3 |  |  | **Software & Hardware specifications** | 9 |
|  | 3.1 |  | Functional/Non-Functional Requirements | 9 |
|  |  | 3.1.1 | Functional Requirements | 9 |
|  |  | 3.1.2 | Non-Functional Requirements | 9 |
|  | 3.2 |  | Software requirements | 10 |
|  | 3.3 |  | Hardware requirements | 10 |
| 4 |  |  | **Design** | 11 |
|  | 4.1 |  | Use case Diagram | 11 |
|  | 4.2 |  | Activity Diagram | 12 |
|  | 4.3 |  | ER Diagram | 16 |
| 5 |  |  | **Implementation & Testing** | 19 |
|  | 5.1 |  | Implementation | 19 |
|  | 5.2 | Testing | | 22 |
| 6 |  | **Conclusion & Future Enhancement** | | 27 |
|  | 6.1 | Conclusion | | 27 |
|  | 6.1 | Future Enhancement | | 27 |
| 7 |  | **References** | | 28 |

**List of Figures**

* 1. Use Case Diagram 7
  2. Activity Diagram for Issuers and admin 12
     1. Activity Diagram for holder 15
     2. Activity Diagram for holder 18

4.4 ER Diagram 18

4.5 Deployment Diagram 19

4.6 Component Diagram 19

* + 1. View Page 20
    2. Issue Page 20
    3. Verification Page 21
    4. Component for viewing public key 21
    5. Meta mask wallet Interface 22
    6. Remix Interface for interacting directly with contracts 22
    7. Issuing the Certificate 23
    8. Enter details 23
    9. Click issue to initiate transaction ad click confirm 24
    10. Transaction details 24
    11. Enter hash of the certificate to claim 25
    12. Click on claim and confirm the transaction 26
    13. State update in UI to all available certificates of the user stored on chain 27
    14. Transaction details 28
    15. Verifying a Certificate 29
    16. Enter hash and holder public key 30
    17. Valid Response implies a Certificate with hash is owned by anticipated holder 31
    18. Invalid hash implies a Certificate with entered hash is not valid 32

**List of Abbreviations**

|  |  |
| --- | --- |
| CSE | Computer Science Engineering |
| WP | **Web 3 provider** |
| ER | Entity Relationship |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| TS | Type Script |
| BSC | Binance Smart Chain |

# INTRODUCTION

## INTRODUCTION

Children are the greatest asset of each nation. The future of any country depends upon the right upbringing of its children. India is the second populous country in the world and children represent a significant percentage of total population. But unfortunately a large number of children go missing every year in India due to various reasons including abduction or kidnapping, run-away children, trafficked children and lost children. A deeply disturbing fact about India’s missing children is that while on an average 174 children go missing every day, half of them remain untraced. Children who go missing may be exploited and abused for various purposes. As per the National Crime Records Bureau (NCRB) report which was cited by the Ministry of Home Affairs (MHA) in the Parliament (LS Q no. 3928, 20-03- 2018), more than one lakh children (1,11,569 in actual numbers) were reported to have gone missing till 2016, and 55,625 of them remained untraced till the end of the year. Many NGOs claim that estimates of missing children are much higher than reported.

Mostly missing child cases are reported to the police. The child missing from one region may be found in another region or another state, for various reasons. So even if a child is found, it is difficult to identify him/her from the reported missing cases. A framework and methodology for developing an assistive tool for tracing missing child is described in this paper. An idea for maintaining a virtual space is proposed, such that the recent photographs of children given by parents at the time of reporting missing cases is saved in a repository. The public is given provision to voluntarily take photographs of children in suspected situations and uploaded in that portal. Automatic searching of this photo among the missing child case images will be provided in the application. This supports the police officials to locate the child anywhere in India.

When a child is found, the photograph at that time is matched against the images uploaded by the Police/guardian at the time of missing. Sometimes the child has been missing for a long time. This age gap reflects in the images since aging affects the shape of the face and texture of the skin. The feature discriminator invariant to aging effects has to be derived. This is the challenge in missing child identification compared to the other face recognition systems. Also facial appearance of child can vary due to changes in pose, orientation, illumination, occlusions, noise in background etc. The image taken by public may not be of good quality, as some of them may be captured from a distance without the knowledge of the child. A deep learning [1] architecture considering all these constrain is designed here.

## MOTIVATION

A countless number of children go missing every year. The category of missing children include a number of problems including abduction or kidnapping of children by family members and by non-family members, run-away children or those forced to run away by family and surrounding circumstances, children who are in a difficult or aggressive environment, trafficked children, and lost children. The convention way of finding a missing person is to broadcast picture, from poster to News Channel. There is no system to identify the photographs of children with different lighting conditions, noises and also images at different ages of children.

## PROPOSED STATEMENT

This system proposes a methodology for missing child identification which combines facial feature extraction based on deep learning and matching based on KNN. The proposed system utilizes face recognition for missing child identification. This is to help authorities and parents in missing child investigation. This system is evaluated with the deep learning model which is trained with feature representations of children faces.

## PROJECT OBJECTIVES

The proposed system is comparatively an easy, inexpensive and reliable method compared to other biometrics like finger print and iris recognition systems. Features extracted using a CNN network for getting facial representations gives better performance in face recognition than handcrafted features.

## PROJECT REPORT ORGANIZATION

1. This report is divided into 6 chapters after this introductory chapter.
2. Chapter 2 summarizes functional, non-functional requirements and system requirements along with software and hardware specifications.
3. Chapter 3 deals with analysis and design of the proposed model which includes use case diagram etc.
4. Chapter 4 encloses Implementation and testing of the proposed model and testing with different scenarios.
5. Chapter 5 includes conclusion and future work.
6. Chapter 6 includes reference.

# LITERATURE SURVEY

## LITERATURE SURVEY

The literature review uncovered distinct design elements commonly discussed in research that affect user engagement. They were

* Blockchain: Blockchain can better be understood as an immutable database and laid the foundation of the whole project. It provides a trusted environment where actions have done are visible and can’t be tampered with.
* Ethereum: Ethereum is a decentralized open-source Blockchain featuring smart contract functionality. Ethereum is itself the best example of Blockchain and its a cryptocurrency system which is the most widely used and is the next expensive cryptocurrency after bitcoin.
* Smart Contract: Smart contracts piece of code that runs on a Blockchain when a user performs some action. A Smart contract is written in many different languages including low- level languages like C++, Java, and high-level languages like Solidity which is closely similar to Typescript.
* Solidity: Solidity is an object-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various Blockchain platforms, most notably, Ethereum. It is closely similar to Typescript but with more specific data types.
* Ethash: Ethash is the proof-of-work function in Ethereum-based Blockchain currencies. It is a hash function belonging to the Keccak family, the same family to which the SHA-3 hash functions belong to. However, Ethash is not an SHA- 3 function, and should not be confused with them.
* IPFS: The InterPlanetary File System is a peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices.
* Metamask: Metamask is an extension for accessing Ethereum enabled distributed applications or DApps in your browser. The extension injects the Ethereum web3 API into every websites JavaScript context so that DApps can read from the blockchain.
* Ganache: Ganache is used for testing Solidity contracts on a personal Ethereum Blockchain. It by default provides an easy setup for spinning up a network with around ten users with each having 100 eths on their account. These accounts can be used to mimic the transactions between the users.
* Truffle: Truffle provides easy compilation, linking, deployment, and binary management of smart contracts written in solidity language.
* Goerli: Goerli provides an Ethereum test network which is used by developers to test their code and perform some actions. It just provides an environment that works on proof of authority, unlike Ethereum which works on proof of work.
* ECDSA: An Elliptic Curve Cryptography algorithm base on field mathematics that is used to generate public and private keys I the Ethereum network. The transactions are signed and verified using the same technique for checking integrity and authenticity.

For each of the above, we calculated the proportion of studies mentioning the element. In this review, we identified elements that were related to the CSE department and included these.

* 1. **EXISTING SYSTEM**
* Mostly missing child cases are reported to the police. The child missing from one region may be found in another region or another state, for various reasons.
* Even if a child is found, it is difficult to identify him/her from the reported missing cases.

* 1. **LIMITATIONS OF EXISTING SYSTEM**
* A deeply disturbing fact about India’s missing children is that while on an average 174 children go missing every day, half of them remain untraced.
* There is so system to identify the photographs of children with different lighting conditions, noises and also images at different ages of children.

# SOFTWARE & HARDWARE SPECIFICATIONS

## Functional/Non-Functional Requirements

* + 1. **Functional Requirements**

Functional requirement are the functions or features that must be included in any system to satisfy the model needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows.

System should be able to make state changes consistently without slot collision

System should be able to offer read only service for verifying.

## Non-Functional Requirements

The performance of the system should be fast and accurate.

The system should be able to handle large amount of data. Thus, it should accommodate large number of data entry without any fault.

User friendly.

## SOFTWARE REQUIREMENT

The following few tools have been used to perform this analysis

* Solidity: For writing smart contract.
* Hardhat: Environment for testing and developing smart contracts.
* Wallet: Any blockchain wallet with chain support. (E.g., Metamask)
* React: Front end framework for UI design
* Node : A runtime for running TS and JS scripts
* Network: Any EVM based network BSC, Ethereum, Polygon etc.
* Execution clients: GETH, Erigon etc. (for running private network)
* Consensus clients: PRYSM, TEKU, NIMBUS etc. (for running private network)

## HARDWARE REQUIREMENTS

## For running on existing networks:

RPC client: Alchemy, Infura and Quicknode offer RPC client services by running their own node in the network.

## For running a private network:

Processor: 2 GHz, multi-core processor

RAM: 4GB

Secondary Storage: 256GB

## For running DApp:

Any smart phone or computer with browser and web 3 wallet injector.

# DESIGN

## USE CASE DIAGRAM

**Diagram

Description automatically generated**

**Figure 4.1: Use Case Diagram**

In our Use Case Diagram, we have three actors namely issuer, holder and verifier. The use case diagram shows the set of use case actor and their relationships. The actors perform the respective use cases mentioned above. The issuer plays a role of adding holders and issuing certificates to the holders. Holders can claim their valid certificates. Verifiers can verify any certificate against its hash.

## ACTIVITY DIAGRAM

## ACTIVITY DIAGRAM FOR ADMIN

**Diagram

Description automatically generated**

**Figure 4.2.1: Activity Diagram for issuer and admin**

The activity diagram provides a view of the classes and what is going on between several classes and use cases. The admin or the issuer of the contract can upgrade the implementation address of a contract, even issue certificates, add holders to the state of the registrar smart contract and verify certificates.

## ACTIVITY DIAGRAM FOR HOLDERS

**Diagram

Description automatically generated**

**Figure 4.2.2: Activity Diagram for holder**

Holder can claim their valid certificate issued by the admin, add verifier to their certificate and verify certificates too.

## ACTIVITY DIAGRAM FOR VERIFIERS

Diagram

Description automatically generated

**Figure 4.2.3: Activity Diagram for holder**

Verifiers assigned by holders have a period in which they can verify the certificates of the respective holders

## CLASS DIAGRAM

**Diagram

Description automatically generated**

**Figure 4.3: Class Diagram**

The above class diagram depicts all the class, objects and functions of the system deployed on the Goerli Ethereum test net.

## ER DIAGRAM

## Diagram Description automatically generated

**Figure 4.4: ER Diagram**

An Entity Relationship illustrates how “entities” such as people, objects or concepts relate to each other within a system. In our ER diagram we have four entities namely holder, verifier, registrar and certificates.

The relation between entities is as follows:

o Holders-Certificates: It is a many-to-many relationship since any number of holders can hold and claim any numbers of validly issued certificates.

o Verifier-Certificates box: It is a many -to-many relationship since any number of verifiers assigned by holders can verify to any number of certificates against the holder’s public key.

o Admin -Registrar: It is a one-to-one relationship since one registrar can have only one admin or issuer.

o Holder-Registrar: It is a many-to-one relationship since there can be any number of holders in the registrar.

## DEPLOYMENT DIAGRAM

## Diagram Description automatically generated

**Figure 4.5: Deployment Diagram**

A deployment diagram shows the execution architecture of a system, including nodes such as hardware or software. In the above diagram the artifacts that are deployed on Goerli test net and depicts all the contract in the network. The RPC client is essential for broadcasting the transaction to the network. DApp (Decentralised application) is hosted on a IPFS network or using a web server which serves UI pages dynamically.

## COMPONENT DIAGRAM

Diagram

Description automatically generated

**Figure 4.6: Component Diagram**

A component diagram describes the organization and writing of the physical components in a system. In the above component diagram, we have 5 components namely user, faculty, student, admin, and database. So, the main users of the website are holders, verifiers and issuer. Issuer or admin is responsible for controlling the issuing of digital certificates. Verify can verify certificates on demand and holders can claim and maintain their educational records in the smart contract on the blockchain network.

# 5. IMPLEMENTATION AND TESTING

## IMPLEMENTATION

## Graphical user interface, website Description automatically generated

## Figure 5.1.1 View Page

**Graphical user interface, website

Description automatically generated**

## Figure 5.1.2 Issue Page

Graphical user interface, chat or text message

Description automatically generated

## Figure 5.3 Verification Page

## Graphical user interface, text, application Description automatically generated

## Figure 5.4 Component for viewing public key of user

## Graphical user interface, text, application Description automatically generated

## Figure 5.5 Metamask Wallet InterfaceGraphical user interface, application Description automatically generated

## Figure 5.6 Remix Interface for interacting directly with smart contracts

## TESTING

## Issuing the certificate

## Graphical user interface, website Description automatically generated

## Figure 5.2.1.1 Enter details

## Graphical user interface, website Description automatically generated

## Figure 5.2.1.2 Click Issue to initiate transaction and click confirm

**Graphical user interface, text, application

Description automatically generated**

## Figure 5.2.1.3 Transaction details

## Claiming the certificate

## Graphical user interface, website Description automatically generated

## Figure 5.2.2.1 Enter hash of the certificate to claim

## Graphical user interface, website Description automatically generated

## Figure 5.2.2.2 Click on claim and confirm the transaction

## Graphical user interface, website Description automatically generated

## Figure 5.2.2.3 State update in UI to all available certificates of the user stored on chain

## Graphical user interface, text, application Description automatically generated

## Figure 5.2.2.4 Transaction details

## Verifying a certificate

## Graphical user interface, website Description automatically generated

## Figure 5.2.3.1 Enter hash and holder public key

## Graphical user interface, text, website Description automatically generated

## Figure 5.2.3.2 Valid response implies a certificate with hash is owned by the anticipated holder

## Graphical user interface, website Description automatically generated

## Figure 5.2.3.3 Invalid response implies a certificate with entered hash was never claimed by the holder

## 

**6. CONCLUSION AND FUTURE ENHANCEMENT**

* 1. **CONCLUSION:**

Creating immutable ledgers is one of the main values of Blockchain. This behaviour helps us to achieve a system in which all the process is transparent and unchangeable. Our System automates the process of generating Certificates and reduces the manual work needed for the verification of the same. Students are also at comparatively low risk of losing the certificate. By using an additional hashing algorithm we are decreasing the percentage of data being tampered with. The Hash of the certificate is being stored in the blockchain while the original document will be stored in the Inter Planetary File System (IPFS). This will help us preserve the data and create transparency.

## FUTURE ENHANCEMENT:

* + - We can further extend the functionality for UI elements to include adding verifier and timestamp for holders.
    - Create a relayer/forwarder to do the transactions instead of users spending the gas.
    - Run the contracts on an app-chain network maintained by aggregated colleges and universities.

# REFERENCES

1. <https://www.researchgate.net/publication/337866436_Frauds_Avoidance_of_Digital_Educational_Certificates_Using_Blockchain>
2. <https://www.cs.miami.edu/home/burt/learning/Csc609.142/ecdsa-cert.pdf>
3. <https://docs.soliditylang.org/en/v0.8.17/>