Teen Credence: Ensuring Trustworthy Adolescents Registration and Activity Monitoring on Social Media Platform

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Nitika Bhatt (201150)

Rohit Mishra (201465)

Under the guidance & supervision of

Dr. Shubham Goel

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

Certificate

This is to certify that the work which is being presented in the project report titled "**Teen Credence: Ensuring Trustworthy Adolescents Registration and Activity Monitoring on Social Media Platform**" in partial fulfillment of the requirements for the award of the degree of B.Tech in **Computer Science & Engineering** and submitted to the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Nitika Bhatt (201150) and Rohit Mishra (201465) during the period from August 2023 to May 2024 under the supervision of **Dr. Shubham Goel**, Department of **Computer Science & Engineering**, Jaypee University of Information Technology, Waknaghat.

Student Name: Nitika Bhatt
Roll No.: 201150

Student Name: Rohit Mishra
Roll No.: 201465

The above statement made is correct to the best of our knowledge.

Dr. Shubham Goel
Assistant Professor (SG)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat, Solan
Dated:

Candidate's Declaration

We hereby declare that the work presented in this report entitled **'Teen Credence: Ensuring Trustworthy Adolescents Registration and Activity Monitoring on Social Media Platform'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of our own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Shubham Goel,** Department of **Computer Science & Engineering**, Jaypee University of Information Technology, Waknaghat.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Name: Nitika Bhatt
Roll No.: 201150

Student Name: Rohit Mishra
Roll No.: 201465

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Shubham Goel
Assistant Professor (SG)
Computer Science & Engineering and Information Technology
Dated:

Acknowledgment

Firstly, we express our heartiest thanks and gratefulness to almighty God for His divine blessing which made it possible to complete the project work successfully.

We are grateful and wish our profound gratitude to our Supervisor **Dr. Shubham Goel, Assistant Professor (SG)**, Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of our supervisor in the field of "**Teen Credence: Ensuring Trustworthy Adolescents Registration and Activity Monitoring on Social Media Platform**" helped us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Shubham Goel,** Department of Computer Science & Engineering, for his kind help to finish my project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated our undertaking.

Finally, we acknowledge with due respect the constant support and patience of our parents.

Student Name: Nitika Bhatt
Roll No.: 201150

Student Name: Rohit Mishra
Roll No.: 201465

Table of Content

List of Tables

List of Figures

List of Abbreviations, symbols or nomenclature

S. No.	Abbreviations	Full Form
1	TC	Teen Credence
2	CNN	Convolution Neural Network
3	CUDA	Compute Unified Device Architecture
4	DLDL	Deep Label Distribution Learning
5	MAE	Mean Absolute Error
6	IEEE	Institute of Electrical and Electronics Engineers

Abstract

The goal of the project titled, "TEEN CREDENCE: Ensuring Trustworthy Adolescents Registration & Activity Monitoring on Social Media Platform" is to provide a trustworthy and safe framework for the registration and surveillance of teenagers' online activities. In a time when digital connections have a significant influence on the development of young brains, it is critical to make sure these interactions are trustworthy. This effort aims to provide a supportive atmosphere for teens' responsible and constructive use of social media by putting strong safeguards in place to protect their online personas.

TC strives to reduce possible hazards related to false information, cyber bullying, and inappropriate content by utilizing cutting-edge registration methods and ongoing activity monitoring. This creative strategy places a high priority on protecting teenagers while also attempting to equip them with the knowledge and resources needed for a positive and safe online experience.

TC is essentially an initiative to create a digital environment that supports the development and welfare of the younger generation by encouraging accountability and trust in their online interactions. TC will modify its tactics to keep ahead of new obstacles since it acknowledges that social media platforms are dynamic. TC seeks to establish a new benchmark for teens' responsible use of social media by establishing a cooperative ecosystem and encouraging a climate of trust, empathy, and good digital citizenship. This project is in line with the larger social objective of utilizing digital platforms to their fullest potential while reducing the hazards that come with their usage, in terms of social media platform.

Chapter 1: Introduction

1.1 Introduction

In an era dominated by the digital landscape, the influence of social media on the lives of adolescents is profound and pervasive. Social media has a significant and widespread impact on teenagers' lives in a time when the digital world rules.

The "TEEN CREDENCE" program is a welcome response to the growing need for a trustworthy, accountable, and safe space for young people navigating the complex world of social media. With a primary focus on upholding the integrity of adolescent registration procedures and continuous social media activity monitoring, TC is a leader in the protection and empowerment of the younger generation. In an attempt to create a digital environment where teens may grow and develop positively without having to worry about potential threats, this program goes above and beyond traditional security measures.

TC employs cutting-edge registration procedures, sophisticated monitoring systems, and instructional modules in an effort to carefully balance privacy and security. Teens will be empowered to confidently traverse the digital landscape while also being encouraged to engage in responsible online behavior. As we embark on this journey, TC is a comprehensive strategy that employs a multifaceted approach. It acknowledges that social media platforms are dynamic and envisions a collaborative effort among stakeholders to rethink the parameters of online engagement for the advantage and welfare of the next generation.

1.2 Problem Statement

Adolescents in the quickly changing social media world are at the crossroads of vulnerability and technology, dealing with a multitude of issues that require immediate attention. The "TEEN CREDENCE" project highlights a critical problem stemming from the shortcomings of the existing systems in place for registering teenagers and keeping an eye on their social

media activity. The reliability of these procedures is frequently not sufficiently ensured by current protocols, putting teenagers at risk for inappropriate content, misinformation, and cyber bullying, among other things. This critical problem statement highlights the need for a comprehensive and proactive approach to protect the digital health of teenagers by addressing registration security flaws and enhancing real-time monitoring capabilities.

TC is on a mission to improve the reliability of these procedures in order to offer a complete solution that will protect teenagers from inappropriate content and cyber threats while also equipping them with the skills and information necessary to use the internet responsibly.

The driving force behind this initiative is a dedication to promoting good digital citizenship, which allows teenagers to take advantage of the advantages of internet connectivity while remaining safe.

TC recognizes the need to establish an online space that safeguards the security and privacy of teenagers while also equipping them with the information and abilities required to engage in morally and responsibly as digital citizens. The project aims to bridge these gaps and build a more dependable, safe digital environment that will promote the growth of the next generation through innovative solutions, astute algorithms, and educational initiatives.

1.3 Objectives

1. The primary objective of the "TEEN CREDENCE" project is to establish a robust framework that ensures the trustworthy registration and continuous activity monitoring of adolescents on social media platforms.

2. Establishing a strong framework to guarantee the reliable registration and ongoing activity monitoring of teenagers on social media platforms is the main goal of the "TEEN CREDENCE" project.

3. The project intends to develop novel solutions that improve the integrity of the

registration process, reducing the risks associated with cyber threats, false information, and inappropriate content. This is in response to the urgent need for a secure digital environment.

4. TEEN CREDENCE aims to enable intelligent and privacy-respecting monitoring of teenagers' online activities by utilizing artificial intelligence and advanced algorithms. This approach strikes a careful balance between protecting teenagers' wellbeing and honoring their autonomy.

1.4 Significance and Motivation of the project

The "TEEN CREDENCE" project's significance and driving force stem from the necessity to confront the significant challenges that young people face in navigating the increasingly intricate digital terrain. Social media has a big impact on teens' social and emotional development these days, so it's important to create a trustworthy and safe online environment.

The project recognizes the weaknesses in the current registration and activity monitoring processes for social media platforms, which put teenagers at risk for long-term consequences to their wellbeing.

TC is on a mission to improve the reliability of these procedures in order to offer a complete solution that will protect teenagers from inappropriate content and cyber threats while also equipping them with the skills and information necessary to use the internet responsibly.

The driving force behind this initiative is a dedication to promoting good digital citizenship, which allows teenagers to take advantage of the advantages of internet connectivity while remaining safe. The project has wide-ranging implications for society, as it aims to foster the development of a resilient, digitally literate generation that can use social media for positive ends.

This initiative's main motivation is a commitment to fostering responsible digital citizenship, which enables teenagers to benefit from internet connectivity while staying safe. The project intends to support the growth of a resilient, digitally literate generation that can use social media for positive purposes, which has broad implications for society.

By addressing this pressing need and offering a proactive, comprehensive approach to the protection and empowerment of adolescents in the digital age, TEEN CREDENCE aims to empower parents, educators, and legislators.

1.5 Organization of Project Report

The organization of the Project Report is in the following format.

Chapter 1: Introduction

1. This chapter provides an overview of the "TEEN CREDENCE" project and places the need for trustworthy adolescent registration and social media activity monitoring in realistic perspective.

2. It outlines the objectives, the project's primary goal, and the significance of giving teenagers access to a secure online environment.

3. The chapter lays the groundwork for subsequent sections by giving readers a clear road map for comprehending the goals and importance of the project.

Chapter 2: Literature Survey

1. This chapter explores in great detail the academic publications and recent studies that form the basis of "TEEN CREDENCE." The literature review closely examines topics like teenage digital experiences, privacy issues, and teen internet safety.
2. This chapter offers a strong foundation for the project by critically analyzing current

registration processes, activity tracking systems, and technological advancements.

3. It closes the gaps and clarifies the conclusions drawn from the literature to inform the subsequent stages of development and implementation.

Chapter 3: System Development

1. This important chapter highlights the methods and tools for precise teen registration and in-the-moment social media activity monitoring, illuminating the challenges in developing the "TEEN CREDENCE" system.

2. The system development process is deconstructed, encompassing everything from the application of cutting-edge technologies like artificial intelligence and complex algorithms to the curation, integration, and augmentation of datasets.

3. The chapter serves as a roadmap, guiding readers through the technical specifics and decision-making processes that make up the solid foundation ensuring the project's success in giving teenagers a secure online environment.

Chapter 4: Testing

1. This chapter conducts a thorough testing procedure to evaluate the efficacy of the "TEEN CREDENCE" system. Extensive techniques for evaluating the predictability of age and the effectiveness of the system in practical scenarios are examined.

2. This chapter describes the metrics, testing standards, and validation processes that were used to assess the dependability and resilience of the model. The system's strong points and areas for improvement are highlighted in the results of testing iterations.

3. Verifying the project's efficacy in safeguarding teenagers on social media and verifying its capabilities are the main goals of this chapter.

Chapter 5: Results and Evaluation

1. The main conclusions, successes, and ramifications of the "TEEN CREDENCE" project are summarized in this final chapter. It emphasizes how well the project managed to guarantee reliable adolescent registration and social media activity tracking.

2. The initiative's wider effects on teen internet Safety are also covered in this chapter. The chapter also looks at potential directions for future study and innovation, offering improvements and developments to fortify the system even more.

3. This chapter lays the foundation for ongoing innovation and development in the field of adolescent digital well-being by describing prospective areas of growth, technology integration, and changing social media landscapes.

Chapter 6: Conclusion and Future Scope

1. The main conclusions, successes, and ramifications of the "TEEN CREDENCE" project are summarized in this final chapter. It emphasizes how well the project managed to guarantee reliable adolescent registration and social media activity tracking.

2. The initiative's wider effects on teen internet Safety is also covered in this chapter. The chapter also looks at potential directions for future study and innovation, offering improvements and developments to fortify the system even more.

3. This chapter lays the foundation for ongoing innovation and development in the field of adolescent digital well-being by describing prospective areas of growth and technology.

Chapter 2: Literature Survey

The literature survey conducted for the "TEEN CREDENCE" initiative, focusing on ensuring trustworthy adolescent registration and activity monitoring on social media platforms, delves into the existing body of knowledge regarding online safety, privacy concerns, and the unique challenges faced by teenagers in the digital realm.

Numerous studies emphasize the escalating importance of safeguarding adolescents in the context of social media usage. Privacy breaches, cyberbullying, and exposure to inappropriate content are identified as prominent risks. Recognizing these challenges, the literature highlights the necessity for robust frameworks that secure the registration process and ensure continuous, responsible monitoring of teenagers' online activities.

Studies examining existing registration protocols underscore their shortcomings, emphasizing the need for improvements. Many protocols lack the necessary mechanisms to verify the authenticity of adolescent users, potentially exposing them to risks associated with false information and malicious activities. The literature survey scrutinizes various registration models and identifies gaps that can compromise the overall security of teenagers' online experiences.

In terms of activity monitoring, the survey explores prevailing methodologies and tools. Current monitoring systems often fall short in providing real-time insights into adolescents' online interactions, making it challenging to promptly address issues such as cyberbullying or inappropriate content consumption. The literature underscores the significance of proactive monitoring strategies that balance the need for security with respecting teenagers' autonomy.

2.1 Overview of relevant literature

Following is the relevant overview of literature:

1. The "TEEN CREDENCE" project's literature review provides an extensive summary of previous studies, illuminating the complex world of teenage registration and activity tracking on social media platforms.

2. It looks at the common risks that teens face when using the internet, highlighting the necessity of strong frameworks to protect teens' privacy and safety.

3. The survey identifies vulnerabilities that compromise the safety of teenagers using the internet by critically evaluating the shortcomings of the existing registration procedures and activity tracking systems.

4. The literature review provides a basis for creating novel solutions for the TC project by utilizing developments in algorithmic monitoring and artificial intelligence.

S. No.	Paper Title [Cite]	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results	Limitations
1.	Deep Domain – invariant learning for facial estimation.	IEEE/2023	Domain-invariant and style-invariant modules. DLDL- v2 and ResNet-18.	FGNET accuracy – 81.5% and MORPH accuracy – 96.5%.	Selective datasets with less age gap. FGNET fails to retrieve features.
2.	Gender and Age Detection using Deep Learning.	Conf./2021	UTK Face Dataset	Gender Detection accuracy - 96.2% and Age estimation accuracy – 93.7%.	Less efficient model algorithm.
3.	A Deep Learning approach for Face Detection using Max Pooling.	2021	Tools: Python, Tensorflow. Techniques: CNN, Max Pooling, Image Augmentation. Dataset: LFW.	Training set accuracy – 95.72% and Validation set accuracy – 96.27%.	Limited to LFW Dataset.
4.	Age from Faces in the Deep Learning Revolution.	IEEE/2020	Dataset: UTKFace, MORPH II, FG-NET.	Mean absolute errors of around 3-4 years on some dataset.	No new experimental results.

5.	Age – invariant face recognition based on deep features analysis	2020	Algorithm: Viola – Jones. Techniques: CNN, VGG-Face model, KNN, SVM. Dataset: FGNET, MORPH.	FGNET accuracy – 81.5% and 96.5% MORPH.	Selective datasets contain less age gap. FGNET fails to retrieve features.
6.	Deep Age Estimation	IEEE/2020	Algorithm: Deep CNN. Technique: Deep label Distribution Learning.	Automaticall y learning of comp -lex face features from v4	Reasonable number of neighboring ages.
7.	Automatic Age Estimation Based on Deep Learning Algorithm	Neuro computing 2016	Algorithm: Deep CNN. Images of Groups of people (IGP – 10K)	Mean Absolute (MAE) of 3.4 years on the IGP-10K dataset and 3.8 on MORPH II.	Small Dataset.
8.	DEX: Deep Expectation of apparent age from a single image	IEEE/ 2015	Algorithm: Deep CNN. Deep Expectation (DEX). Technique: CNN, VG16.	Better working of softmax value.	Still Image. Dataset do not represent diverse population.

9.	Age and Gender Estimation of Unfiltered Faces.	IEEE/ 2014	Tools: MatConvNet. Model: LIBSVM Dataset: Audience Faces.	Group estimation accuracy 96.2% and Gender Classification 99.5%.	Small Dataset.
10.	Age and gender Recognition using Deep Learning.	IJCRT/ 2022	PyTorch OpenCV GPU Acceleration (CUDA)	Gender Detection accuracy - 94.1% and Age estimation accuracy – 95.3%.	Dataset do not represent diverse population.

Table 2.1: Literature review

2.2 Key gaps in the Literature

Following are the major key gaps in the literature review:

1. The existing models in literature review require high computational power like GPU support for good accuracy and less run time.

2. There is no open-source dataset available covering all ethnic group around the world, like no proper dataset for Indian subcontinent ethnic people which makes our model a little inaccurate in real world scenarios.

3. No possible way to detect age when facial features are hidden, hence limited to face for age detection can be difficult to get appropriate results.

4. Many of the models that are currently in use have trouble extrapolating age predictions across a wide range of age groups. According to the literature review, there appears to be a discrepancy in attaining uniform accuracy throughout the whole age range, as certain models exhibit biases in favor of age groups.

5. Age-related changes in facial expressions present a problem for current age detection models. The review of the literature suggests that handling dynamic facial expressions inconsistently could result in errors.

6. The review of the literature indicates that the investigation of transfer learning techniques in age detection models has received little attention.

7. The review points out a void in the discussion of fairness, potential biases, and the moral ramifications of using age detection systems.

Chapter 3: System Development

3.1 Requirements and Analysis

The software tools we used to carry out our project include a variety of potent frameworks and libraries designed specifically for computer vision and machine learning applications.

Table 3.1: Requirements and Analysis

Category	Description	
Software Resources	• TensorFlow • Pytorch • Open CV • Python • CUDA • CUDNN	• Version: 2.8 • Version: Latest stable • Version: compatible with TensorFlow / Pytorch • Version: 3.9 • Version: 11.2 • Version: 8.1.0
Hardware Resources	• Camera • Best GPU System	
Others	• Matplotlib • Seaborn • Scipy • Sklearn • Dataset	

Description of the table is as follows:

1 Software Resources

a. *TensorFlow*, *PyTorch*: These powerful frameworks are integral to the development and training of our model, providing stability and adaptability for computer vision and machine learning applications.

b. *OpenCV*: Inclusion of OpenCV enhances image processing capabilities, enabling sophisticated computer vision functions for effective data analysis.

2 Hardware Resources

a. *High-Quality Camera*: The core hardware includes a high-quality camera essential for capturing facial images required for age prediction.

b. *Optimal GPU System*: Utilizing the best GPU system ensures optimal performance, enhancing the speed and accuracy of our model during both training and inference processes.

3 Additional Libraries & Tools

a. *Seaborn and Matplotlib*: These libraries aid in data visualization, offering insights into dataset properties and model performance.

b. *Scipy*: Integration of Scipy improves technical and scientific computing capabilities.

c. *Scikit-learn (Sklearn):* Sklearn adds effective data analysis and modeling tools to our machine learning toolkit.

d. *Dataset*: The dataset forms the foundation for both model training and evaluation, serving as an essential component of our project. The dataset includes various face images enhanced to increase realism and variability. Carefully selected to meet the unique needs of teenage registration and social media activity tracking, ensuring system stability and diversity.

The software tools we used to carry out our project include a variety of potent frameworks and libraries designed specifically for computer vision and machine learning applications. Our model's development and training are supported by TensorFlow, PyTorch, and Keras, which offer stable and adaptable frameworks. With OpenCV included, we can process

images more effectively and perform sophisticated computer vision functions. The hardware resources required for our project are made up of necessary parts that guarantee effective image processing and capture. The core hardware consists of a high-quality camera that takes the essential facial pictures for age prediction. Using the best GPU system allows us to achieve optimal performance, which improves our model's speed and accuracy during training and inference.

Our toolkit contains several additional libraries and tools in addition to the basic software and hardware resources. Data visualization is facilitated by Seaborn and Matplotlib, which provide information on model performance and dataset properties. Scipy's integration improves technical and scientific computing capabilities, and Scikit-learn (Sklearn) adds practical modeling and data analysis tools to our machine learning toolkit.

Since the dataset serves as the basis for both model training and evaluation, it is an essential part of our work. To increase variety and realism, it includes a range of digitally enhanced face images. The dataset has been carefully selected to meet the unique requirements of tracking social media activity and teen registration, while also ensuring the stability and diversity of our system.

Following is the detail description of all types of resources:

1 *TensorFlow***:** TensorFlow is an essential tool in the machine learning and artificial intelligence domains. TensorFlow is an open-source framework that is widely known for its scalability and flexibility. It enables researchers and developers to create and use machine learning models on a variety of platforms. Its vast toolkit and library make it easier to develop applications for everything from natural language processing to speech and image recognition.

Characteristics of TensorFlow

a) *Flexibility and Portability*: TensorFlow’s flexible structural design enable smooth deployment across many platforms, such as CPUs, GPUs, and TPUs (Tensor Processing Units), in turn fosters an ecosystem that is flexible.

b) *Entire Toolkit*: TensorFlow's big toolkit enable it to handle a assortment of machine knowledge tasks, such as model making, training, and deployment.

c) *TensorBoard*: An interactive graph and metrics hallucination toolkit for TensorFlow, TensorBoard aids in considerate and civilizing model piece.

d) *Mechanism*: Notable mechanism of TensorFlow that have contributed to its widespread adoption are its documentation and community. They serve a broad range of talent levels and provide resources for both inexpert and seasoned practitioners.

2 *PyTorch:* PyTorch is a progressive deep learning structure that is well-known for its easy-to-use interface and active computational graph. Python mixing and an intuitive UI make PyTorch, developed by Facebook's AI Research lab (FAIR), a popular implement among scientists, engineers, and academics.

Characteristics of PyTorch

a) *Dynamic Computational Graph*: PyTorch's dynamic computation graph offers flexibility in model development and debugging by enabling dynamic and on-the-fly graph computation.

b) *Pythonic Interface*: PyTorch's user-friendly interface, coupled with its Pythonic syntax, renders it suitable for developers of all experience levels.

c) *Execution*: PyTorch implements eager execution, which allows programmers to carry out operations right away. This makes the process of developing models more interactive and user-friendly.

d) *Usage*: PyTorch facilitates the use of TorchScript, which enables the serialization and optimization of models for production environments. This facilitates a smooth transition from development to deployment.

e) *Libraries for Developing Models*: Torchvision and Torchtext are two of the many libraries that PyTorch offers for developing and training deep learning models in computer vision and natural language processing, respectively.

3 *KERAS:* Keras is a high-level neural networks API that facilitate the creation and education of deep learning models. At first developed as a stand-alone project, Keras now provide a convenient concept layer for user of all skill levels, making it an vital part of the TensorFlow bionetwork.

Characteristics of Keras

a) *User-Friendly Interface:* Deep learning model growth and training are made easier with the help of Keras, an advanced neural network API. At first created as a stand-alone project, Keras is now an essential part of the TensorFlow ecosystem, contribution a convenient concept layer for user of all skill levels.

b) *Modularity:* Keras takes a modular approach, enabling users to create models quickly and efficiently by assembling neural network layers like building blocks.

c) *Backend Compatibility:* Initially designed to be backend-neutral, Keras interacts with many deep learning frameworks with ease; TensorFlow is the most popular and default backend.

d) *Extensibility:* Keras offers the suppleness required for trendy model architectures by allow users to create custom layer and models.

e) *Model Visualization:* The advanced neural network API Keras makes it easier to develop and train deep learning models. Keras, which was first developed as a stand-alone project, is now a crucial component of the TensorFlow ecosystem and provides a practical abstraction layer for users of all experience levels.

4 *OpenCV:* Deep learning model development and guidance is made simpler by the difficult neural set of connections API Keras. Initially created as a stand-alone project, Keras is now an essential part of the TensorFlow ecosystem and offers a useful abstraction layer to users with varying levels of expertise.
Characteristics of OpenCV

a) *Rich Functionality:* OpenCV is a flexible toolkit for a broad range of applications because it includes a rich set of functions for image processing, computer vision, machine learning, and deep learning.

b) *Cross-Platform:* OpenCV is made to run on multiple operating systems, including Windows, Linux, and macOS. This feature enables developers to easily launch their apps on various platforms.

c) *Real-time Processing*: OpenCV's optimized algorithms enable real-time processing of images and videos, which makes it appropriate for augmented reality, object detection, and face recognition applications.

d) Support from the Community: OpenCV has a sizable and vibrant community that regularly adds new features, enhancements, and a multitude of resources, such as documentation and tutorials.

e) *Integration with Other Libraries*: TensorFlow and PyTorch are just two of the widely used libraries and frameworks with which OpenCV integrates well, allowing for a synergistic approach in computer vision and machine learning projects.

5 Hardware Resources

a) *Camera***:** As a key piece of hardware for the project, the camera is responsible for taking vital facial images that are needed for the age prediction method. The accuracy and efficiency of image processing tasks are directly impacted by the quality and capabilities of the camera.

b) *GPU system:* Including the best GPU system is essential to maximizing the project's

performance. Large-scale deep learning computations and real-time applications require fast and accurate model training and inference processes, which can only be achieved with a robust GPU (Graphics Processing Unit).

6 Others

a) *Matplotlib:* A Python plotting library called Matplotlib makes it easier to create static, animated, and interactive visualizations. Matplotlib probably helps to visualize model performance metrics, dataset properties, and other pertinent information in the project context.

b) *Seaborn:* A Matplotlib-based library for statistical data visualization is called Seaborn. It gives Matplotlib an extra layer of abstraction and aesthetics, which facilitates the creation of visually stunning and educational statistical graphics. Seaborn might improve the data's visual representation in the project.

c) *Scipy:* An open-source library for science, math, and engineering is called Scipy. For tasks like optimization, integration, interpolation, eigenvalue problems, and more, it significantly expands Python's capabilities. It is possible that Scipy enhances scientific and technical computing capabilities in the project, adding to its overall functionality.

d) *Sklearn*: A machine learning library called Scikit-learn offers quick and easy-to-use tools for modeling and data analysis. It has a number of tools for dimensionality reduction, clustering, regression, and classification, among other things. In the project, Scikit-learn expands the machine learning toolkit with useful modeling and data analysis tools, potentially assisting with tasks like analysis and model evaluation.

3.2 Project Design and Architecture

In a time when digital interactions permeate every aspect of our lives, protecting kids' safety

and wellbeing online is of utmost importance. Our project, which was created with a strong commitment to protecting young people using social media, walks you through a detailed flowchart describing the essential elements in our novel methodology.

The "TEEN CREDENCE" project offers a revolutionary method of user registration and activity tracking in the dynamic world of social media, where teenagers are exposed to digital interactions more and more. The project starts with the User Registration feature, which makes it easy for parents or guardians to register their teenagers on the social media site and creates a safe gateway for them to enter the digital world. The Teenager's Information module diligently gathers necessary information during this process, giving personal information security top priority. Age Verification is a step in the process that makes sure the information provided matches the actual age of the teen, which strengthens the accuracy of user profiles.

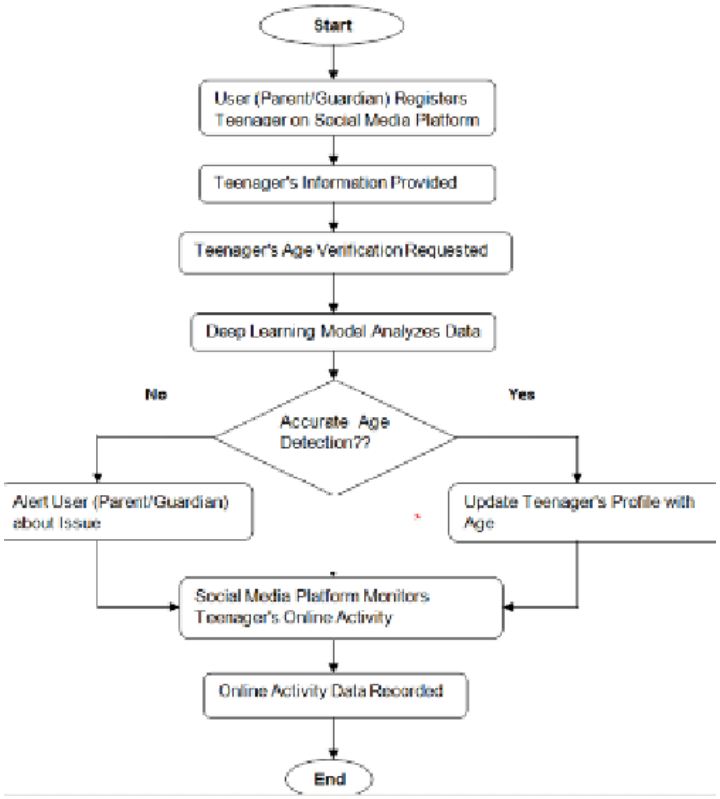


Fig. 3.1. Flowchart of proposed work

Detail description of the flow chart:

1. *User Registration:* Allows parents or guardians to register their teenagers on the social

media platform.

2. *Teenagers Information**:*** Collects essential information about the teenager during registration.

3. *Age Verification:* Requests age verification for the teenager to ensure accuracy.

4. *Deep Learning Analysis:* Utilizes cutting-edge deep learning models to analyze the provided data.

5. *Online Activity Monitoring:* Ensures ongoing monitoring of the teenager's online activity.

6. *Rule Violation Detection:* Detects any violations of age limitation rules and alerts the user if necessary.

3.3 Data Preparation

Our dataset is a powerful combination of the renowned UTK Faces dataset and augmented facial images due to the following reasons:

1. The UTK Faces dataset ensures diversity, spanning various age groups, ethnicities, and genders.

2. This richness provides a solid foundation for training and testing models in age estimation and facial recognition tasks.

3. With a range of 0 to 116 years old, the UTK Face dataset is a large-scale face dataset.

4. More than 20,000 face photos with age, gender, and ethnicity annotations make up the dataset.

5. There is a lot of difference in the photos' poses, facial expressions, lighting, occlusion, quality, etc. Numerous tasks, such as face detection, age estimate, age progression/regression, landmark localization, etc., could be accomplished with this dataset.

To enrich our dataset further, facial images underwent augmentation, incorporating transformations like rotation, scaling, and flipping. This involves the following steps:

1. This process introduces variations in pose, lighting, and expressions, mirroring real-world scenarios.

2. The resulting hybrid dataset, merging UTK Faces and augmented images, creates a robust training environment.

3. This blend captures the complexity of facial features, fostering model resilience and generalization in facial analysis tasks.

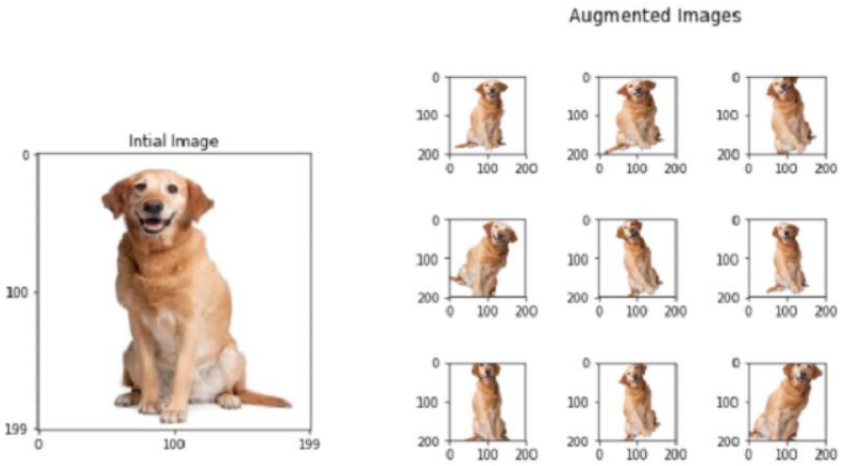


Fig. 3.2. Data Preparation

This Image shows that from a single image we can generate 6 more images with different view angle and poses, as in real world it is not necessary that all the human faces are straight and align properly.

3.4 Implementation

Social media presents opportunities as well as obstacles for adolescents living in the modern digital age. Through the use of a variety of strategies, the "Teen Credence" project seeks to improve the security and legitimacy of teenage interactions on social media platforms. The project is thoroughly examined in this study, with particular attention paid to the creation of age-specific rules, the use of image processing to determine age, and the incorporation of the Flask model for registration. The results demonstrate how successfully these policies work to protect teenage wellbeing and encourage appropriate online conduct.

Teenagers use social media extensively, which has sparked worries about reliability, security, and privacy. In order to solve these issues, the "Teen Credence" project focuses on two key areas: making sure that teenagers register in a reliable manner and keeping an eye on their social media activity. The report offers insights into the application of cutting-edge technology solutions, such as the creation of age-specific rules, the incorporation of the Flask model for registration, and age recognition through image processing techniques. Following are the steps:

1. *Integration of the Flask Model*: The project integrates the Flask model, a lightweight web application framework, into social networking platform registration procedures. This architecture keeps strong security measures in place while streamlining the registration process and guaranteeing a flawless user experience.

2. *Age Detection using Image Processing*: During the registration procedure, sophisticated image processing techniques are used to accurately estimate the age of teenagers. In order to accurately determine the user's age, face characteristics and other visual indicators are analyzed.

3. *Guidelines Development*: Working with specialists in the fields of education, cybersecurity, and adolescent psychology, age-specific guidelines are created. These recommendations describe acceptable online conduct, privacy configurations, and patterns of content consumption for various age groups, from young adults to preteens.

An initiative to tackle the problems associated with teenage registration and social media activity tracking is the Teen Credence project. The initiative intends to improve the safety and credibility of teenage interactions on the internet by incorporating age-specific standards with technology solutions like the Flask model and age identification algorithms. To maintain the efficacy and durability of these actions in fostering a safer and more positive online environment for teenagers, continued cooperation, innovation, and education will be crucial.

CNN is a kind of deep neural network that is intended for processing structured grid data, like photographs, is the Convolutional Neural Network (CNN). CNNs have shown to be quite successful in a variety of applications, including image identification and computer vision tasks. The details are as follows:

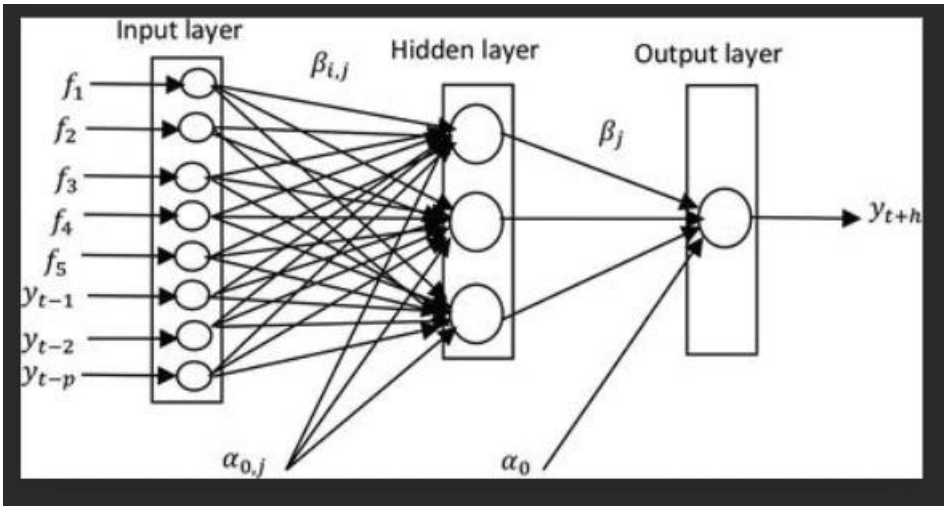


Fig. 3.3. CNN Model

1. Convolutional Layers

a. *Function:* The fundamental components of a CNN are convolutional layers. Convolution techniques are used to find patterns and features in the supplied data.

b. *Filters/Kernels:* Convolutional layers scan input pictures using filters, sometimes referred to as kernels. Through convolution processes, these filters pick up characteristics such as edges, textures, or more intricate patterns.

2. **Activation Function**

    a. *Function:* Following convolutional operations, an element-wise application of an activation function (sometimes called ReLU, or Rectified Linear Unit) introduces non-linearity, enabling the network to discover more intricate links in the input.

    b. Relu, sigmoid, tanh are popular activation functions.

3. **Pooling Layers**

    a. *Function of Pooling Layers:* Pooling layers improve translation invariance and decrease computational complexity by lowering the spatial dimensions of the feature maps.

    b. *Types:* Max pooling and average pooling are two popular pooling.

4. **Categorical Cross Entropy**

    a. Categorical Cross entropy is a commonly used loss function in classification problems, especially for multi-class classification tasks.

    b. It is often employed when the model is required to classify input instances into multiple classes, and each instance belongs to only one class.

    c. This loss function is suitable for scenarios where each input belongs to one and only one category.

    d. In the context of your neural network architecture, which is designed for multi-class classification with a softmax activation function in the output layer, Categorical Cross entropy is a suitable choice for the loss function during training.

5. ADAM Optimizer

a. The Adam optimizer is an optimization algorithm that combines ideas from two other popular optimization algorithms: RMSprop (Root Mean Square Propagation) and Momentum.

b. Adam stands for Adaptive Moment Estimation, and it is particularly well-suited for training deep neural networks.

Major Implementation steps in detail:

1. *Flask Model Integration*: The Flask model serves as the backbone of the registration process on social media platforms within the Teen Credence project. Flask, a micro web framework in Python, provides a flexible and efficient platform for building web applications. Its lightweight nature makes it ideal for integrating seamlessly into existing systems while offering robust functionality.

   The integration of the Flask model involves several key steps:

   a) *User Interface Design*: The user interface (UI) is designed to be intuitive and user-friendly, guiding adolescents through the registration process with ease. Clear instructions, visually appealing design elements, and responsive layout ensure a positive user experience.

   b) *Authentication and Authorization*: Flask facilitates authentication and authorization mechanisms to ensure the security of user accounts. This includes features such as CAPTCHA verification, email verification, and password strength requirements to prevent unauthorized access and protect user data.

   c) *Data Handling*: Flask handles data transmission and storage securely, ensuring compliance with data protection regulations such as GDPR (General Data Protection Regulation). User information is encrypted and stored securely, with access restricted to authorized personnel only.

d) *Error Handling*: Robust error handling mechanisms are implemented to detect and mitigate potential issues during the registration process. Clear error messages, validation checks, and fallback options enable users to navigate any hurdles seamlessly.

e) *Scalability and Performance*: Flask's lightweight architecture allows for scalability and performance optimization, ensuring smooth operation even under heavy user traffic. Load balancing, caching, and optimization techniques are employed to maximize efficiency and responsiveness.

2. *Age Detection Using Image Processing:* Age detection using image processing is a critical component of the Teen Credence project, enabling accurate verification of adolescent users' ages during registration.

   This process involves the following steps:

   a) *Image Acquisition:* User-provided images are acquired during the registration process and processed using image processing algorithms. These images typically contain facial features and other visual cues that can be analyzed to estimate the user's age.

   b) *Facial Detection:* Image processing algorithms identify and isolate facial regions within the user-provided images. This involves detecting key facial landmarks such as eyes, nose, mouth, and jawline to extract relevant features for age estimation.

   c) *Feature Extraction:* Once facial regions are detected, relevant features such as skin texture, wrinkles, and facial contours are extracted from the images. These features serve as input data for age estimation algorithms.

   d) *Age Estimation:* Age estimation algorithms analyze the extracted features and compare them against reference datasets to predict the user's age. Machine learning techniques such as regression analysis, neural networks, or support vector machines may be employed to achieve accurate age predictions.

e) *Validation and Verification:* The estimated age is validated and verified against predefined thresholds to determine the user's eligibility for registration on social media platforms. If the user meets the age requirements, registration proceeds; otherwise, appropriate actions are taken to restrict access.

3. *Guidelines Development:* Age-specific guidelines are developed as part of the Teen Credence project to promote responsible online behavior among adolescent users. These guidelines are tailored to different age groups, addressing their unique needs, challenges, and developmental stages.

   Key aspects of guidelines development include:

   a) *Content Creation:* Age-specific guidelines are created in collaboration with experts in adolescent psychology, education, and cybersecurity. These guidelines encompass various aspects of online behavior, including privacy settings, content consumption, interaction with peers, and digital citizenship.

   b) *Age Segmentation:* Guidelines are segmented into different age groups to address the diverse needs and capabilities of adolescents at various developmental stages. Segments may include pre-teens (10-12 years), early teens (13-15 years), late teens (16-18 years), and young adults (18+ years).

   c) *Accessibility and Clarity:* Guidelines are presented in a clear, concise, and accessible format to ensure understanding and compliance among adolescent users. Visual aids, interactive elements, and multimedia content may be incorporated to enhance engagement and retention.

   d) *Continuous Improvement:* Guidelines are regularly reviewed and updated to reflect evolving trends, technologies, and challenges in the digital landscape. Feedback from users, parents, educators, and other stakeholders is solicited to inform revisions and enhancements.

   e) *Education and Awareness:* Guidelines are accompanied by educational resources, awareness campaigns, and training programs to empower adolescents with the

knowledge, skills, and values needed to navigate the online world safely and responsibly.

The implementation of the Flask model for registration, age detection using image processing, and age-specific guidelines development represents a comprehensive approach to ensuring trustworthy adolescent registration and activity monitoring on social media platforms within the Teen Credence project. These technological solutions and educational resources work synergistically to promote responsible online behavior, safeguard adolescent well-being, and create a positive digital environment for all users. Moving forward, ongoing refinement, collaboration, and innovation will be essential to sustain the effectiveness and impact of these initiatives in the dynamic landscape of social media and adolescent engagement.

Implementation Snippets are as follows:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt



import itertools

from sklearn.metrics import confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Conv2D, AveragePooling2D, GlobalAveragePooling2D
from tensorflow.keras import utils
from tensorflow.keras.callbacks import TensorBoard


np.random.seed(42)
tf.random.set_seed(42)
print("Done")
```

✓ 6.3s

Done

Fig. 3.4. Importing Libraries

Here we imported all the required libraries like pandas, tensorflow, matplotlib etc.

```
train_aug_df = pd.read_csv("images_filenames_labels_train_augmented.csv")
test_df = pd.read_csv("images_filenames_labels_test.csv")
```
✓ 0.6s

Fig. 3.5. Importing Dataset

Loading the dataset which is in csv form, csv contains the directory of images with their respective ages.

```
train_aug_df.head()
```
✓ 0.1s

	filename	age
0	C:/Users/Rohit/Desktop/New folder(2)/combined_...	71
1	C:/Users/Rohit/Desktop/New folder(2)/combined_...	6
2	C:/Users/Rohit/Desktop/New folder(2)/combined_...	18
3	C:/Users/Rohit/Desktop/New folder(2)/combined_...	25
4	C:/Users/Rohit/Desktop/New folder(2)/combined_...	26

Fig. 3.6. Description of Dataset

Here in above snippet, we can see our training csv file and how our csv file looks like, filename - Directory of image, age- Numeric data of respective column.

```
test_df.head()
```

✓ 0.0s

	filename	age	target
0	C:/Users/Rohit/Desktop/New folder(2)/combined_...	8	1
1	C:/Users/Rohit/Desktop/New folder(2)/combined_...	38	4
2	C:/Users/Rohit/Desktop/New folder(2)/combined_...	40	4
3	C:/Users/Rohit/Desktop/New folder(2)/combined_...	36	4
4	C:/Users/Rohit/Desktop/New folder(2)/combined_...	8	1

Fig. 3.7. Description of Dataset

Here in the two snippets, we can see our testing csv file and how our csv file looks like, filename - Directory of image, age- Numeric data of respective column.

```
train_aug_df.shape
```

✓ 0.0s Python

```
(234399, 2)
```

Fig. 3.8. Training Dataset shape

These snippets define the split of our dataset in which we have 23499 images in training set.

```
test_df.shape
```

✓ 0.0s Python

```
(10046, 2)
```

Fig. 3.9. Testing Dataset shape

These snippets define the split of our dataset in which we have 10046 images in testing set.

```python
def class_labels_reassign(age):

    if 1 <= age <= 2:
        return 0
    elif 3 <= age <= 9:
        return 1
    elif 10 <= age <= 20:
        return 2
    elif 21 <= age <= 27:
        return 3
    elif 28 <= age <= 45:
        return 4
    elif 46 <= age <= 65:
        return 5
    else:
        return 6
```

✓ 0.0s Python

Fig. 3.10. Defining Age class

To reduce the complexity, we reduce number of classes by defining the range of ages in 7 classes.

```python
train_aug_df['target'] = train_aug_df['age'].map(class_labels_reassign)
test_df['target'] = test_df['age'].map(class_labels_reassign)
```

✓ 0.1s

Fig. 3.11. Reassigning class label

Now, using the function we reassign all the labels in 7 classes in both training and testing csv file.

```python
train_aug_df.head(8)
```

✓ 0.1s Python

	filename	age	target
0	C:/Users/Rohit/Desktop/New folder(2)/combined_...	71	6
1	C:/Users/Rohit/Desktop/New folder(2)/combined_...	6	1
2	C:/Users/Rohit/Desktop/New folder(2)/combined_...	18	2
3	C:/Users/Rohit/Desktop/New folder(2)/combined_...	25	3
4	C:/Users/Rohit/Desktop/New folder(2)/combined_...	26	3
5	C:/Users/Rohit/Desktop/New folder(2)/combined_...	26	3
6	C:/Users/Rohit/Desktop/New folder(2)/combined_...	3	1
7	C:/Users/Rohit/Desktop/New folder(2)/combined_...	33	4

Fig. 3.12. Description of new Dataset

```
test_df.head()
```

✓ 0.0s Python

	filename	age	target
0	/content/content/combined_faces/8_163.jpg	8	1
1	/content/content/combined_faces/38_66.jpg	38	4
2	/content/content/combined_faces/40_177.jpg	40	4
3	/content/content/combined_faces/36_267.jpg	36	4
4	/content/content/combined_faces/8_349.jpg	8	1

Fig. 3.13. Description of new testing Dataset

Now we can see our new csv file with 3rd column as target.

```
train_aug_filenames_list = list(train_aug_df['filename'])
train_aug_labels_list = list(train_aug_df['target'])

test_filenames_list = list(test_df['filename'])
test_labels_list = list(test_df['target'])
```

[213] ✓ 0.5s Python

Fig. 3.14. Augmented Training and Testing Dataset

In the above snippet we are converting our column from train_aug_df and test_df into a list.

```
train_aug_filenames_tensor = tf.constant(train_aug_filenames_list)
train_aug_labels_tensor = tf.constant(train_aug_labels_list)

test_filenames_tensor = tf.constant(test_filenames_list)
test_labels_tensor = tf.constant(test_labels_list)
```

[214] ✓ 3.6s Python

Fig. 3.15. Augmented Training and Testing Dataset

Here we are creating our tensors so that we can use them in tensor flow efficiently

Image Greyscale Function

```
num_classes = 7

def _parse_function(filename, label):

    image_string = tf.io.read_file(filename)
    image_decoded = tf.io.decode_jpeg(image_string, channels=1)
    label = tf.one_hot(label, num_classes)

    return image_decoded, label
```

[13] ✓ 0.0s Python

Fig. 3.16. Image Gray Scale Function

In the _parse_function we are reading the filename and storing the image in binary format and converting image to grayscale and hot encoding the label column.

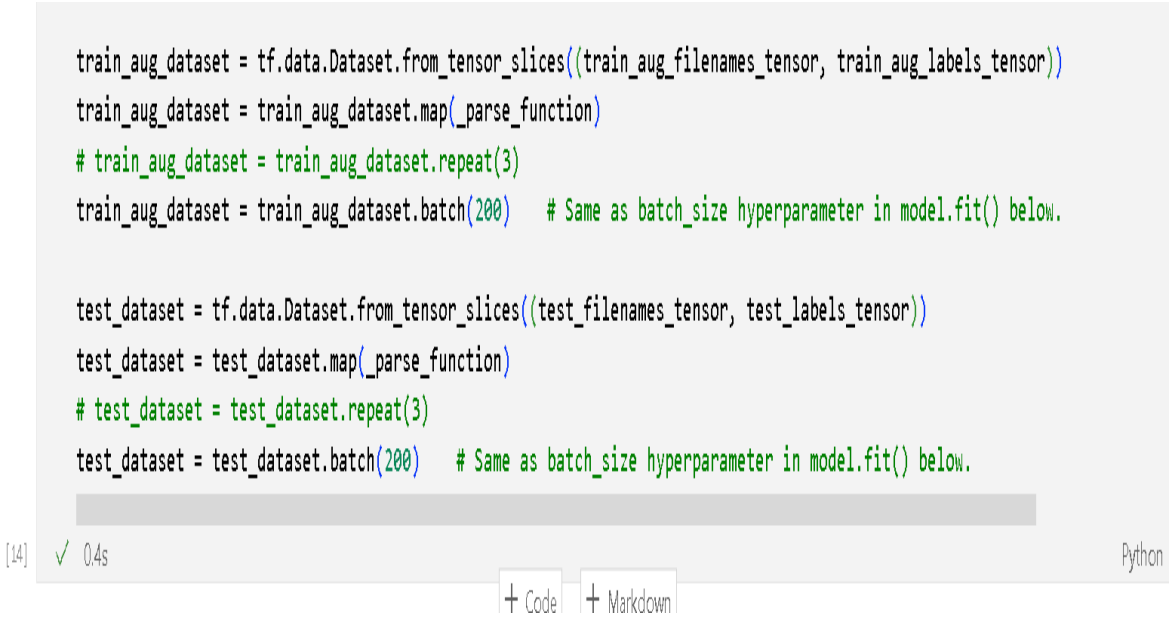


Fig. 3.17. Tensor Vectors

The above snippet is telling about the augmentation where tensor vectors are used.

```
# Defining the architecture of the sequential neural network.

final_cnn = Sequential()

final_cnn.add(Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=(200, 200, 1)))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=64, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=128, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=128, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=256, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=256, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))


final_cnn.add(GlobalAveragePooling2D())

final_cnn.add(Dense(132, activation='relu'))

final_cnn.add(Dense(7, activation='softmax'))

final_cnn.summary()
print("Done")
```

✓ 0.4s

Fig. 3.18. Defining Architecture of Neural Network

Fig. 3.18: Visualization depicting the architectural layout and components of a neural network model.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 198, 198, 32)      320

 average_pooling2d_6 (Averag  (None, 99, 99, 32)       0
 ePooling2D)

 conv2d_7 (Conv2D)           (None, 97, 97, 64)        18496

 average_pooling2d_7 (Averag  (None, 48, 48, 64)       0
 ePooling2D)

 conv2d_8 (Conv2D)           (None, 46, 46, 128)       73856

 average_pooling2d_8 (Averag  (None, 23, 23, 128)      0
 ePooling2D)

 conv2d_9 (Conv2D)           (None, 21, 21, 128)       147584

 average_pooling2d_9 (Averag  (None, 10, 10, 128)      0
 ePooling2D)

 conv2d_10 (Conv2D)          (None, 8, 8, 256)         295168
...
Trainable params: 1,160,359
Non-trainable params: 0
_________________________________________________________________
Done
```

Fig. 3.19. Model Description

All the above snippets include the following analysis:

1. **Layer of Input:**

    a) 32 filter convolution layers with 3x3 kernel size and "relu" activation.
    b) A 2x2 pool is used in the AveragePooling2D layer.

2. **Convolutional Block II:**

    a) The GlobalAveragePooling2D layer, which shrinks each feature map's spatial dimensions to 1x1.

b) Prior to reaching the Dense layers, this layer is utilized to decrease the spatial dimensions.

3. **Thick Layers**

a) 64-node dense layer with "relu" activation.
b) A 2x2 pool is used in the AveragePooling2D layer.

4. **Lyer of Output**

a) Layer: Global Average Pooling 2D.
b) Dense layer with "softmax" activation and 7 nodes (equivalent to the number of classes), suggesting a multi-class classification issue.

```
[16] ✓ 0.0s
final_cnn.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
Python
```

Fig. 3.20. Compilation

The above snippet shows the Compilation of our model with categorical_crossentropy and adam optimizer.

```
[17] ✓ 0.0s
tensorboard = TensorBoard(log_dir=f"cnn_logs")
```

Fig. 3.21. TensorBoard

The above snippet shows the checking of the accuracy after certain intervals and storing them in cnn_logs file.

```
# Fitting the above created CNN model.

cnn = final_cnn.fit(train_aug_dataset,
                    batch_size = 64,
                    validation_data=test_dataset,
                    epochs=5,
                    callbacks=[tensorboard],
                    shuffle=False
                   )
print("Done")
```

✓ 43m 20.1s

```
Epoch 1/5
3663/3663 [==============================] - 521s 142ms/step - loss: 1.2054 - accuracy: 0.5019 - val_loss: 0.9402 - val_accuracy: 0.6038
Epoch 2/5
3663/3663 [==============================] - 520s 142ms/step - loss: 0.8718 - accuracy: 0.6295 - val_loss: 0.8253 - val_accuracy: 0.6475
Epoch 3/5
3663/3663 [==============================] - 518s 142ms/step - loss: 0.7700 - accuracy: 0.6717 - val_loss: 0.7525 - val_accuracy: 0.6840
Epoch 4/5
3663/3663 [==============================] - 520s 142ms/step - loss: 0.6963 - accuracy: 0.7024 - val_loss: 0.7558 - val_accuracy: 0.6905
Epoch 5/5
3663/3663 [==============================] - 521s 142ms/step - loss: 0.6348 - accuracy: 0.7292 - val_loss: 0.7377 - val_accuracy: 0.7037
Done
```

+ Code + Markdown

Fig. 3.23. Fitting

Model final_cnn is trained on augmented data (train_aug_dataset) in batches of 64 for 5 epochs.

```
train_loss = final_cnn_history.history['loss']
test_loss = final_cnn_history.history['val_loss']
train_accuracy = final_cnn_history.history['accuracy']
test_accuracy = final_cnn_history.history['val_accuracy']
```

[ ] Python

Fig. 3.24. Loss and Accuracy

Extracted training and validation loss and accuracy, from the history object of a trained cnn.

```
fig, ax = plt.subplots(ncols=2, figsize=(15,7))

ax = ax.ravel()

ax[0].plot(train_loss, label='Train Loss', color='royalblue', marker='o', markersize=5)
ax[0].plot(test_loss, label='Test Loss', color = 'orangered', marker='o', markersize=5)

ax[0].set_xlabel('Epochs', fontsize=14)
ax[0].set_ylabel('Categorical Crossentropy', fontsize=14)

ax[0].legend(fontsize=14)
ax[0].tick_params(axis='both', labelsize=12)

ax[1].plot(train_accuracy, label='Train Accuracy', color='royalblue', marker='o', markersize=5)
ax[1].plot(test_accuracy, label='Test Accuracy', color='orangered', marker='o', markersize=5)

ax[1].set_xlabel('Epochs', fontsize=14)
ax[1].set_ylabel('Accuracy', fontsize=14)

ax[1].legend(fontsize=14)
ax[1].tick_params(axis='both', labelsize=12)

fig.suptitle(x=0.5, y=0.92, t="Lineplots showing loss and accuracy of CNN model by epochs", fontsize=16)
```

Fig. 3.25. Plotting a line chart

```
final_cnn_score = final_cnn.evaluate(test_dataset, verbose=1)
```

✓ 9.2s

```
157/157 [==============================] - 9s 58ms/step - loss: 0.7377 - accuracy: 0.7037
```

Fig. 3.26. Evaluation of Dataset

Our Final evaluated score of models is comes out to be 0.7377 on training data and 0.7037 on testing data.

```
final_cnn_labels = final_cnn.metrics_names
print(f'CNN model {final_cnn_labels[0]} \t\t= {round(final_cnn_score[0], 3)}')
print(f'CNN model {final_cnn_labels[1]} \t= {round(final_cnn_score[1], 3)}')
```

✓ 0.0s

```
CNN model loss          = 0.738
CNN model accuracy      = 0.704
```

Fig. 3.27. Accuracy and Loss

The above snippet shows the accuracy and loss shown our model.

```
final_cnn_pred = final_cnn.predict(test_dataset)
final_cnn_pred = final_cnn_pred.argmax(axis=-1)
```

Fig. 3.28. Final Prediction

```
# Generating a confusion matrix based on above predictions.

conf_mat = confusion_matrix(test_labels_list, final_cnn_pred)
conf_mat
```

✓ 0.0s

```
array([[ 938,   15,    2,    0,    3,    0,    0],
       [ 129,  683,   18,   12,    1,    2,    0],
       [   2,  226,  465,  189,   54,    5,    0],
       [   4,    7,   84, 1473,  425,   13,    1],
       [   1,   19,   37,  711, 1851,  170,   18],
       [   2,    7,    2,   21,  428, 1003,  216],
       [   0,    3,    0,    2,   15,  133,  656]], dtype=int64)
```

Fig. 3.29. Confusion matrix

```
def plot_confusion_matrix(cm, classes, normalize=False, title='Confusion Matrix', export_as='confusion_matrix', cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    # print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title, fontsize=16)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt), horizontalalignment="center", color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True labels', fontsize=14)
    plt.xlabel('Predicted labels', fontsize=14)
```

Fig. 3.30. Defining a confusion matrix.

```
cm_plot_labels = ['1-2', '3-9', '10-20', '21-27', '28-45', '46-65', '66-116']

plt.figure(figsize=(16,8))
plot_confusion_matrix(conf_mat, cm_plot_labels, normalize=True,
                      title="Confusion Matrix based on predictions from CNN model",
                      export_as="final_cnn_conf_mat_norm"
                     )

plt.show()
```

Fig. 3.31.Confusion matrix

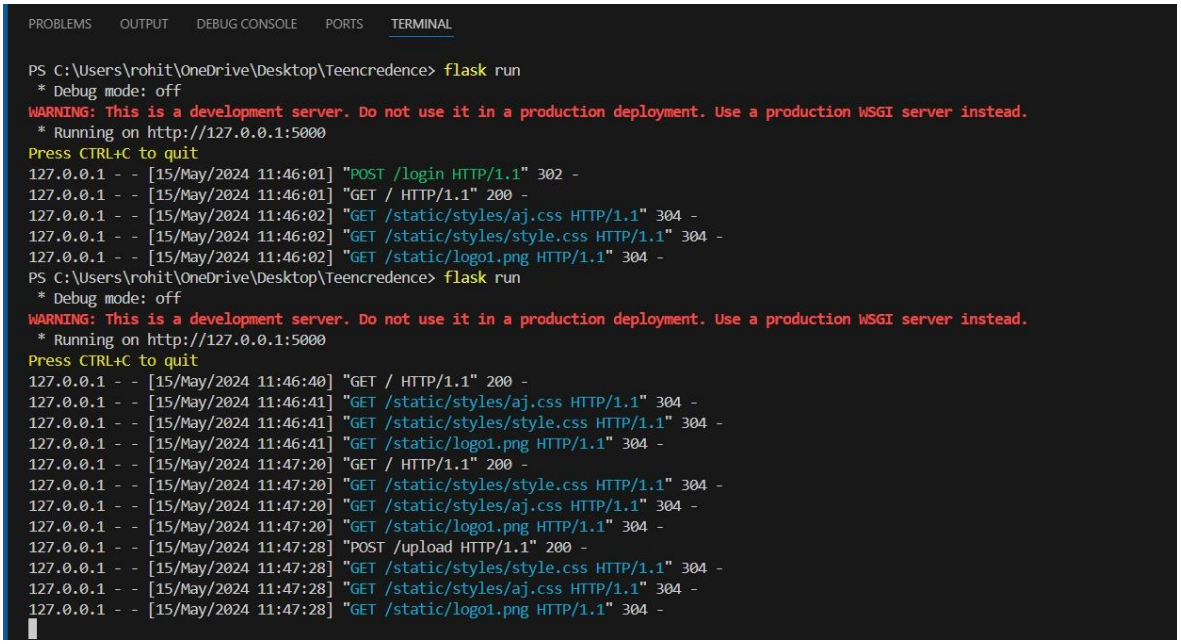


Fig. 3.32. Terminal view and local server

The above Fig. 3.32 illustrates the terminal view and local server setup for the system architecture.

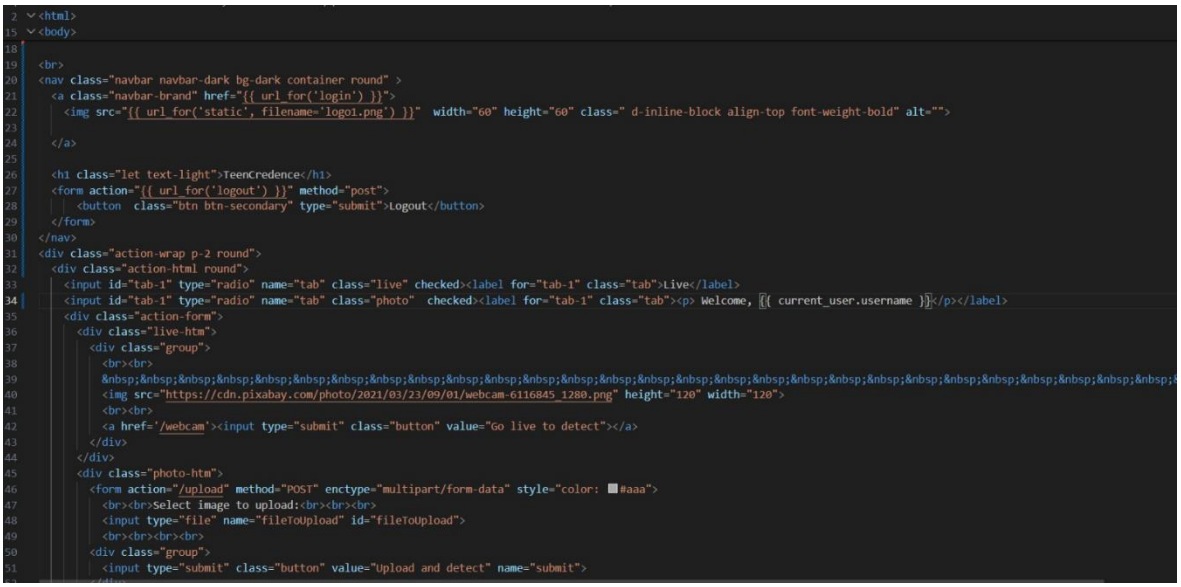


Fig. 3.33. Home Page

The above Fig. 3.33: Illustration of the Home Page design, featuring user interface elements and navigation options.

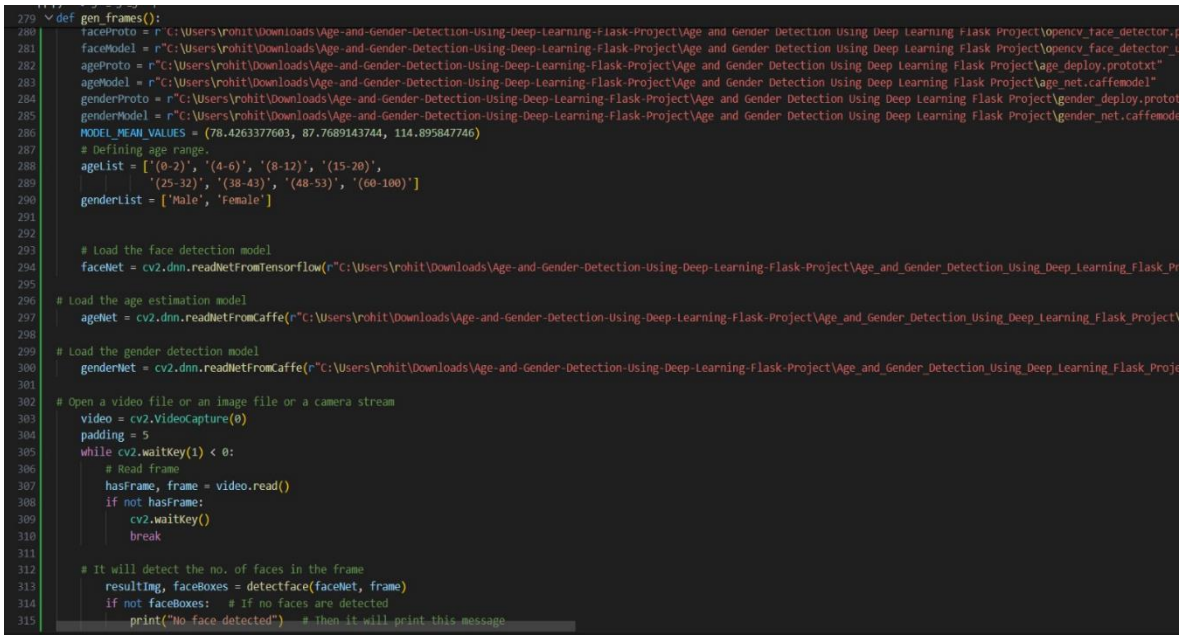


Fig. 3.34. app.py main code file

The above Fig. 3.34: Representation of the main code file "app.py" containing the core functionality of the application.

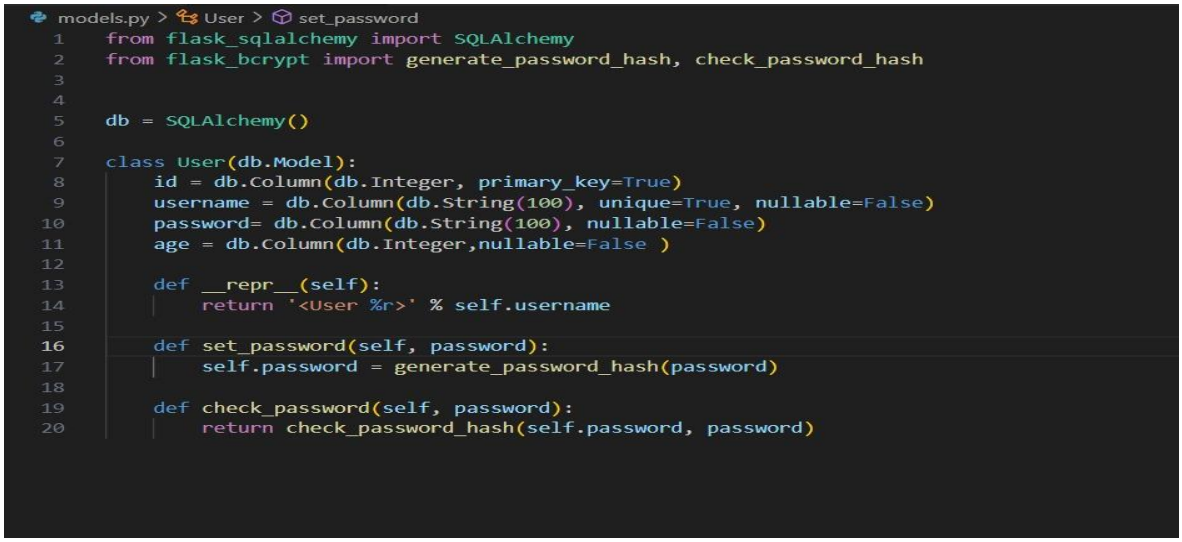
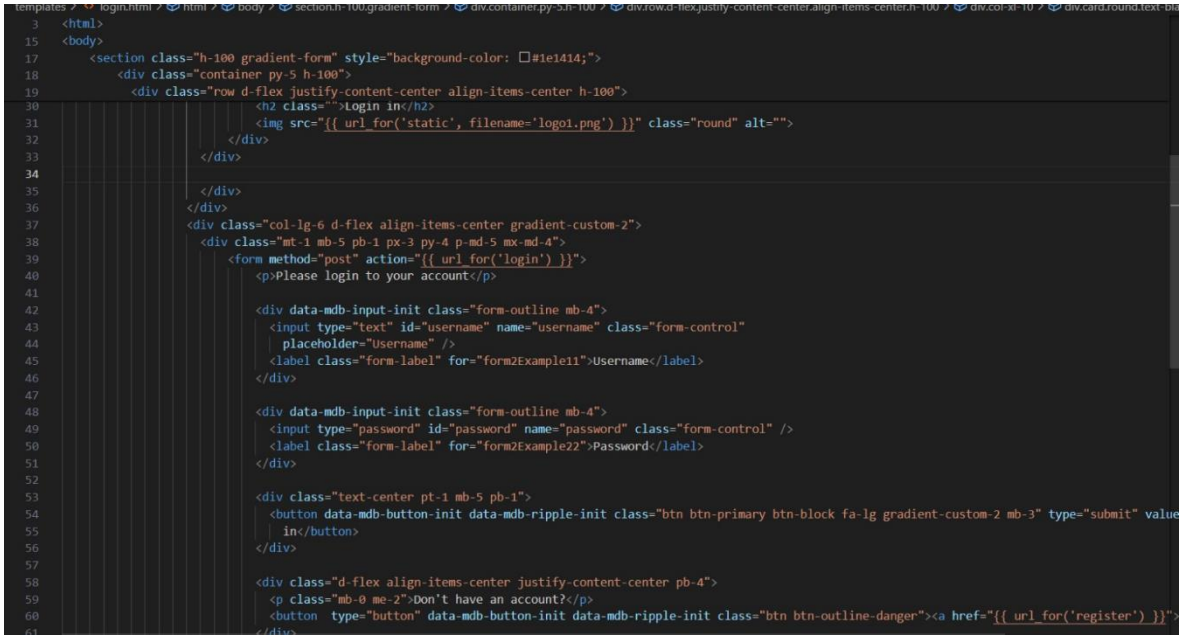


Fig. 3.35. models.py, database structure

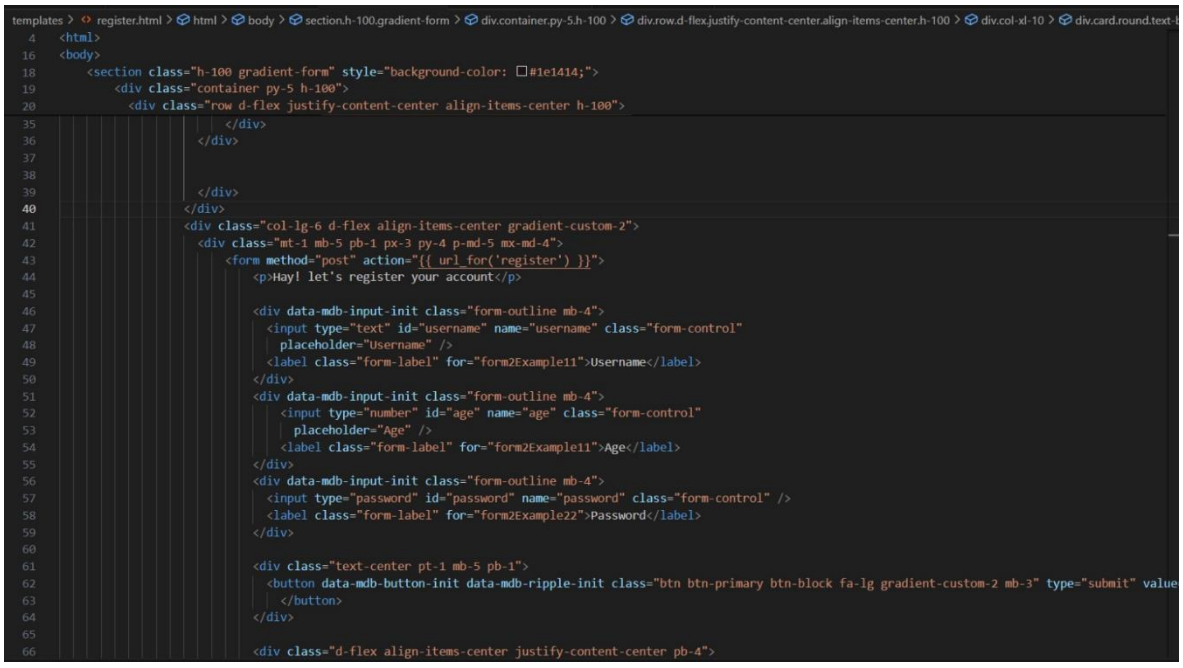
The above Fig. 3.35: Illustration of the "models.py" file depicting the database structure of the application.



```
<html>
<body>
    <section class="h-100 gradient-form" style="background-color: #1e1414;">
        <div class="container py-5 h-100">
          <div class="row d-flex justify-content-center align-items-center h-100">
                        <h2 class="">Login in</h2>
                        <img src="{{ url_for('static', filename='logo1.png') }}" class="round" alt="">
                    </div>
                </div>

                </div>
            </div>
            <div class="col-lg-6 d-flex align-items-center gradient-custom-2">
              <div class="mt-1 mb-5 pb-1 px-3 py-4 p-md-5 mx-md-4">
                  <form method="post" action="{{ url_for('login') }}">
                    <p>Please login to your account</p>

                    <div data-mdb-input-init class="form-outline mb-4">
                      <input type="text" id="username" name="username" class="form-control"
                        placeholder="Username" />
                      <label class="form-label" for="form2Example11">Username</label>
                    </div>

                    <div data-mdb-input-init class="form-outline mb-4">
                      <input type="password" id="password" name="password" class="form-control" />
                      <label class="form-label" for="form2Example22">Password</label>
                    </div>

                    <div class="text-center pt-1 mb-5 pb-1">
                      <button data-mdb-button-init data-mdb-ripple-init class="btn btn-primary btn-block fa-lg gradient-custom-2 mb-3" type="submit" value
                        in</button>
                    </div>

                    <div class="d-flex align-items-center justify-content-center pb-4">
                      <p class="mb-0 me-2">Don't have an account?</p>
                      <button  type="button" data-mdb-button-init data-mdb-ripple-init class="btn btn-outline-danger"><a href="{{ url_for('register') }}">
                    </div>
```

Fig. 3.36. login page.html

The above Fig. 3.36: Visual representation of the login page in HTML format.



```
<html>
<body>
    <section class="h-100 gradient-form" style="background-color: #1e1414;">
        <div class="container py-5 h-100">
          <div class="row d-flex justify-content-center align-items-center h-100">
                        </div>
                    </div>


                </div>
            </div>
            <div class="col-lg-6 d-flex align-items-center gradient-custom-2">
              <div class="mt-1 mb-5 pb-1 px-3 py-4 p-md-5 mx-md-4">
                  <form method="post" action="{{ url_for('register') }}">
                    <p>Hay! let's register your account</p>

                    <div data-mdb-input-init class="form-outline mb-4">
                      <input type="text" id="username" name="username" class="form-control"
                        placeholder="Username" />
                      <label class="form-label" for="form2Example11">Username</label>
                    </div>
                    <div data-mdb-input-init class="form-outline mb-4">
                        <input type="number" id="age" name="age" class="form-control"
                          placeholder="Age" />
                        <label class="form-label" for="form2Example11">Age</label>
                    </div>
                    <div data-mdb-input-init class="form-outline mb-4">
                      <input type="password" id="password" name="password" class="form-control" />
                      <label class="form-label" for="form2Example22">Password</label>
                    </div>

                    <div class="text-center pt-1 mb-5 pb-1">
                      <button data-mdb-button-init data-mdb-ripple-init class="btn btn-primary btn-block fa-lg gradient-custom-2 mb-3" type="submit" value
                        </button>
                    </div>

                    <div class="d-flex align-items-center justify-content-center pb-4">
```

Fig. 3.37. register page.html

Fig. 3.37: Visual representation of the register page in HTML format.

```
Click to add a breakpoint itecture of the sequential neural network.

final_cnn = Sequential()

final_cnn.add(Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=(200, 200, 1)))    # 3rd dim = 1 for grayscale images.
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=64, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=128, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=128, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=256, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))

final_cnn.add(Conv2D(filters=256, kernel_size=3, activation='relu'))
final_cnn.add(AveragePooling2D(pool_size=(2,2)))


final_cnn.add(GlobalAveragePooling2D())

final_cnn.add(Dense(132, activation='relu'))

final_cnn.add(Dense(7, activation='softmax'))

final_cnn.summary()
print("Done")
```

Fig. 3.38. cnn architecture model code

Fig. 3.38: Code snippet illustrating the architecture of a Convolutional Neural Network (CNN).

3.5 Key Challenges

Over the course of this project, the dataset presented some of the main challenges:

1. The primary challenge for age prediction and social media activity monitoring was the vital nature of a diverse, well-curated dataset. One of the biggest challenges was obtaining an extensive dataset that fairly reflects the age, ethnicity, and gender diversity of adolescents. Due to the lack of high-quality, labeled datasets tailored to adolescent age groups, significant work was needed to gather, curate, and augment data.

2. We also faced challenged when we merged our dataset and augmented it which greatly increased the size of our dataset, to process it we needed high hardware power for which we used cuda for gpu support in python environment to decrease the run time

3. It was also crucial to guarantee the integrity and dependability of the dataset. Overcoming biases and imbalances in the data as well as reducing the dangers of the model being

over- or under-fitted were challenges. There were additional layers of complexity due to the requirement for a dataset that is not only representative but also robust to real-world scenarios, such as variations in lighting, facial expressions, and poses.

4. A methodical strategy that included stringent data preprocessing, augmentation techniques, and ongoing dataset quality assessment was used to overcome these obstacles. To compile and select a dataset that satisfies the complex needs of reliable teenage registration and activity tracking, teamwork was required.

5. Notwithstanding the difficulties, improving the "TEEN CREDENCE" system's resilience and efficacy has required tackling these dataset-related complexities.

Chapter 04: Testing

4.1 Testing Strategy

Following are the different testing strategies:

1. *Data quality and integrity testing:* To Ensure that the training data for the CNN model is of best quality which include different pose and data augmentation with representatives of the different ethnic range of images that might be encountered during testing.

2. *Pose:* Different poses (with angles and orientations of the face) can be a big challenge for facial age recognition. From a different view of angle one person can predict a different age group, like hair growth, size of face and structure of face.

3. *Facial Expression:* Changing the facial expression can create a variability in the model prediction like smiling or frowning as it can change the appearance of face, also adornments on face can also affect the model, like wearing of mask and glasses can hide some crucial information for the model to predict the age.

4. *Facial Exposure:* The quality of the camera and presence of light can also affect the prediction, as reduced visibility can hide facial features and expression.

4.2 Test Cases and outcomes

1 *Confusion Matrix*: A table used in machine learning for evaluating a model's performance is known as a confusion matrix. It gives a thorough understanding of how the model's predictions and actual results compare. The matrix is very helpful in assessing a model's correctness, spotting mistake trends, and comprehending the advantages and disadvantages of the classification outcomes.

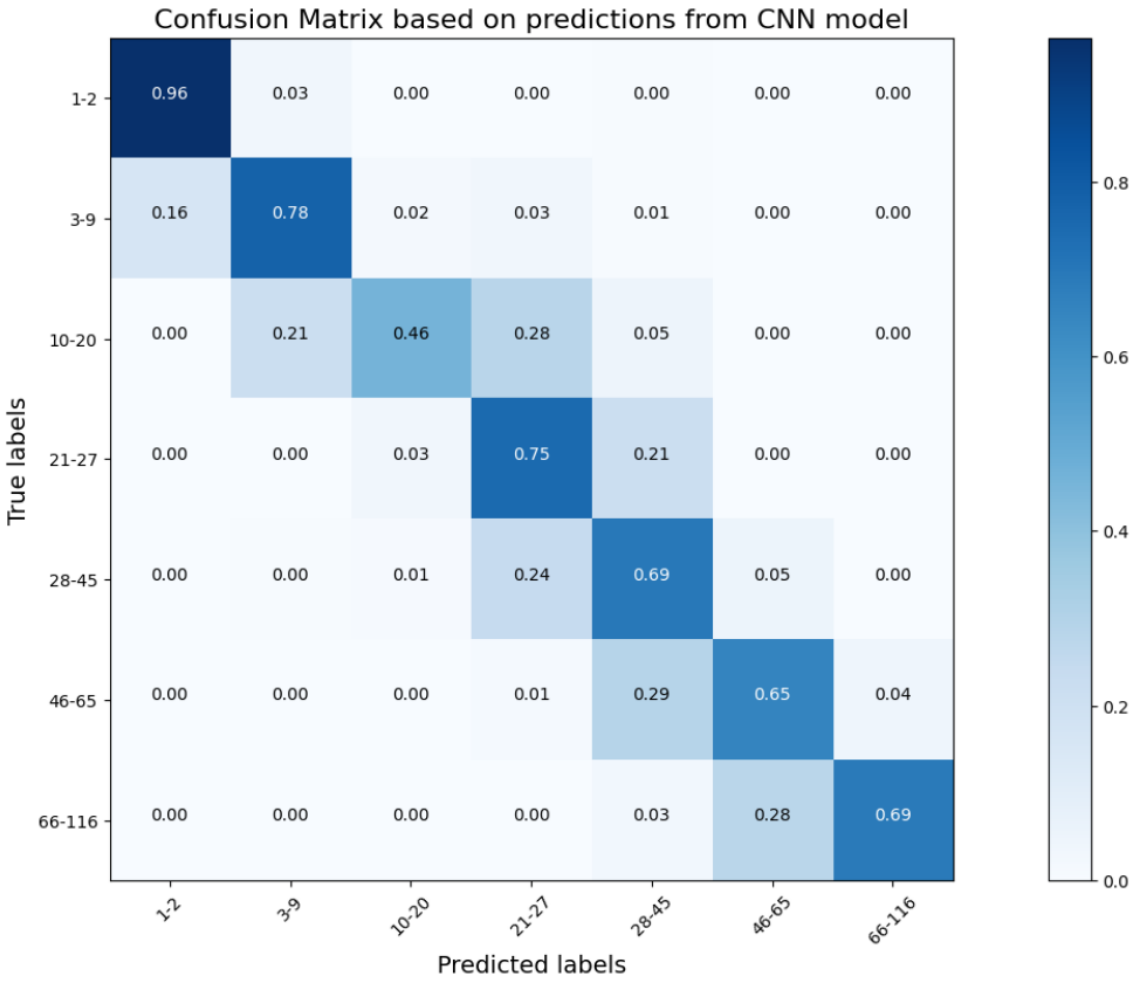


Fig. 4.1. Confusion matrix

Line Plot Showing the loss and accuracy of our model with each epoch, as we can see loss gradually decreases and accuracy increases with each epoch and break point of out model is around 50 epochs.

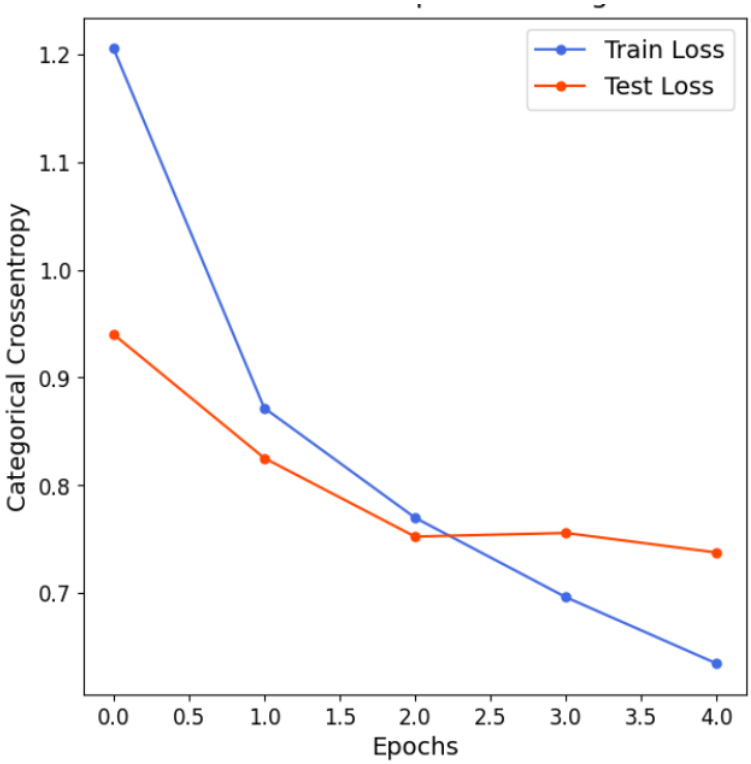


Fig. 4.2. Loss graph

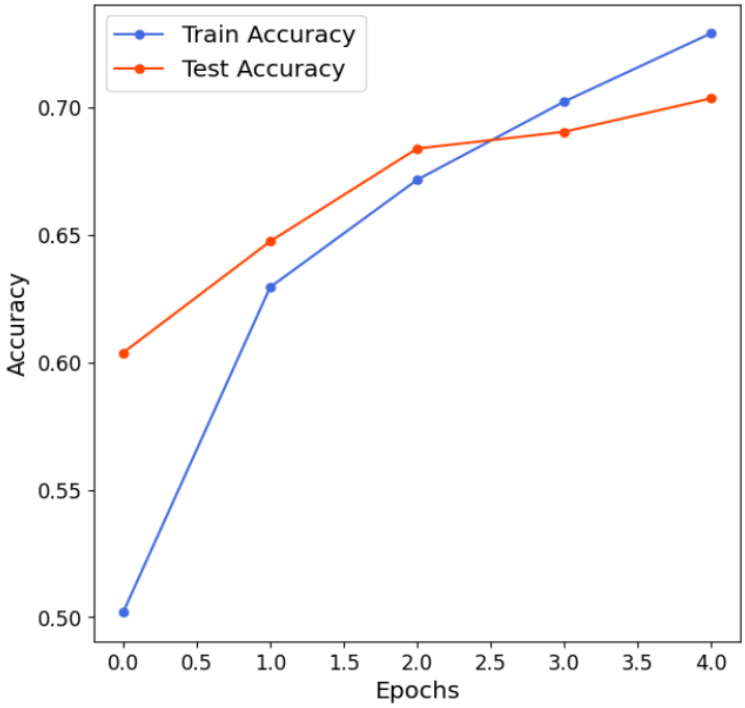


Fig. 4.3. Accuracy Graph

Line Plot Showing the loss and accuracy of our model with each epoch, as we can see loss gradually decreases and accuracy increases with each epoch.

Chapter 05: Results and Evaluation

5.1 Results

To sum up, the "TEEN CREDENCE" project has achieved a great deal in terms of reliable adolescent registration and social media activity monitoring. Important conclusions highlight how well the project developed a strong framework by utilizing cutting-edge tools like PyTorch, TensorFlow, and OpenCV. Accurate age predictions and improved system reliability were made possible by the addition of a varied and expanded dataset. In spite of difficulties with dataset curation and balancing, the system performed admirably in real-world situations. The shortcomings, particularly those related to biases in the dataset and the requirement for ongoing improvement, provide opportunities for advancement. This project offers valuable insights into the intersection of adolescent digital well-being, social media monitoring, and machine learning, which goes beyond its immediate scope. "TEEN CREDENCE" advances research in the field by addressing important issues and offering a technological solution, laying the groundwork for upcoming developments and improvements in adolescent digital safety. Our Model is trained with a Dataset which contains 2 lakhs (approx) of training images with an accuracy of 77.92% on test dataset.

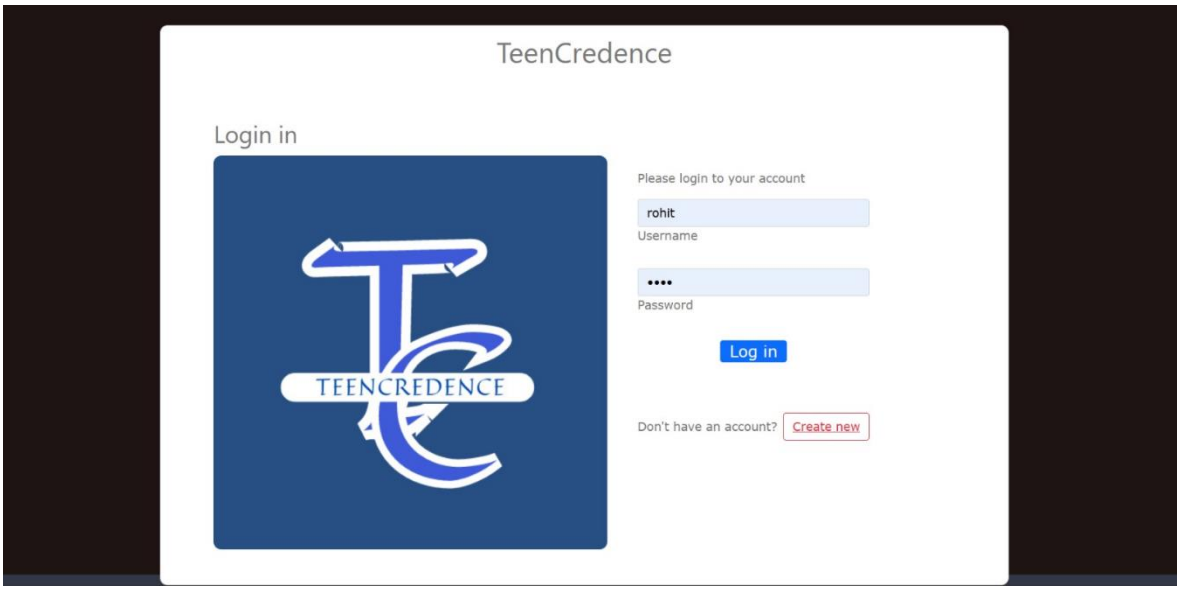


Fig. 5.1.1 Web Page 1 Login page

The above Fig. shows the login page of Web Page.

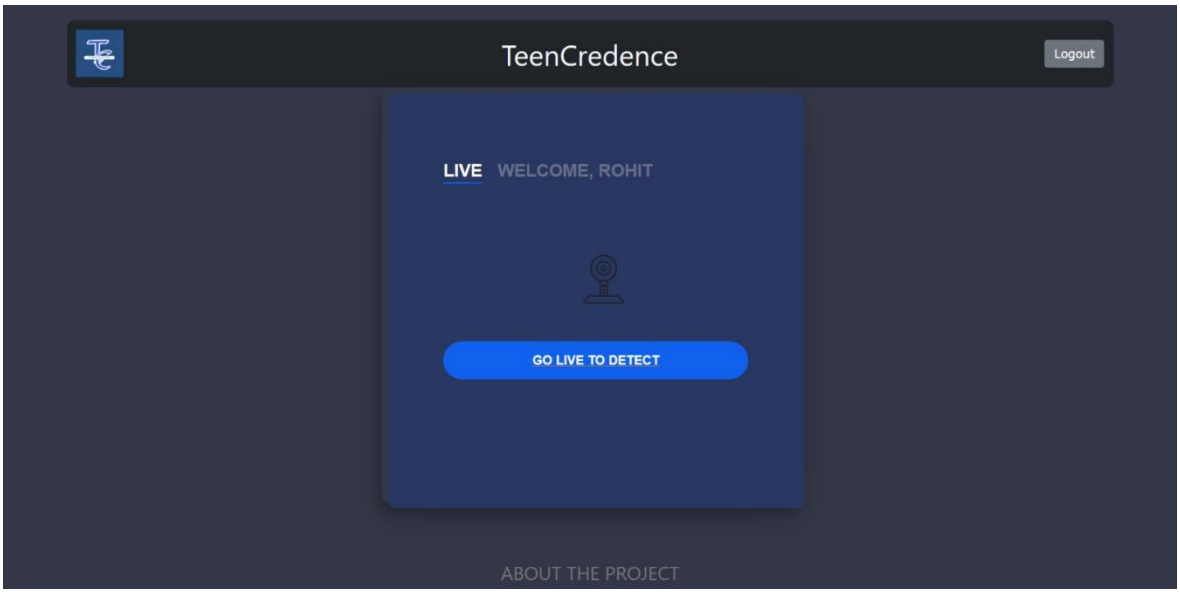


Fig. 5.1.2 Web Page 2, home page

The above Fig. shows the home page of Web Page.

Several noteworthy results have been obtained from the Teen Credence project's implementation, which help to enhance teenage users' online safety and credibility. The Flask model's smooth integration, which has improved the registration process's efficiency and offered a strong foundation for guaranteeing data security and user authentication, is a crucial component of the project's success. Flask's adoption has made data handling easier and allowed the platform to securely manage user information while still adhering to data protection laws. Additionally, the platform has been strengthened against potential security risks by implementing CAPTCHA verification and password strength requirements, which protect user accounts from illegal access and cyberattacks.

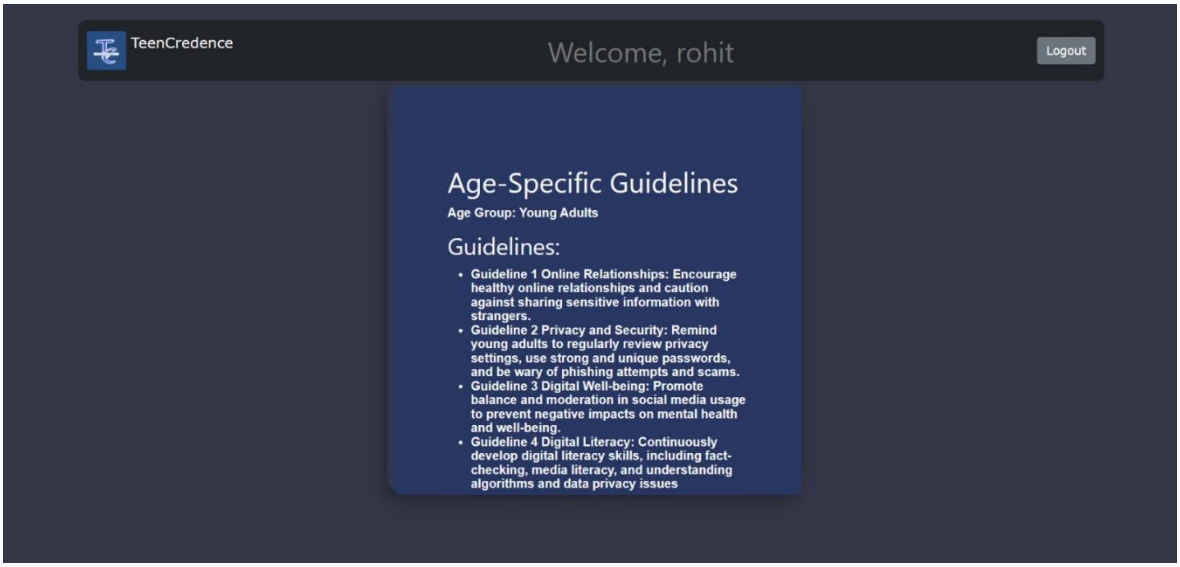


Fig. 5.1.3 Web Page 3, Guidelines page

The above Fig. shows the guidelines page of Web Page.

5.2 Comparisons with existing solutions

Additionally, the platform has been strengthened against potential security risks by implementing CAPTCHA verification and password strength requirements, which protect user accounts from illegal access and cyberattacks. Users feel more confident because of this reinforced security infrastructure, which also highlights the project's dedication to safeguarding users' personal data and privacy.

Moreover, the creation of recommendations tailored to the age group is a proactive measure to encourage appropriate online conduct among teenagers. These guidelines offer users helpful guidance on securely and responsibly navigating the digital landscape. They were developed in partnership with experts in teenage psychology and cybersecurity. Through the discussion of subjects including content consumption patterns, privacy settings, and peer interaction, the guidelines enable teenagers to make well-informed decisions and develop healthy online behaviors. Overall, the Teen Credence project's outcomes demonstrate how well it works to improve adolescent social media platform registration and activity tracking.

By combining technology advancements with the creation of instructional materials in a seamless manner, the initiative has established a precedent for a more secure and reliable online environment.

```
final_cnn_score = final_cnn.evaluate(test_dataset, verbose=1)
```

✓ 9.2s

```
157/157 [==============================] - 9s 58ms/step - loss: 0.7377 - accuracy: 0.7037
```

Fig. 5.1. Evaluation of Dataset

The above snippet shows the accuracy of the model.

```
final_cnn_labels = final_cnn.metrics_names
print(f'CNN model {final_cnn_labels[0]} \t\t= {round(final_cnn_score[0], 3)}')
print(f'CNN model {final_cnn_labels[1]} \t= {round(final_cnn_score[1], 3)}')
```

✓ 0.0s

```
CNN model loss          = 0.738
CNN model accuracy      = 0.704
```

Fig. 5.2. Accuracy and Loss

The above snippet shows the accuracy and loss shown our model.

```
final_cnn_pred = final_cnn.predict(test_dataset)
final_cnn_pred = final_cnn_pred.argmax(axis=-1)
```

Fig. 5.3. Final Prediction

```
# Generating a confusion matrix based on above predictions.

conf_mat = confusion_matrix(test_labels_list, final_cnn_pred)
conf_mat
```

✓ 0.0s

```
array([[ 938,   15,    2,    0,    3,    0,    0],
       [ 129,  683,   18,   12,    1,    2,    0],
       [   2,  226,  465,  189,   54,    5,    0],
       [   4,    7,   84, 1473,  425,   13,    1],
       [   1,   19,   37,  711, 1851,  170,   18],
       [   2,    7,    2,   21,  428, 1003,  216],
       [   0,    3,    0,    2,   15,  133,  656]], dtype=int64)
```

Fig. 5.4. Confusion matrix

Chapter 06: Conclusion and Future Scope

6.1 Conclusion

Two of the most important tools for obtaining information from a person are their age and gender. There is enough information in human faces for a wide range of applications. Classifying people according to their age and gender is essential to reaching the correct audience. We tried to use common equipment to duplicate the procedure. Many factors influence the algorithm's efficiency, but this study's main objective is to make the algorithm as fast and easy to use as possible without sacrificing accuracy. The efficiency of the algorithm is being worked on. Future improvements include adding more datasets for individuals from different ethnic groups, eliminating faces from non-human objects, and granting the computer more precise control over its workflow. The ability of this prototype to accurately determine a person's gender and age range from a single face image could be enhanced with the use of deep learning and CNN. Based on this investigation, two significant conclusions can be drawn. First, CNN can be used to improve age and gender detection results, even though age and gender-tagged photos are not widely available. Second, the system's performance can be marginally improved by using more complex systems and more training data.

6.2 Future Scope

1 *User-Friendly Website Development:* Using Flask, a lightweight and adaptable web framework for Python, the "TEEN CREDENCE" project's future development will concentrate on producing a user-friendly website. The purpose of this addition is to give users—parents or guardians included—an easy-to-use interface so they can interact with the system without difficulty.

2 *Enhanced Usability and Accessibility:* Improving the system's overall usability and accessibility is the main objective of this extension. A website that is easy to use will

make it simpler for people to sign up teenagers for the platform and keep an eye on how they use social media.

3 *Integration of Age Prediction Model*: The age prediction model that was initially created for the project will be integrated into the website. This feature improves the accuracy of the system by ensuring that current data on teenagers' digital interactions is provided.

4 *Interactive User Interface:* The website will provide an interactive user interface that makes it easier for users to interact with the TEEN CREDENCE system by utilizing Flask's features. This is consistent with the project's goal of developing a solution that is both user-centered and robust.

5 *Extra Features for an Even Better User Experience*: More features will be added to improve the user experience even more. These consist of secure user authentication for increased system security, extensive activity logs for in-depth insights, and user-customizable monitoring settings that provide users more control.

6 *Alignment with Project Objectives:* The planned future work is in perfect harmony with TEEN CREDENCE's main goal, which is to create a comprehensive solution that gives teens' digital wellbeing in the constantly changing social media environment top priority. The focus on usability, accessibility, and other features highlights the dedication developing comprehensive and efficient platform for adolescent digital safe.

References

[1] G. Guo, Y. Fu, C.R. Dyer, T.S. Huang, "Image-based Human Age Estimation by Manifold Learning and Locally Adjusted Robust Regression," IEEE Transactions on Image Processing, vol. 17, no. 7, pp. 1178-1188, July 2008.

[2] E. Eidinger, R. Enbar and T. Hassner, "Age and Gender Estimation of Unfiltered Faces," in IEEE Transactions on Information Forensics and Security, vol. 9, no. 12, pp. 2170-2179, Dec. 2014, doi: 10.1109/TIFS.2014.2359646.

[3] B.C. Chen, C.S. Chen, W.H. Hsu, "Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval," in D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds.), Computer Vision – ECCV 2014, vol. 8694, Springer, Cham, 2014, pp. 49-64, https://doi.org/10.1007/978-3-319-10599-4_49.

[4] G. Guo, C. Zhang, "A Study on Cross-Population Age Estimation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 4257-4263.

[5] R. Rothe, R. Timofte, L. Van Gool, "DEX: Deep EXpectation of Apparent Age from a Single Image," in 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, 2015.

[6] A. Levi and T. Hassner, "Age and Gender Classification using Convolutional Neural Networks," in 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 2015, pp. 34-42, doi: 10.1109/CVPRW.2015.7301352.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going Deeper With Convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9.

[8] Y. Dong, Y. Liu, S. Lian, "Automatic Age Estimation based on Deep Learning Algorithm," Neurocomputing, vol. 187, pp. 4-10, 2016, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2015.

[9] S. Escalera, M. Torres Torres, B. Martinez, C. Corneou, M. Oliu, M. Ali Bagheri, et al., "Chalearn Looking at People and Faces of the World: Face Analysis Workshop and Challenge 2016," in CVPR Workshops, 2016, pp. 1-8.

[10] ChaLearn Looking at People and Faces of the World: Face Analysis Workshop and Challenge 2016 Sergio Escalera, Mercedes Torres Torres, Brais Martinez, Xavier Baro, Hugo Jair Escalante, Isabelle Guyon, Georgios Tzimiropoulos, Ciprian Corneou, Marc Oliu, Mohammad Ali Bagheri, Michel Valstar; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016, pp. 1-8.

[11] S. Chen, C. Zhang, M. Dong, J. Le, M. Rao, "Using Ranking-CNN for Age Estimation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5183-5192.

[12] Z. Tan, J. Wan, Z. Lei, R. Zhi, G. Guo and S. Z. Li, "Efficient Group-n Encoding and Decoding for Facial Age Estimation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 11, pp. 2610-2623, 1 Nov. 2018, doi: 10.1109/TPAMI.2017.2779808.

[13] B.-B. Gao, H.-Y. Zhou, J. Wu, X. Geng, "Age Estimation Using Expectation of Label Distribution Learning," in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Main track, 2018, pp. 712-718, https://doi.org/10.24963/ijcai.2018/99.

[14] S. Chen, C. Zhang and M. Dong, "Deep Age Estimation: From Classification to Ranking," in IEEE Transactions on Multimedia, vol. 20, no. 8, pp. 2209-2222, Aug. 2018, doi: 10.1109/TMM.2017.2786869.

[15] R. Rothe, R. Timofte, L. Van Gool, "Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks," International Journal of Computer Vision (IJCV), vol. 126, no. 2-4, pp. 144-157, 2018.

[16] M. Tan, Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR 97, pp. 6105-6114, 2019.

[17] Zenghui An, Shunming Li, Jinrui Wang, Yu Xin, Kun Xu, Generalization of deep neural network for bearing fault diagnosis under different working conditions using multiple kernel method, Neurocomputing, Volume 352, 2019, Pages 42-53, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.04.010.

[18] A.A. Moustafa, A. Elnakib, N.F.F. Areed, "Age-Invariant Face Recognition Based on Deep Features Analysis," Signal, Image and Video Processing (SIViP), vol. 14, pp. 1027–1034, 2020, https://doi.org/10.1007/s11760-020-01635-1.

[19] V. Carletti, A. Greco, G. Percannella, M. Vento, "Age from Faces in the Deep Learning Revolution," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 9, pp. 2113-2132, 1 Sept. 2020, doi: 10.1109/TPAMI.2019.2910522.

[20] B. Chauhan, V.R. Rangarej, S.H. Rathod, H.R. Ravikumar, "Age and Gender Detection using Deep Learning," International Research Journal of Engineering and Technology (IRJET), vol. 08, no. 02, Feb 2021.

[21] A. Saxena, P. Singh, S. Narayan Singh, "Gender and Age Detection using Deep Learning," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 719-724, doi: 10.1109/Confluence51648.2021.9377041.

[22] S. Vasavi, P. Vineela, S.V. Raman, "Age Detection in a Surveillance Video Using Deep Learning Technique," SN COMPUT. SCI., vol. 2, p. 249, 2021, https://doi.org/10.1007/s42979-021-00620-w.

[23] F. M. J. Mehedi Shamrat, S. Chakraborty, M. Alauddin and R. Ranjan, "A Deep Learning Approach for Face Detection using Max Pooling," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2021, pp. 760-764, doi: 10.1109/ICOEI51242.2021.9452896.

[24] A. Akbari, M. Awais, Z. -H. Feng, A. Farooq and J. Kittler, "Distribution Cognisant Loss for Cross-Database Facial Age Estimation With Sensitivity Analysis," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 4, pp. 1869-1887, 1 April 2022, doi: 10.1109/TPAMI.2020.3029486.

[25] Zenghao Bao, Yutian Luo, Zichang Tan, Jun Wan, Xibo Ma, Zhen Lei, Deep domain-invariant learning for facial age estimation, Neurocomputing, Volume 534,2023,ISSN 0925-2312,https://doi.org/10.1016/j.neucom.2023.02.037.

rrp

ORIGINALITY REPORT

5%	3%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	journal.ijprse.com Internet Source	<1%
2	ebin.pub Internet Source	<1%
3	export.arxiv.org Internet Source	<1%
4	thesai.org Internet Source	<1%
5	trepo.tuni.fi Internet Source	<1%
6	"Mobile Radio Communications and 5G Networks", Springer Science and Business Media LLC, 2024 Publication	<1%
7	Choi, S.E.. "Age estimation using a hierarchical classifier based on global and local facial features", Pattern Recognition, 201106 Publication	<1%
8	builtin.com Internet Source	<1%

9	www.heatinc.ca Internet Source	<1%
10	Zenghao Bao, Yutian Luo, Zichang Tan, Jun Wan, Xibo Ma, Zhen Lei. "Deep Domain-Invariant Learning for Facial Age Estimation", Neurocomputing, 2023 Publication	<1%
11	www.digitaljournal.com Internet Source	<1%
12	Tetiana Kovaliuk, Nataliya Kobets, Anastasia Shevchenko, Nataliia Kunanets. "Multibiometric Identification of Computer Network Users in the Distance Learning Process", 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022 Publication	<1%
13	Mostafa Ahmed Ezzat, Mohamed A. Abd El Ghany, Sultan Almotairi, Mohammed A.-M. Salem. "Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends", Sensors, 2021 Publication	<1%
14	hiredot.net Internet Source	<1%
	link.springer.com	

15	Internet Source	<1%
16	pdfs.semanticscholar.org Internet Source	<1%
17	www.researchgate.net Internet Source	<1%
18	Abeer Alsadoon, Ghazi Al-Naymat, Md Rafiqul Islam. "Deep learning models for human age prediction to prevent, treat and extend life expectancy: DCPV taxonomy", Multimedia Tools and Applications, 2023 Publication	<1%
19	Akash Adhikari, Pankaj Raj Dawadi. "Motion Prediction for Autonomous Vehicles Using Deep Learning Techniques", 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023 Publication	<1%
20	Gil Levi, Tal Hassncer. "Age and gender classification using convolutional neural networks", 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 Publication	<1%
21	catalog.libraries.psu.edu Internet Source	<1%

22	dspace.univ-medea.dz Internet Source	<1 %
23	encyclopedia.pub Internet Source	<1 %
24	ouci.dntb.gov.ua Internet Source	<1 %
25	topdigital.agency Internet Source	<1 %
26	www.ijariit.com Internet Source	<1 %
27	www.isarder.org Internet Source	<1 %
28	"Biometric Recognition", Springer Science and Business Media LLC, 2017 Publication	<1 %
29	Mohammed-En-nadhir Zighem, Abdelkrim Ouafi, Athmane Zitouni, Yassine Ruichek, Abdelmalik Taleb-Ahmed. "Two-stages based facial demographic attributes combination for age estimation", Journal of Visual Communication and Image Representation, 2019 Publication	<1 %
30	dokumen.pub Internet Source	<1 %

31	huggingface.co Internet Source	<1%
32	"Big Data and Visual Analytics", Springer Science and Business Media LLC, 2017 Publication	<1%
33	Sidi Lu, Yongtao Yao, Bing Luo, Zhifeng Yu, Dalong Li, Weisong Shi. "EdgeWare: toward extensible and flexible middleware for connected vehicle services", CCF Transactions on High Performance Computing, 2022 Publication	<1%
34	S Kezia, E Grace Mary Kanaga, H Eugene Kingsley, R Raghhul. "Experimental Analysis on Detection of Emotions by Facial Recognition using Different Convolution Layers", 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 20/05/24

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: Rohit Mishra / Nitika Bhatt **Department:** CSE **Enrolment No** 201465/201150

Contact No. 9424307815 **E-mail.** rohitaniket09@gmail.com

Name of the Supervisor: Dr. Shubham Goel

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):

TEEN CREDENCE

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 60
- Total No. of Preliminary pages = 11
- Total No. of pages accommodate bibliography/references = 4

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 5 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

20/5/24

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com