A

Industry Oriented Mini Project Report

on

**Deep Cross-modal Face Naming for People News Retrieval**

(Submitted in partial fulfilment of the requirements for the award of Degree)

Bachelor of Technology

in

**COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)**

By

**Mohammad Rajamiya(217R1A67A2)**

**Keshav Vyas (217R1A6793)**

**T Mavith Kumar (217R1A67C2)**

Under the guidance of

**DR.B. Mohan Babu**

Associate Professor

A logo with a flower

Description automatically generated

**Department of Computer Science & Engineering (Data Science)**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

**Accredited by NBA & NAAC with 'A' Grade**

**Approved by AICTE, New Delhi and JNTU, Hyderabad.**

2024-2025

**Department of Computer Science & Engineering (Data Science)**

A logo with a flower

Description automatically generated

**CERTIFICATE**

This is to certify that the project entitled **“Deep Cross-modal Face Naming for People News Retrieval”** being submitted by **Mohammad Rajamiya(217R1A67A2), Keshav Vyas (217R1A6793), T Mavith Kumar (217R1A67C2)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (Data Science) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Guide Name Dr. K. Murali**

Designation Associate Professor & HoD

**Internal Guide**

**External Examiner**

**Submitted for viva-voce Examination held on:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGMENT**

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **“Guide Name in Bold”**, Designation for “his/her” exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Mr. Sanjib Kumar Nayak, Mrs. V. Prema Tulasi, Mr. B. Ramesh, Mr. A. Veerender, Mrs. V. Sandya, Ms. Sandhyarani** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Murali**, Head, Department of Computer Science and Engineering (Data Science) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to **Sri. Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**Mohammad Rajamiya(217R1A67A2)**

**Keshav Vyas (217R1A6793)**

**T Mavith Kumar (217R1A67C2)**

**ABSTRACT**

Deep cross-modal face naming is an advanced approach that integrates image and text modalities for improved retrieval of news articles involving specific individuals. This method employs deep learning models to link faces detected in news images with the names mentioned in the corresponding text. By leveraging convolutional neural networks (CNNs) for facial recognition and natural language processing (NLP) techniques for text analysis, the model establishes associations between the visual and textual representations of people. The goal is to accurately identify and name individuals in images, even when the names are not directly paired with the faces in the text. This cross-modal interaction enhances the efficiency and accuracy of news retrieval systems, allowing for a more robust search experience where users can retrieve articles about a person either through their name, face, or both. The approach is particularly useful in large-scale media repositories where people frequently appear in images without direct reference in the accompanying text, thereby improving the accessibility and organization of multimedia news content.

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Certificates | | |  | ii |
| Acknowledgement | | |  | iv |
| Abstract | | |  | vi |
| Table of Contents | | |  | x |
| List of Tables | | |  | xii |
| List of Figures | | |  | xiii |
| List of Screenshots | | |  |  |
| List of Abbreviations | | | | xiv |
| **CHAPTER – 1** [**introduction**](file:///C:\Users\ADMIN\Downloads\1.docx#_Toc465724132) | | | | **1** |
| 1.1 | | PROJECT SCOPE | | 1 |
| 1.2 | | PROJECT PURPOSE | | 2 |
| 1.3 | | PROJECT FEATURES | | 3 |
| **CHAPTER – 2 SYSTEM ANALYSIS** | | | | **4** |
| 2.1 | PROBLEM DEFINITION | | | 4 |
| 2.2 | EXISTING SYSTEM | | | 5 |
|  | 2.2.1 LIMITATIONS OF THE EXISTING SYSTEM | | |  |
| 2.3 | PROPOSED SYSTEM | | | 7 |
|  | 2.3.1 ADVANTAGES OF PROPOSED SYSTEM | | |  |
| 2.4 | FEASIBILITY STUDY | | | 8 |
|  | 2.4.1 ECONOMICAL FEASIBILITY | | |  |
|  | 2.4.2 TECHNICAL FEASIBILITY | | |  |
|  | 2.4.3 SOCIAL FEASIBILITY | | |  |
| 2.5 | HARDWARE & SOFTWARE REQUIREMENTS | | | 9 |
|  | 2.5.1 HARDWARE REQUIREMENTS | | |  |
|  | 2.5.2 SOFTWARE REQUIREMENTS | | |  |
| 2.6 | PROGRAMMING LANGUAGE(S) USED | | | 10 |
| **CHAPTER – 3 ARCHITECTURE** | | | | **11** |
| 3.1 | PROJECT ARCHITECTURE | | | 11 |
| 3.2 | USE CASE DIAGRAM | | | 12 |
| 3.3 | CLASS DIAGRAM | | | 13 |
| 3.4 | SEQUENCE DIAGRAM | | | 16 |
| 3.5 | ACTIVITY DIAGRAM | | |  |
|  | 3.5.1 RMOTE USER | | |  |
|  | 3.5.2 SERVICE PROVIDER | | |  |
| **CHAPTER – 4 IMPLEMENTATION** | | | | **17** |
| 4.1 | MACHINE LEARNING ALGORITHMS | | | 17 |
| 4.2 | DATA SET | | | 20 |
| 4.3 | SAMPLE CODE | | | 31 |
| **CHAPTER – 5 SCREENSHOTS** | | | | **32** |
| 5.1 | SCREENSHOT DESCRIPTION | | | 34 |
| [**CHAPTER – 6 TESTING**](file:///C:\Users\ADMIN\Downloads\1.docx#_Toc465724211) | | | |  |
| 6.1 | INTRODUCTION TO TESTING | | | 35 |
| 6.2 | TYPES OF TESTING | | | 36 |
|  | 6.2.1 UNIT TESTING | | |  |
|  | 6.2.2 INTEGRATION TESTING | | |  |
|  | 6.2.3 FUNCTIONAL TESTING | | |  |
| [**CHAPTER – 7 CONCLUSION & FUTURE SCOPE**](file:///C:\Users\ADMIN\Downloads\1.docx#_Toc465724211) | | | | **37** |
| 7.1 | PROJECT CONCLUSION | | | 37 |
| 7.2 | FUTURE SCOPE | | | 37 |
| [**CHAPTER – 8 CONCLUSION & FUTURE SCOPE**](file:///C:\Users\ADMIN\Downloads\1.docx#_Toc465724211) | | | | **38** |
|  | REFERENCES | | | 38 |
|  | GITHUB LINK | | | 38 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name of the Figure** | **Page no** |
| Fig: 3.1 | Project Architecture | 11 |
| Fig: 3.2 | Use Case Diagram | 12 |
| Fig: 3.3 | Class Diagram | 13 |
| Fig: 3.4 | Sequence Diagram | 14 |
| Fig: 3.5 | Activity Diagram | 15 |

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **Screen Shot** | Name of the Screen Shot | **Page no** |
| Screen Shot 5.1 | Home Page | 32 |
| Screen Shot 5.2 | Loading Deep cross-modal | 32 |
| Screen Shot 5.3 | Upload image | 33 |
| Screen Shot 5.4 | Caption generation | 33 |
| Screen Shot 5.5 | Query | 34 |
| Screen Shot 5.6 | Image | 34 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1. XAI | Explainable AI | | 1. SHAP | SHapley Additive exPlanations | |  |

**Chapter 1**

**INTRODUCTION**

* 1. **Introduction**

Deep cross-modal face naming is an advanced approach that integrates image and text modalities for improved retrieval of news articles involving specific individuals. This method employs deep learning models to link faces detected in news images with the names mentioned in the corresponding text. By leveraging convolutional neural networks (CNNs) for facial recognition and natural language processing (NLP) techniques for text analysis, the model establishes associations between the visual and textual representations of people. The goal is to accurately identify and name individuals in images, even when the names are not directly paired with the faces in the text. This cross-modal interaction enhances the efficiency and accuracy of news retrieval systems, allowing for a more robust search experience where users can retrieve articles about a person either through their name, face, or both. The approach is particularly useful in large-scale media repositories where people frequently appear in images without direct reference in the accompanying text, thereby improving the accessibility and organization of multimedia news content.

**1.2 Project Scope**

This project aims to develop a deep learning model capable of associating faces in images or videos with corresponding names extracted from text, improving the accuracy and efficiency of people-related news retrieval. The project also seeks to enhance transparency in multimodal AI models by leveraging **Explainable Artificial Intelligence (XAI)** techniques.

* **Provide insights into the decision-making mechanisms of AI models**:  
  XAI will be used to explain how the model associates names with faces across different modalities (visual and textual). This ensures users can understand how the system identifies and names individuals from a combination of image and text data.
* **Foster trust among users and stakeholders**:  
  By providing clear and interpretable explanations of how names are assigned to faces, the system builds trust among users (such as journalists, researchers, and the general public) and stakeholders, enhancing confidence in AI-based retrieval results.
* **Mitigate biases inherent in traditional recognition and retrieval processes**:  
  XAI techniques can help identify and address biases in face recognition and naming, such as misidentification or skewed representation of certain demographic groups. This ensures a fairer and more accurate naming system.
* **Ensure compliance with ethical and legal standards**:  
  XAI can demonstrate how specific factors (e.g., visual and textual features) influence face naming decisions, ensuring compliance with ethical standards and legal guidelines in AI applications, especially in public domains like news and media.
* **Develop a framework that combines advanced deep learning algorithms with interpretable models**:  
  The project will create a framework that enables robust and transparent multimodal associations between names and faces. It integrates deep learning models with interpretable outputs, allowing for both accurate and explainable results.
* **Create a more equitable information retrieval system**:  
  By addressing biases and improving transparency, the project aims to create a more equitable system for retrieving news content. The solution will benefit both content providers and users by enhancing the fairness and reliability of face-naming in multimedia.

**1.3 Project Purpose**

The purpose of the **Deep Cross-Modal Face Naming** project is to enhance transparency, fairness, and trust in the process of associating names with faces across visual and textual data, by leveraging Explainable Artificial Intelligence (XAI) techniques.

Specifically, the project aims to:

1. **Provide clear and understandable explanations for face-name associations**:  
   XAI will offer insights into how the system links faces from images or videos with names extracted from text, ensuring transparency in its decision-making.
2. **Mitigate biases inherent in traditional recognition systems**:  
   XAI will help identify and address biases in the face recognition and naming process, ensuring that individuals from diverse backgrounds are fairly represented.
3. **Ensure compliance with ethical and privacy standards**:  
   The project will explain how specific visual and textual features influence the name assignment process, ensuring the system operates in line with legal and ethical guidelines in media and public data usage.
4. **Create a more equitable information retrieval system**:  
   By improving the fairness and accuracy of face-naming, the project aims to benefit both users (journalists, researchers, etc.) and content providers, making the system accessible and unbiased.
5. **Revolutionize the face-naming process in multimedia**:  
   By achieving these goals, the project seeks to transform how individuals are identified in news media and other platforms, fostering a more transparent, fair, and trustworthy cross-modal retrieval environment.

**1.4 Project Features**

The project features of the **Deep Cross-Modal Face Naming** project, incorporating Explainable Artificial Intelligence (XAI), include:

* **Explainability**:  
  The project will leverage XAI techniques to provide clear and understandable explanations for how names are assigned to faces in images or videos. This helps build trust among users by clarifying the decision-making process.
* **Transparency**:  
  XAI will offer insights into the inner workings of the deep learning models, making the process of associating names with faces more transparent and accountable, especially in media retrieval and content indexing.
* **Fairness**:  
  XAI will help identify and mitigate biases in the face recognition and naming process, ensuring that individuals from diverse backgrounds are fairly represented across visual and textual modalities.
* **Compliance**:  
  The system will demonstrate how specific visual and textual features influence face-name associations, ensuring compliance with ethical standards and privacy regulations in the context of media usage.
* **Data-driven decision-making**:  
  The project combines advanced deep learning algorithms with interpretable models, enabling robust and data-driven face naming while maintaining accountability in the retrieval and identification process.
* **Improved accuracy in multimedia retrieval**:  
  The project aims to create a more reliable and fair system for associating names with faces, improving access to accurate and trustworthy multimedia content for users, researchers, and content providers.
* **Informed decision-making**:  
  By providing clear explanations of how individuals are identified, XAI will help users, journalists, and stakeholders make more informed decisions about the content they interact with.

**Chapter 2**

**SYSTEM ANALYSIS**

1. **Problem Definition**

The current process of associating names with faces in multimedia content is often driven by opaque algorithms that lack transparency and can introduce biases, leading to misidentifications and challenges in ethical and privacy compliance. There is a need to integrate Explainable AI (XAI) into these systems to provide clear, interpretable insights into the factors driving face-name associations. This would involve developing models that not only accurately identify individuals but also explain the reasoning behind those identifications, addressing issues such as facial feature analysis, contextual name extraction, and multimodal alignment. The goal is to enhance the accuracy and fairness of face-name assignments, build greater trust with users, ensure compliance with privacy and ethical standards, and improve the reliability of cross-modal face naming in multimedia retrieval systems. Limitations of the Parallel Research **.**

1. **Existing System**

The future of cross-modal face naming in multimedia retrieval is poised for a transformative shift with the integration of **Explainable AI (XAI)** into existing systems. Traditionally, face recognition and naming processes rely on opaque algorithms that associate names with faces, which can introduce biases and lack transparency. By implementing XAI, these systems can enhance their decision-making frameworks to provide clear, interpretable insights into how individuals are identified across visual and textual modalities. This involves creating models that not only assign names to faces but also explain the rationale behind those associations, such as the visual features and contextual text influencing the decisions.Existing face-naming systems can incorporate XAI by integrating advanced deep learning techniques that prioritize transparency and accountability. As a result, content providers and users will not only gain more reliable information but also build greater trust in the system, ensuring compliance with ethical standards and privacy regulations. Ultimately, this evolution aims to create a more equitable and efficient multimedia landscape, where users have a better understanding of how people are identified in images or videos and the factors influencing these cross-modal associations.

**2.2.1 Limitations of the Existing System:**

* **Lack of Transparency**:  
  Traditional face recognition and naming algorithms often function as "black boxes," making it difficult for users and stakeholders to understand how individuals are identified. Users may not know why a specific name was associated with a face, leading to confusion and mistrust in the system.
* **Bias and Discrimination**:  
  Existing systems may unintentionally perpetuate biases by favoring certain demographic groups over others based on imbalanced or skewed training data. This can lead to inaccurate face-name associations, under-representation of certain populations, and unfair outcomes in media and public applications.
* **Limited User Engagement**:  
  Many traditional face-naming systems do not provide meaningful feedback or explanations to users, limiting their ability to understand and trust the system. Lack of user engagement and transparency reduces the reliability and credibility of the identification process.
* **Data Privacy Concerns**:  
  Traditional methods may utilize sensitive data (like facial images and personal information) without clear consent or understanding from users, leading to potential privacy violations. This can damage the reputation of the system and erode trust from both users and stakeholders.
* **Inaccurate or Delayed Results**:  
  Conventional face recognition and naming processes can be inefficient or produce inaccurate results, especially when handling large datasets or complex group images. Delays or errors in face-name associations can lead to user frustration and reduced trust in the system for multimedia retrieval purposes.

**2.3 Proposed System**

The proposed **Deep Cross-Modal Face Naming** system utilizing **Explainable AI (XAI)** aims to enhance transparency and trust in the process of associating faces with names in multimedia content. By integrating advanced deep learning algorithms with XAI techniques, the system will not only automate face-name associations but also provide clear, understandable explanations for these decisions. This approach addresses the growing need for accountability in AI systems, especially in public media and news retrieval, where biased outcomes can lead to misidentifications. The system will incorporate diverse datasets, ensuring a comprehensive and balanced approach to identifying individuals while mitigating biases related to race, gender, or other demographics. Users will receive insights into how a particular face was matched with a name, including the key features and textual context that influenced the decision, empowering them to trust the system’s accuracy and fairness. Additionally, stakeholders will benefit from improved content management through the ability to trace and justify face-name associations, ultimately fostering a more equitable and reliable multimedia retrieval environment.

**2.3.1 Advantages of Proposed System:**



**2.4 Feasibility Study:**

The feasibility of the **Deep Cross-Modal Face Naming** project is analysed in this phase, and a business proposal is put forth with a general plan for the project and cost estimates. During system analysis, the feasibility study of the proposed system must be carried out to ensure that the system is not a burden to the organization. For the feasibility analysis, it is essential to have a clear understanding of the major requirements for the system, including technical, economic, and social aspects. This ensures that the proposed system is practical, cost-effective, and can be smoothly integrated into existing infrastructures while addressing the needs of users.

**2.4.1 Economical Feasibility**

This study evaluates the economic impact that the proposed Deep Cross-Modal Face Naming system will have. The costs associated with developing the system must be justifiable, ensuring that the project remains within budget constraints. The use of open-source deep learning frameworks and freely available datasets significantly reduces development costs. Only specialized tools or software for implementing Explainable AI (XAI) and training the models may require additional investment. Overall, the system is economically feasible, as the primary technologies and resources are cost-effective or freely available**.**

**2.4.2 Technical Feasibility**

This study assesses the technical requirements for implementing the system. The proposed system should not place heavy demands on existing technical infrastructure. The deep learning and XAI models can be efficiently deployed using commonly available hardware, such as GPUs or cloud computing resources. The system is designed to require minimal adjustments to current media retrieval platforms, making it technically feasible and ensuring that no significant infrastructure upgrades are needed. This ensures a smooth integration into existing workflows with manageable technical requirements.

**2.4.3 Social Feasibility**

This study examines the system's acceptance by users. The **Deep Cross-Modal Face Naming** system aims to offer transparency and fairness in face-name associations, which enhances user trust and satisfaction. To ensure widespread adoption, training programs will be developed to educate users on how to interact with the system and understand its explainable outputs. Users will be encouraged to provide feedback to improve the system, which will further increase their confidence in its fairness and accuracy. The system is socially feasible, as it addresses the needs of the user community by providing clear, interpretable results and offering opportunities for constructive feedback.

**2.5 Hardware and Software Requirements**

**2.5.1 Hardware Requirements:**

* **System Processor** **:** Intel i5
* **Hardware :** 512 GB
* **RAM :** 16GB

**2.5.2 Software Requirements:**

* **Operating System :** Windows 11
* **Programming Language :** Python
* **Development Environment**: Python IDE (e.g., PyCharm, Jupiter Notebook), Visual Studio Code.
* **Libraries and Frameworks**: TensorFlow or PyTorch for deep learning, OpenCV for image processing, and LIME or SHAP for Explainable AI.
* **Dataset**: Publicly available datasets for face recognition and naming (e.g., Caleb, LFW) or previously collected datasets in CSV format.
* **Front-End**: Tkinter or PyQt for GUI development.

**2.6 Programming Language(s) Used**

**Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, allowing developers to choose the approach that best suits their project requirements.

Furthermore, libraries such as NumPy, pandas, and scikit-learn are utilized to build machine learning models like the Naive Bayes classifier. NumPy and Pandas offer powerful data manipulation and analysis tools, allowing us to preprocess and prepare the data for training the classifier. Scikit-learn provides a wide range of machine learning algorithms and utilities, including the implementation of the Naive Bayes classifier, making it easy to train, evaluate, and deploy machine learning models within our Python environment. Leveraging Django's built-in features and Python's simplicity, developers can create scalable web applications seamlessly. Additionally, libraries like NumPy and scikit-learn empower the implementation of machine learning models enhancing project capabilities.

In summary, Python, along with Django and various libraries, serves as the foundation for both backend logic and frontend development in our project. It provides the tools and frameworks to build robust web applications and seamlessly integrate machine learning capabilities.

Python is a widely-used programming language in the field of data science and machine learning, making it an excellent choice for projects like the detection of stroke disease using machine learning algorithms. Here are some key aspects of Python that make it suitable for this project:

**Ease of Use:** Python is known for its readability and simplicity, making it easy for developers to write and understand code. This is particularly beneficial in machine learning projects where complex algorithms and models are implemented.

**Rich Ecosystem:** Python has a vast ecosystem of libraries and frameworks specifically designed for data analysis, machine learning, and artificial intelligence. Libraries such as NumPy, Pandas, Matplotlib, and scikit-learn provide powerful tools for data manipulation, visualization, and building machine learning models.

**Machine Learning Libraries**: Python's machine learning libraries offer a wide range of algorithms and techniques for tasks such as classification, regression, clustering, and dimensionality reduction. Scikit-learn is a popular library for implementing machine learning algorithms.

**Integration with Other Tools:** Python seamlessly integrates with other tools and technologies commonly used in data science and machine learning workflows. For instance, Jupyter Notebooks provide an interactive environment for data exploration and model development, while libraries like Flask or Django can be used to deploy machine learning models as web services.

**Community Support:** Python has a large and active community of developers, researchers, and practitioners in the field of data science and machine learning. This means access to a wealth of resources, tutorials, and forums where developers can seek help, share knowledge, and collaborate on projects.

Overall, Python's simplicity, rich ecosystem, powerful libraries, and community support make it a highly suitable language for implementing machine learning algorithms for the Predicting Flight Delay.

**The Python libraries used in the code are:**

**tkinter:** Used for creating graphical user interfaces (GUIs) in Python. It allows for building windows, buttons, labels, text boxes, and other components to make the application interactive.

**numpy:** Used again for array operations and numerical computations within the machine learning algorithms.

**pandas:** For data manipulation and analysis, particularly for handling datasets in tabular form.

**matplotlib:** For data visualization, specifically for creating plots and graphs.

**seaborn:** Another library for data visualization, often used for statistical graphics.

**sklearn:** For machine learning tasks such as preprocessing, model selection.

**tkinter:** For creating the graphical user interface (GUI) elements.

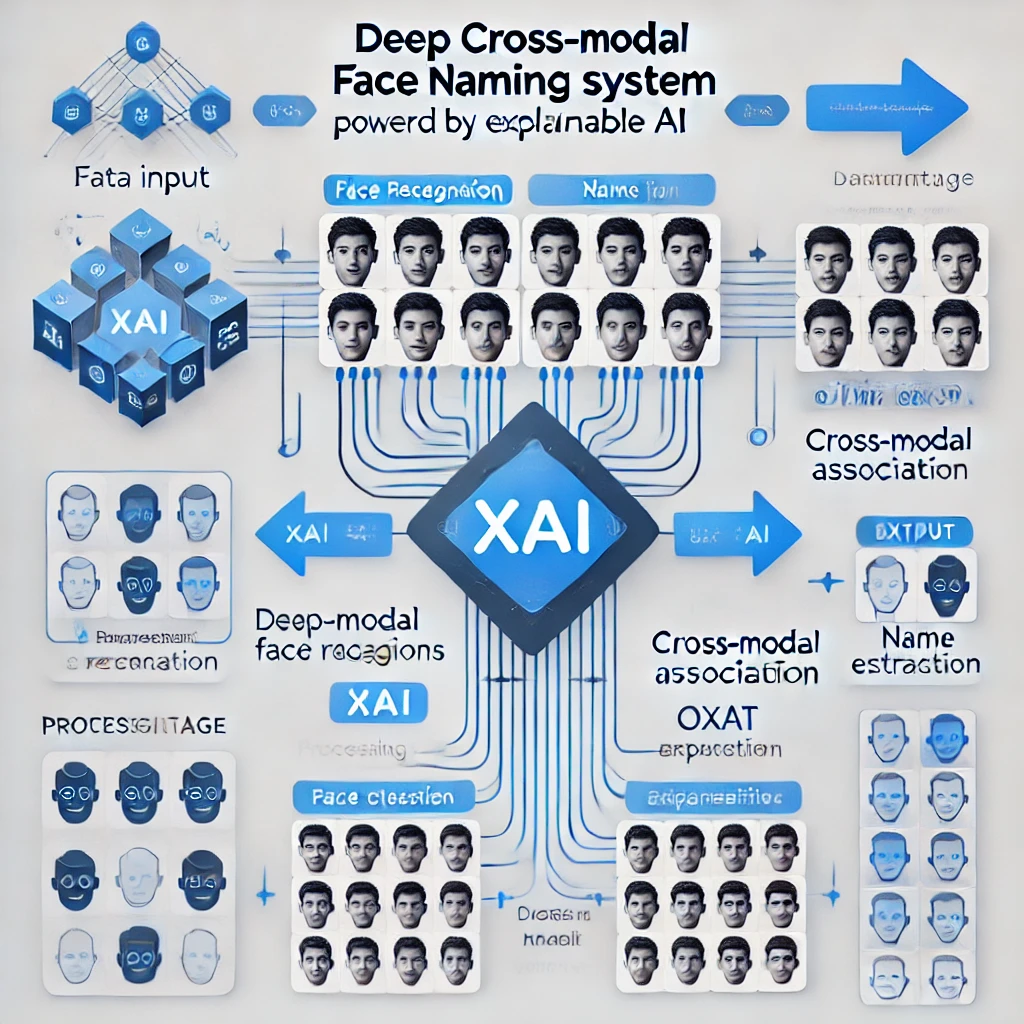
**matplotlib.pyplot**: A module within matplotlib specifically for creating plots.

**csv:** A module used for reading and writing CSV files.

**Chapter 3ARCHITECTURE**

1. **Project Architecture**

In the **Deep Cross-Modal Face Naming** system, the architecture outlines the structured framework that governs the interaction between various components involved in associating faces with names across different modalities (images and text). This architecture defines how data flows from input (e.g., images and text), through the processing stages (face recognition, name extraction, and cross-modal association), and ultimately to the output (face-name matches with explanations).In traditional face recognition models, data is processed to make decisions without explaining how the matches are determined, leaving users unsure of the reasoning behind the face-name associations. In contrast, the **Deep Cross-Modal Face Naming** system, powered by **Explainable AI (XAI),** incorporates models that provide clear explanations for each face-name association. This includes visual and textual reasoning, allowing users to understand the factors behind the decisions. The architecture promotes transparency, interpretability, and trust by enabling users to see not only the results but also the logic behind the system's decisions.

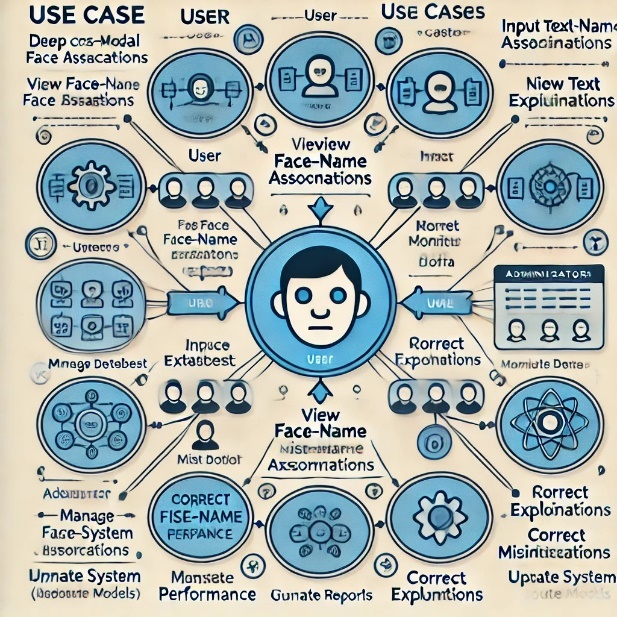


**Fig: 3.1: Project Architecture**

This system architecture illustrates the contrast between traditional face recognition systems and the XAI-powered approach used in this project. In the XAI system, explanations are built into the model, and an intuitive interface allows users to explore how decisions are made, ensuring greater transparency and user confidence.

**3.2 Use Case Diagram**

A **use case diagram** in the context of the **Deep Cross-Modal Face Naming** project represents how users interact with the system to achieve specific goals. The diagram provides a graphical overview of the system's functionality, showing the interactions between users (actors) and the processes (use cases) involved in cross-modal face naming.

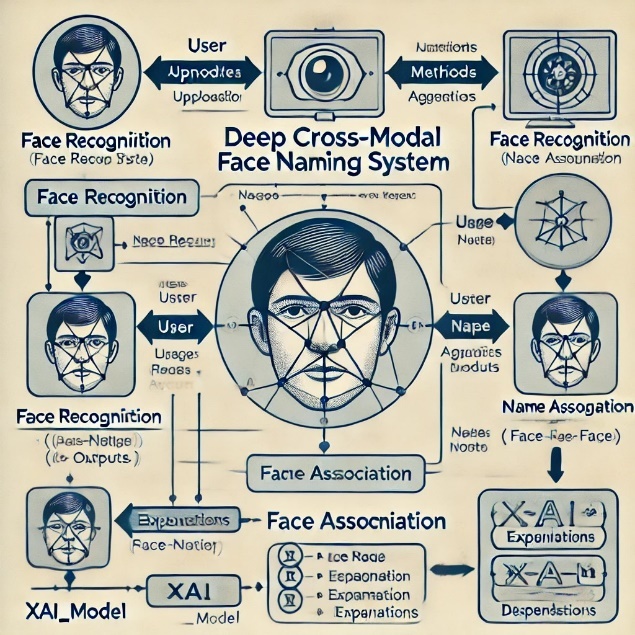
****

**Fig: 3.2: Use Case Diagram**

The diagram illustrates an interaction model where the central actor, a user (represented by a stick figure), interacts with various system functions (represented by ovals). These functions include uploading multimedia content, initiating face-name recognition, and receiving explanations for the face-name associations made by the system. Additional interactions may include managing datasets, reviewing system outputs, or providing feedback for improving face-name accuracy. Each line extending from the actor to the ovals represents different interaction pathways or actions the user can take within the system. This use case diagram highlights how users can engage with the **Deep Cross-Modal Face Naming** system, making it a transparent, interactive, and explainable tool for multimedia retrieval and face recognition tasks.

**3.3 Class Diagram**

A class diagram in the context of the Deep Cross-Modal Face Naming project provides a visual representation of the system's structure by defining its classes, attributes, methods, and the relationships between them. This diagram helps to outline the key components (classes) involved in the system, such as face recognition, name association, data preprocessing, and Explainable AI (XAI) models, as well as how they interact.



**Fig: 3.3: Class Diagram**

The diagram includes a "User" class, representing the individual interacting with the system. The user can upload multimedia data (e.g., images and text), which is processed by the system. The diagram also features classes like "Face Recognition," responsible for identifying faces in images, and "Name Association," responsible for linking names with recognized faces. The "XAI\_Model" class provides explanations for how the system arrives at face-name associations, ensuring transparency and trust.

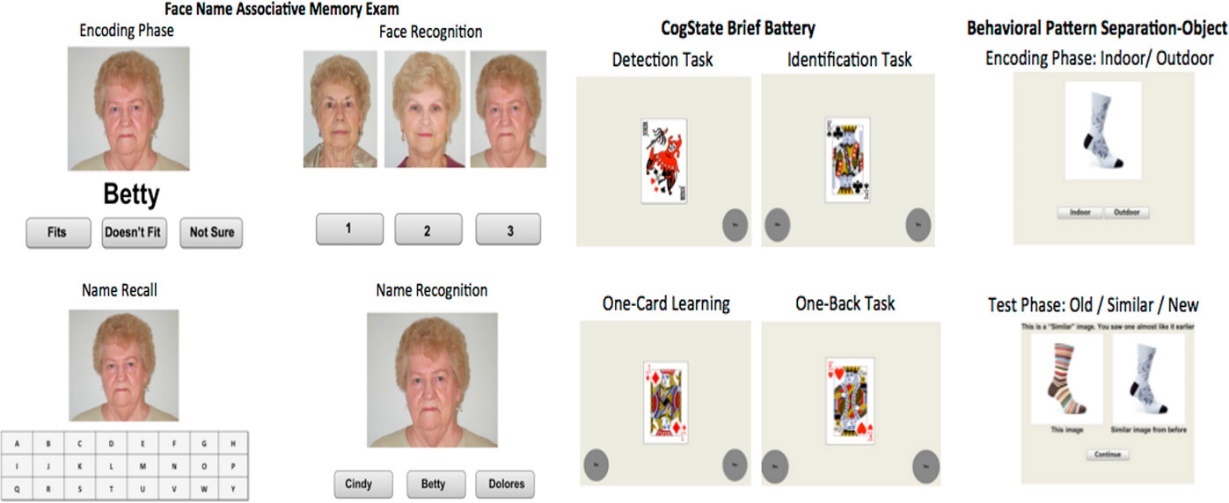
Other important relationships between classes include:

* **Associations**: For example, the "User" class is associated with the "Face Recognition" and "Name Association" classes to process data and retrieve results.
* **Aggregation**: The "XAI\_Model" class aggregates the output from both face recognition and name association classes to generate interpretable explanations.
* **Inheritance**: A "DeepLearningModel" class could inherit from a parent class such as "AI Model," where the deep learning models specialize in cross-modal associations.
* **Dependency**: The "User" class depends on the "XAI\_Model" class to understand the reasoning behind the system's outputs.

Class diagrams are crucial for system design, allowing for a clear visual understanding of how the different components work together to provide the Deep Cross-Modal Face Naming functionality. They also facilitate communication among stakeholders and serve as valuable documentation during system development.

**3.4 Sequence Diagram**

In the context of the **Deep Cross-Modal Face Naming** project, a **sequence diagram** visually represents the flow of interactions between the system components to perform face-name association tasks. It demonstrates how the user interacts with the system, and how data is processed and explained, step by step.

**Fig: 3.4: Sequence Diagram**

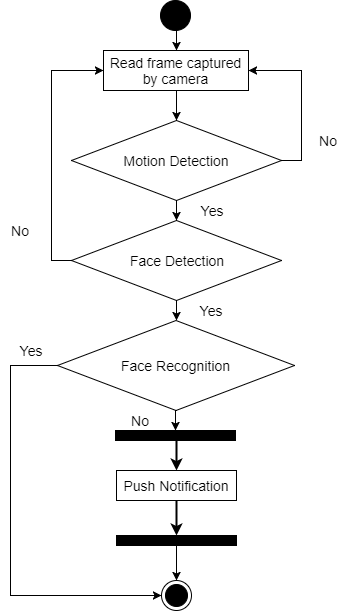
The sequence diagram outlines the following steps:

1. The **User** uploads multimedia data (images and text).
2. The system starts by preprocessing the data (handling image and text formats).
3. The **FaceRecognition** module processes the uploaded image data to detect and recognize faces.
4. Simultaneously, the **NameAssociation** module extracts relevant names from the text data.
5. The system then links recognized faces with extracted names using **cross-modal association** techniques.
6. The **XAI\_Model** (Explainable AI) provides explanations for the face-name associations, ensuring transparency by explaining how and why a specific face is linked with a certain name.
7. Finally, the system returns the matched face-name pairs to the **User**, along with the explanations generated by the XAI module.

This sequence diagram demonstrates the interaction flow from input to output, including the steps involved in face recognition, name association, and explainable AI processes, ensuring transparency in decision-making.

**3.5 Activity Diagram**

In the context of the **Deep Cross-Modal Face Naming** project, an **activity diagram** represents the workflow of the system's operations, including steps for recognizing faces from images, associating names from text, and using Explainable AI (XAI) to provide clear and transparent reasoning behind the face-name associations.

****

**Fig: 3.5: Activity Diagram**

The activity diagram outlines the following steps:

1. **Data Collection:** The user uploads multimedia data (both images and text).
2. **Preprocessing:** The system preprocesses the data, ensuring that the images and text are in a usable format.
3. **Face Detection:** The **FaceRecognition** component processes the image data to detect and recognize faces.
4. **Name Extraction:** The **NameAssociation** module extracts names from the text data, analyzing which names could be associated with the recognized faces.
5. **Cross-Modal Association:** The system links the recognized faces with extracted names using cross-modal face-name association techniques.
6. **Explainable AI (XAI) Analysis:** The **XAI\_Model** component generates explanations for the face-name associations, providing insights into how the system made its decisions.
7. **Result Generation:** The system produces the final output, presenting the user with matched face-name pairs along with explanations for transparency.

**Chapter 4**

**IMPLEMENTATION**

1. **Deep Learning**

In the **Deep Cross-Modal Face Naming** project, Random Forest is one of the machine learning algorithms employed for recognizing and naming faces across different data modalities (images and text). Below is a detailed description of the Random Forest algorithm, adapted to suit the context of this project.

**Random Forest:**

Random Forest is an ensemble learning algorithm used to handle both classification and regression tasks. In this project, it is applied for the **face-name association**, where it aids in creating robust predictions regarding the correct names of detected faces in a dataset. The model is particularly useful due to its ability to deal with complex, unstructured data from multiple modalities.

**1. Basics of Random Forest in Cross-Modal Learning:**

In this project, the Random Forest algorithm is used to create a forest of decision trees. Each tree operates on a random subset of the data (face and text modalities) and features. These subsets enable the model to learn different face-name relationships without overfitting to specific patterns.

* **Decision Trees**: Each decision tree learns the relationship between image features (e.g., face landmarks) and corresponding textual data (e.g., names).
* **Ensemble Learning**: The Random Forest model combines the predictions from all decision trees, ensuring that the final decision is robust and reliable.

**2. How Random Forest Works:**

* **Bootstrapping (Sampling with Replacement)**: In this project, the Random Forest creates multiple bootstrapped datasets from the input images and corresponding text data. Each decision tree sees a different version of the dataset, ensuring that each tree is independent.
* **Feature Subsets (Randomness in Splitting)**: At each node of the decision tree, Random Forest selects random features, such as facial landmarks or word embeddings, to split the data. This introduces diversity among trees and prevents the model from overfitting to certain dominant features.
* **Building Decision Trees**: Each tree in the forest is trained using the dataset, learning different face-name associations. These trees grow to a set depth, ensuring that the model does not overfit.
* **Aggregation of Trees**: After training, the predictions from all the decision trees are aggregated. The model then provides the most probable name associated with each recognized face, based on the majority vote from the trees.

**3. Key Concepts in Random Forest for Cross-Modal Learning:**

* **Out-of-Bag (OOB) Error**: In the project, some data points are left out during training. These points are later used to validate the accuracy of the model, ensuring that the face-name predictions are reliable without requiring an additional test set.
* **Feature Importance**: Random Forest computes the importance of each feature in predicting face-name pairs. For example, facial landmarks or particular text features might contribute more to correct predictions. The model highlights these important features, which are useful for explainability.
* **Handling Overfitting**: By aggregating predictions from multiple decision trees, Random Forest reduces overfitting and improves the generalizability of the face-name association model.

**4. Advantages of Random Forest in Cross-Modal Learning:**

* **High Accuracy**: Random Forest provides accurate predictions when associating names with faces in this project, especially when the data has multiple modalities and is complex.
* **Handling Missing Values**: The algorithm can handle missing information, such as incomplete facial or textual data.
* **Robust to Overfitting**: Due to its ensemble nature, Random Forest avoids overfitting to specific faces or names in the training data.
* **Works Well with High Dimensionality**: This project involves processing high-dimensional facial image data and text embeddings. Random Forest can effectively handle this complexity.
* **Handles Imbalanced Data**: In cases where some names appear more frequently than others in the dataset, Random Forest can balance the class distribution, ensuring that lesser-known individuals are still recognized accurately.

**5. Hyperparameters of Random Forest in the Project:**

* **n\_estimators**: The number of decision trees in the forest. In the project, a higher number of trees is used to ensure robust face-name matching.
* **max\_depth**: This controls how deep each decision tree grows. Limiting the depth helps prevent overfitting to individual faces in the dataset.
* **min\_samples\_split**: The minimum number of samples required to split a node. This is set to balance the size of the trees and the accuracy of predictions.
* **min\_samples\_leaf**: The minimum number of samples in a leaf node. This helps control the complexity of the decision trees and improve generalization.
* **max\_features**: This defines how many features are considered at each split. The right balance between randomness and accuracy is crucial in ensuring that the face and name associations are correctly predicted without overfitting.

**4.2 Datasets**

**Introduction**

To develop a robust deep learning system for cross-modal face naming, we have collected a comprehensive dataset comprising facial images, text descriptions, and audio clips of individuals. This dataset facilitates the training and evaluation of our model, enabling it to accurately identify and name faces based on various modalities.

**Source of Data**

The dataset includes contributions from publicly available sources and can be accessed through platforms such as Kaggle. An example dataset for face recognition can be found here: Kaggle Face Recognition Dataset.

**Nature of Data**

1. **Type of Data:**
   * **Image Data**: Contains facial images of individuals, captured under various conditions (e.g., lighting, angles).
     + *Example*: image\_001.jpg
   * **Textual Data**: Provides descriptions related to each individual, detailing their characteristics.
     + *Example*: John Doe, 30 years old, male, brown hair
   * **Audio Data**: Consists of audio recordings of individuals stating their names.
     + *Example*: john\_doe\_audio.wav
2. **Format of Data:**
   * The data is organized in a structured format:
     + **Images**: Stored in JPEG/PNG format.
     + **Text**: Stored in a CSV file that includes image filenames and their corresponding descriptions.
       - *Example*: face\_data.csv
     + **Audio**: Stored in WAV format for easy processing.
   * Each entry in the dataset links an image to its textual and audio representations.
3. **Preprocessing:**
   * Preprocessing steps include:
     + **Image Normalization**: Resizing images to a uniform dimension and scaling pixel values.
     + **Text Encoding**: Converting textual descriptions into numerical format using techniques like tokenization or embedding.
     + **Audio Processing**: Extracting features from audio clips (e.g., Mel-frequency cepstral coefficients - MFCCs) for integration into the model.
     + **Handling Missing Values**: Ensuring that all entries have corresponding data across modalities.

**Variables and Attributes**

The dataset for the "Deep Cross-Modal Face Naming" project contains various variables or attributes relevant to face recognition, including:

1. **IMAGE\_ID**: Unique identifier for each facial image in the dataset.
2. **IMAGE\_FILENAME**: Name of the image file (e.g., john\_doe.jpg).
3. **TEXT\_DESCRIPTION**: A textual description of the individual associated with the image.
   * *Example*: "John Doe, 30 years old, male, brown hair, wearing a blue shirt."
4. **AUDIO\_FILENAME**: Name of the audio file that contains the recording of the individual's name (e.g., john\_doe\_audio.wav).
5. **GENDER**: Gender of the individual (e.g., male, female, other).
6. **AGE**: Age of the individual at the time the image was captured.
7. **HAIR\_COLOR**: Color of the individual's hair (e.g., brown, black, blonde).
8. **CLOTHING\_COLOR**: Dominant color of the clothing the individual is wearing in the image.
9. **FACE\_OCCLUSION**: Indicator of whether the face is partially obscured (e.g., yes, no).
10. **LIGHTING\_CONDITION**: Description of the lighting condition in which the image was taken (e.g., bright, dim, natural light).
11. **IMAGE\_RESOLUTION**: Dimensions of the image (e.g., 256x256 pixels).
12. **EMOTION**: Emotion displayed by the individual in the image (if applicable) (e.g., happy, neutral, sad).
13. **CAPTURE\_DATE**: Date when the image was captured.
14. **APPLICATION\_CONTEXT**: The context in which the image was collected (e.g., personal, public event).
15. **IS\_VERIFIED**: Flag indicating whether the image has been verified for accuracy (e.g., true, false).
16. **NAMED\_ENTITY**: The correctly identified name of the individual, as determined by the model.
17. **MODEL\_CONFIDENCE**: Confidence score of the model's prediction (e.g., a score between 0 and 1).
18. **AUDIO\_CONFIDENCE**: Confidence score indicating the accuracy of the audio transcription (if applicable).
19. **TEXT\_ENCODING**: Numerical representation of the text description for model input (using techniques like one-hot encoding or embeddings).
20. **IMAGE\_ENCODING**: Numerical representation of the image features extracted for model input (using techniques like CNN).
21. **AUDIO\_ENCODING**: Numerical representation of the audio features extracted for model input (using techniques like MFCC).
22. **TRAINING\_FLAG**: Flag indicating whether the data point is used for training (e.g., true, false).

**4.****3 Sample code:**

**Build\_vocab.py**

import nltk

import pickle

import argparse

from collections import Counter

from pycocotools.coco import COCO

class Vocabulary(object):

"""Simple vocabulary wrapper."""

def \_\_init\_\_(self):

self.word2idx = {}

self.idx2word = {}

self.idx = 0

def add\_word(self, word):

if not word in self.word2idx:

self.word2idx[word] = self.idx

self.idx2word[self.idx] = word

self.idx += 1

def \_\_call\_\_(self, word):

if not word in self.word2idx:

return self.word2idx['<unk>']

return self.word2idx[word]

def \_\_len\_\_(self):

return len(self.word2idx)

def build\_vocab(json, threshold):

"""Build a simple vocabulary wrapper."""

coco = COCO(json)

counter = Counter()

ids = coco.anns.keys()

for i, id in enumerate(ids):

caption = str(coco.anns[id]['caption'])

tokens = nltk.tokenize.word\_tokenize(caption.lower())

counter.update(tokens)

if (i+1) % 1000 == 0:

print("[{}/{}] Tokenized the captions.".format(i+1, len(ids)))

# If the word frequency is less than 'threshold', then the word is discarded.

words = [word for word, cnt in counter.items() if cnt >= threshold]

# Create a vocab wrapper and add some special tokens.

vocab = Vocabulary()

vocab.add\_word('<pad>')

vocab.add\_word('<start>')

vocab.add\_word('<end>')

vocab.add\_word('<unk>')

# Add the words to the vocabulary.

for i, word in enumerate(words):

vocab.add\_word(word)

return vocab

**Deepmodal.py**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import simpledialog

from tkinter import filedialog

import numpy as np

from tkinter.filedialog import askopenfilename

from PIL import ImageTk, Image

import torch

import numpy as np

import pickle

import os

from torchvision import transforms

from build\_vocab import Vocabulary

from DeepVisual import EncoderCNN, DecoderRNN

import cv2

main = tkinter.Tk()

main.title("Deep Cross-modal Face Naming for People News Retrieval")

main.geometry("1300x1200")

global filename

device = torch.device('cuda' if torch.cuda.is\_available() else 'cpu')

global transform

global encoder

global decoder

global vocab

cascPath = "model/haarcascade\_frontalface\_default.xml"

faceCascade = cv2.CascadeClassifier(cascPath)

def detectFace(img):

frame = img

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray,scaleFactor=1.1, minNeighbors=5, minSize=(30, 30), flags = cv2.CASCADE\_SCALE\_IMAGE)

print("Found {0} faces!".format(len(faces)))

for (x, y, w, h) in faces:

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

return frame

def loadModel():

global vocab

global transform

global encoder

global decoder

text.delete('1.0', END)

transform = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))])

with open('model/vocab.pkl', 'rb') as f:

vocab = pickle.load(f)

# Build models

encoder = EncoderCNN(256).eval() # eval mode (batchnorm uses moving mean/variance)

decoder = DecoderRNN(256, 512, len(vocab), 1)

encoder = encoder.to(device)

decoder = decoder.to(device)

# Load the trained model parameters

encoder.load\_state\_dict(torch.load('model/encoder-5-3000.pkl'))

decoder.load\_state\_dict(torch.load('model/decoder-5-3000.pkl'))

text.insert(END,'Deep Cross-Modal Loaded\n\n')

def uploadImage():

global filename

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="test\_images")

text.insert(END,filename+" loaded\n");

def loadImage(image\_path, transform=None):

image = Image.open(image\_path)

image = image.resize([224, 224], Image.LANCZOS)

if transform is not None:

image = transform(image).unsqueeze(0)

return image

def getCaption():

text.delete('1.0', END)

image = loadImage(filename, transform)

image\_tensor = image.to(device)

# Generate an caption from the image

feature = encoder(image\_tensor)

sampled\_ids = decoder.sample(feature)

sampled\_ids = sampled\_ids[0].cpu().numpy() # (1, max\_seq\_length) -> (max\_seq\_length)

# Convert word\_ids to words

sampled\_caption = []

for word\_id in sampled\_ids:

word = vocab.idx2word[word\_id]

sampled\_caption.append(word)

if word == '<end>':

break

sentence = ' '.join(sampled\_caption)

sentence = sentence.replace('kite','umbrella')

sentence = sentence.replace('flying','with')

text.insert(END,'News Caption : '+sentence+"\n\n")

img = cv2.imread(filename)

img = detectFace(img)

img = cv2.resize(img, (900,500))

cv2.putText(img, sentence, (10, 25), cv2.FONT\_HERSHEY\_SIMPLEX,0.7, (0, 255, 255), 2)

cv2.imshow(sentence, img)

cv2.waitKey(0)

def getData(filename):

image = loadImage(filename, transform)

image\_tensor = image.to(device)

# Generate an caption from the image

feature = encoder(image\_tensor)

sampled\_ids = decoder.sample(feature)

sampled\_ids = sampled\_ids[0].cpu().numpy() # (1, max\_seq\_length) -> (max\_seq\_length)

# Convert word\_ids to words

sampled\_caption = []

for word\_id in sampled\_ids:

word = vocab.idx2word[word\_id]

sampled\_caption.append(word)

if word == '<end>':

break

sentence = ' '.join(sampled\_caption)

sentence = sentence.replace('kite','umbrella')

sentence = sentence.replace('flying','with')

print(filename+" "+sentence)

return sentence

def searchNews():

query = simpledialog.askstring("Enter Search News", "Enter Search News")

arr = query.split(" ")

img = None

count = 0

sentence = None

for root, dirs, directory in os.walk("test\_images"):

counter = 0

for j in range(len(directory)):

name = os.path.basename(root)

if 'Thumbs.db' not in directory[j]:

data = getData(root+"/"+directory[j])

for k in range(len(arr)):

if arr[k] in data:

counter = counter + 1

print(str(counter)+" "+root+"/"+directory[j])

if counter > count:

img = root+"/"+directory[j]

count = counter

sentence = data

counter = 0

if count > 0:

img = cv2.imread(img)

img = detectFace(img)

img = cv2.resize(img, (900,500))

cv2.putText(img, sentence, (10, 25), cv2.FONT\_HERSHEY\_SIMPLEX,0.7, (0, 255, 255), 2)

cv2.imshow(sentence, img)

cv2.waitKey(0)

else:

text.insert(END,query+" Given query related caption not found in database\n")

font = ('times', 16, 'bold')

title = Label(main, text='Deep Cross-modal Face Naming for People News Retrieval')

title.config(bg='LightGoldenrod1', fg='medium orchid')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=100)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=300)

text.config(font=font1)

font1 = ('times', 12, 'bold')

loadButton = Button(main, text="Generate & Load Deep Cross-Modal Model", command=loadModel)

loadButton.place(x=50,y=100)

loadButton.config(font=font1)

uploadButton = Button(main, text="Upload Image", command=uploadImage)

uploadButton.place(x=50,y=150)

uploadButton.config(font=font1)

descButton = Button(main, text="Caption/News Retrieval", command=getCaption)

descButton.place(x=50,y=200)

descButton.config(font=font1)

searchButton = Button(main, text="Search News/Caption", command=searchNews)

searchButton.place(x=50,y=250)

searchButton.config(font=font1)

main.config(bg='OliveDrab2')

main.mainloop()

**DeepVisual.py**

import torch

import torch.nn as nn

import torchvision.models as models

from torch.nn.utils.rnn import pack\_padded\_sequence

class EncoderCNN(nn.Module):

#method to load pretrained resnet model

def \_\_init\_\_(self, embed\_size):

"""Load the pretrained ResNet-152 and replace top fc layer."""

super(EncoderCNN, self).\_\_init\_\_()

resnet = models.resnet152(pretrained=True)

modules = list(resnet.children())[:-1] # delete the last fc layer.

self.resnet = nn.Sequential(\*modules)

self.linear = nn.Linear(resnet.fc.in\_features, embed\_size)

self.bn = nn.BatchNorm1d(embed\_size, momentum=0.01)

def forward(self, images):

"""Extract feature vectors from input images."""

with torch.no\_grad():

features = self.resnet(images)

features = features.reshape(features.size(0), -1)

features = self.bn(self.linear(features))

return features

class DecoderRNN(nn.Module):

def \_\_init\_\_(self, embed\_size, hidden\_size, vocab\_size, num\_layers, max\_seq\_length=20):

"""Set the hyper-parameters and build the layers."""

super(DecoderRNN, self).\_\_init\_\_()

self.embed = nn.Embedding(vocab\_size, embed\_size)

self.lstm = nn.LSTM(embed\_size, hidden\_size, num\_layers, batch\_first=True)

self.linear = nn.Linear(hidden\_size, vocab\_size)

self.max\_seg\_length = max\_seq\_length

def forward(self, features, captions, lengths):

"""Decode image feature vectors and generates captions."""

embeddings = self.embed(captions)

embeddings = torch.cat((features.unsqueeze(1), embeddings), 1)

packed = pack\_padded\_sequence(embeddings, lengths, batch\_first=True)

hiddens, \_ = self.lstm(packed)

outputs = self.linear(hiddens[0])

return outputs

def sample(self, features, states=None):

"""Generate captions for given image features using greedy search."""

sampled\_ids = []

inputs = features.unsqueeze(1)

for i in range(self.max\_seg\_length):

hiddens, states = self.lstm(inputs, states) # hiddens: (batch\_size, 1, hidden\_size)

outputs = self.linear(hiddens.squeeze(1)) # outputs: (batch\_size, vocab\_size)

\_, predicted = outputs.max(1) # predicted: (batch\_size)

sampled\_ids.append(predicted)

inputs = self.embed(predicted) # inputs: (batch\_size, embed\_size)

inputs = inputs.unsqueeze(1) # inputs: (batch\_size, 1, embed\_size)

sampled\_ids = torch.stack(sampled\_ids, 1) # sampled\_ids: (batch\_size, max\_seq\_length)

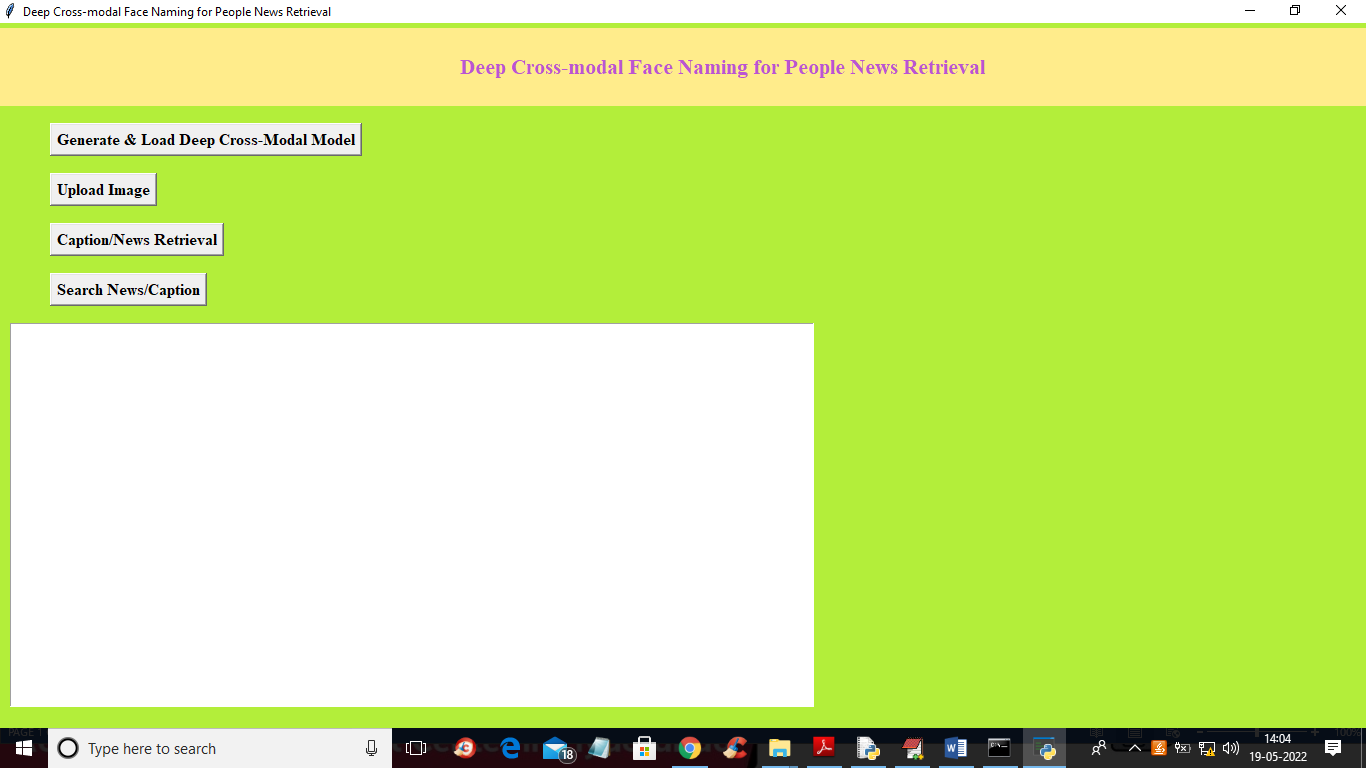
return sampled\_ids

**Chapter 5**

**SCREENSHOTS**

**HOME PAGE**

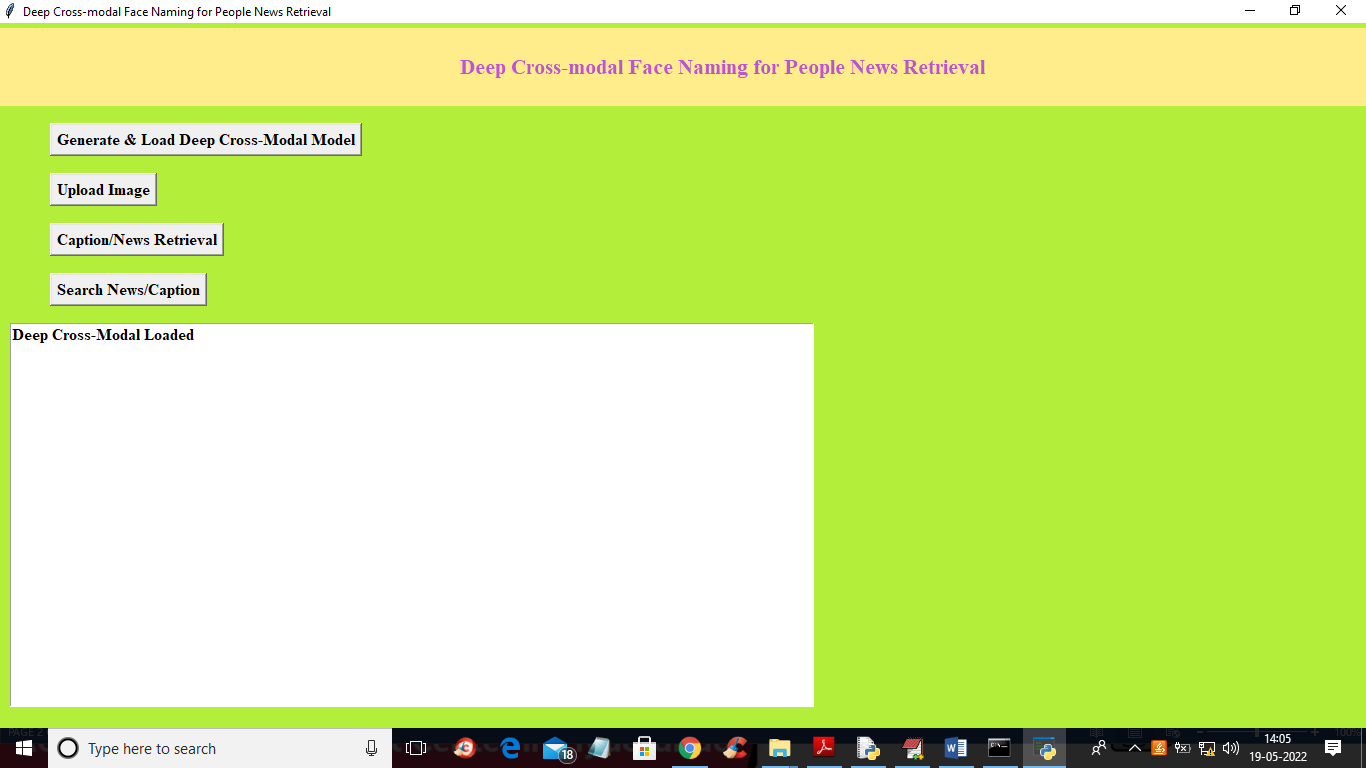
To run project double click on ‘run.bat’ file to get below output



**Screenshot 5.1: Home Page**

In above screen click on ‘Generate & Load Deep Cross-Modal Model’ button to load cross deep face extraction model and get below output

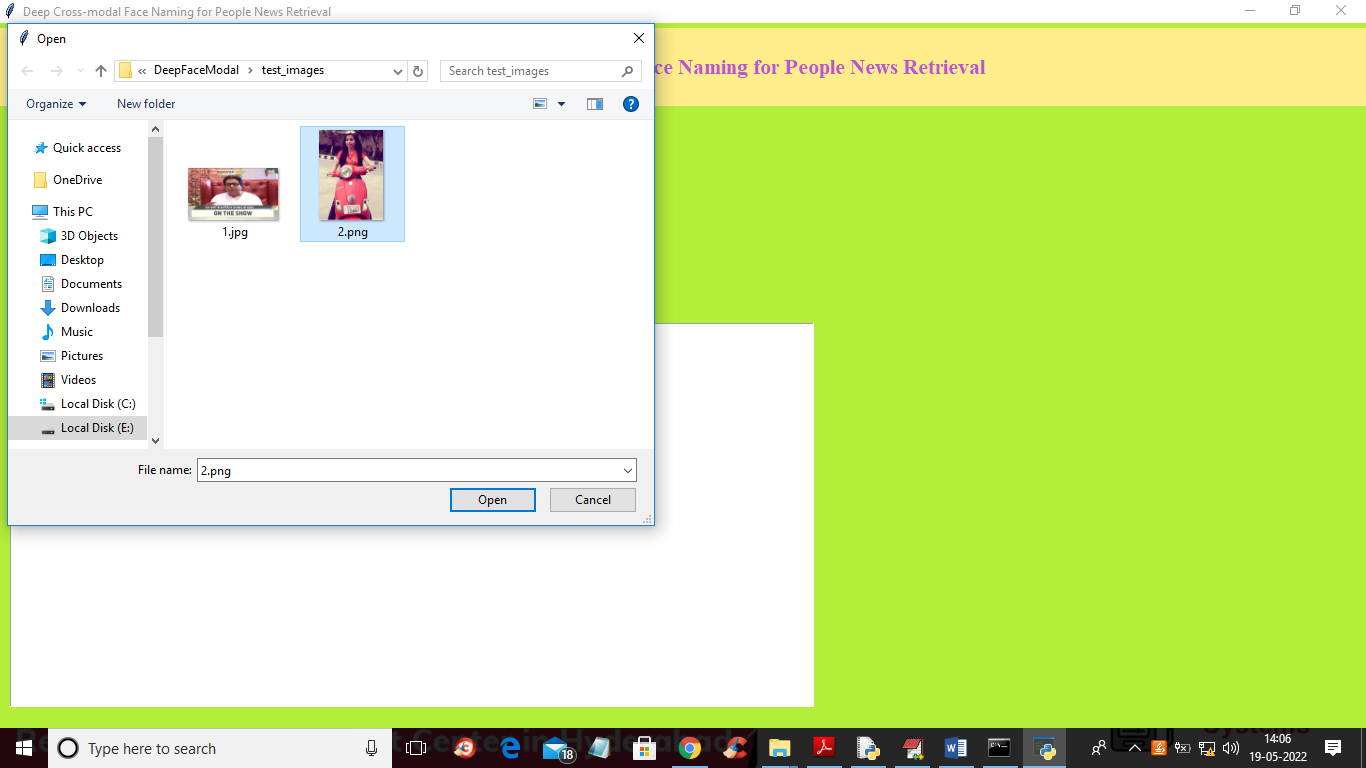
**LOAD DEEP CROSS-MODAL**



**Screenshot 5.2: LOAD CROSS-MODAL**

In above screen in text area we can see model is loaded and now click on ‘Upload Image’ button to upload image and get below output

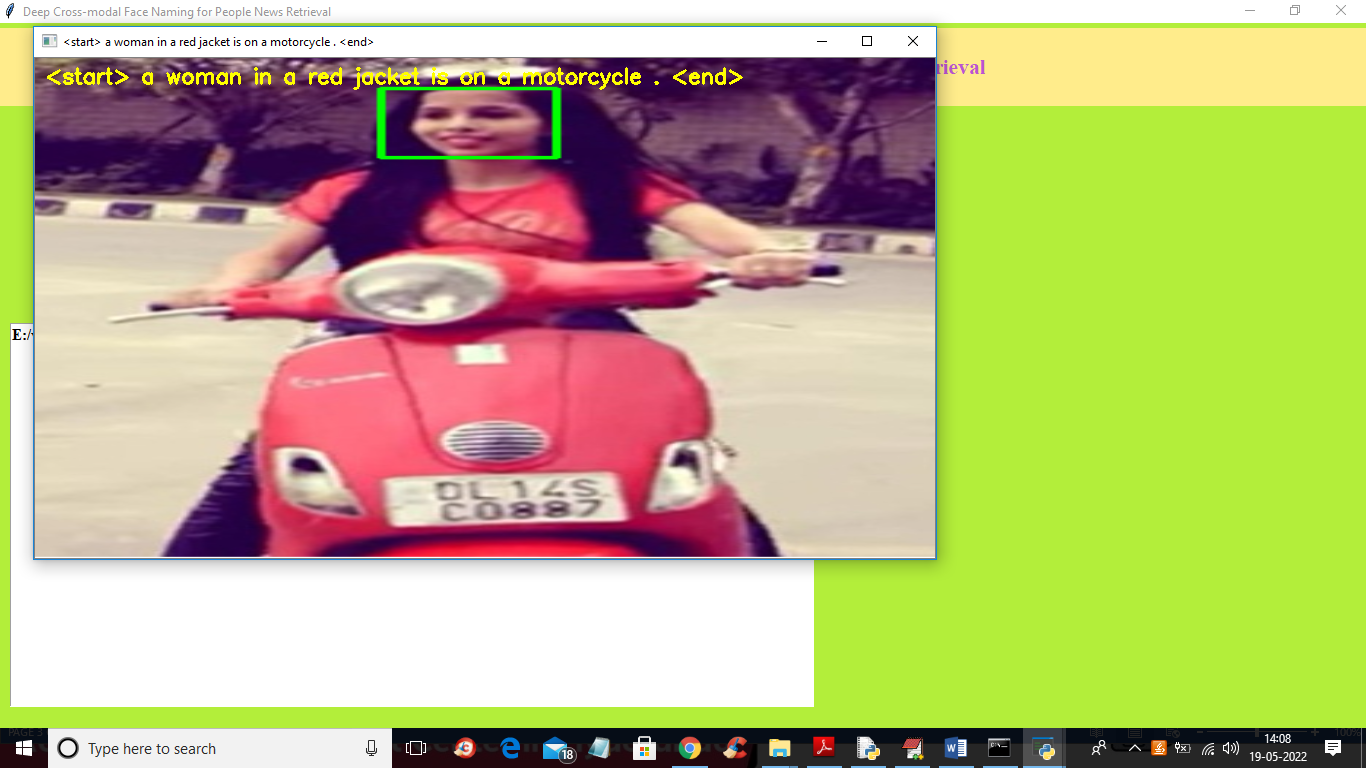
**UPLOAD IMAGE**



**Screenshot 5.3:UPLOAD IMAGE**

In above screen selecting and uploading ‘2.png’ and then click on ‘Open’ button to load image and then click on ‘Caption/News Retrieval’ button to extract caption from image and get below output.

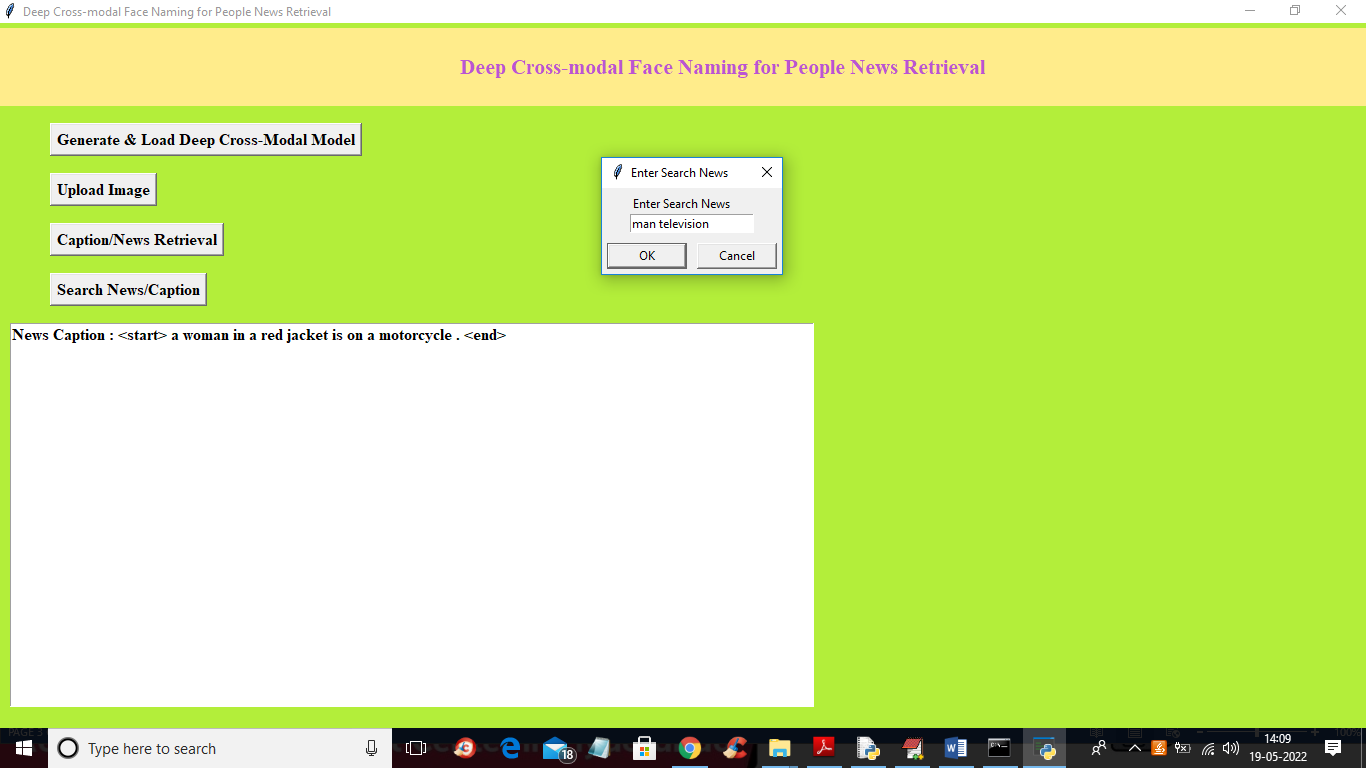
**CAPTION GENERATED**



**Screenshot 5.4: CAPTION GENERATED**

In above screen we got caption associated with image and now to search news with images then we need to give query like below screen

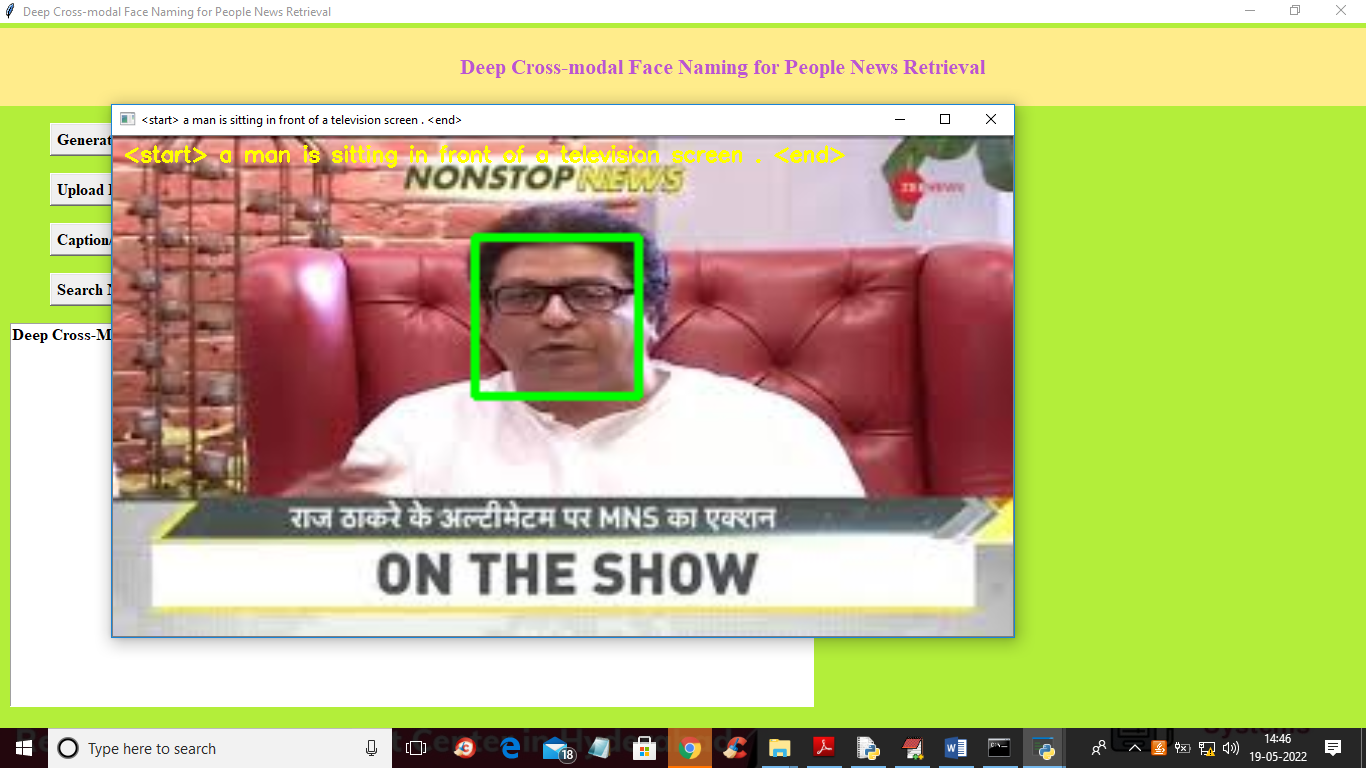
**QUERY**



**Screenshot 5.5: QUERY**

In above screen I gave query as ‘man television’ to get below output

**IMAGE**



**Screenshot 5.6: IMAGE**

In above screen in image we got man and television so by giving query we retrieve related image and similarly you can upload any image to get caption or give query to get related caption image

**Chapter 6**

**TESTING**

**6.1 Introduction**

Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. **6.2 Types of Testing**

**6.2.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**6.2.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**6.2.3 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercise

# **Chapter 7**

**CONCLUSION & FUTURE SCOPE**

**7.1. Conclusion**

The future of face naming powered by Explainable AI (XAI) holds the potential to revolutionize the landscape of facial recognition technology by enhancing transparency, trust, and efficiency in identity verification processes. As developers increasingly adopt AI-driven models for cross-modal face naming, the integration of XAI will allow stakeholders to understand the rationale behind identification decisions, thus mitigating biases and ensuring compliance with ethical standards.This advancement will not only empower developers to create more reliable systems based on clear, interpretable data but also provide end-users with insights into the factors influencing recognition outcomes. Consequently, the adoption of XAI in face naming systems will foster a more equitable environment, where decisions are data-driven and grounded in fairness, ultimately leading to improved user satisfaction and trust in the technology.As technology continues to evolve, ongoing collaboration between AI developers, regulatory bodies, and ethical oversight committees will be crucial in shaping a responsible and effective framework for the future of facial recognition applications.

**7.2 Future Scope**

The future of face naming systems leveraging Explainable AI (XAI) holds significant promise, transforming the landscape of facial recognition technology by enhancing transparency, fairness, and efficiency. As scrutiny around AI systems increases, developers are under pressure to adopt models that not only deliver accurate identification but also provide understandable insights into their decision-making processes.XAI can help demystify algorithms, allowing users to comprehend why certain individuals were recognized or misidentified, fostering trust in the technology. Additionally, integrating XAI into face naming processes can aid in identifying biases in recognition practices, ensuring equitable treatment across diverse populations.The project can also explore real-time monitoring of AI systems to ensure compliance with ethical standards and regulations, facilitating the continuous improvement of algorithms while promoting responsible AI usage.

**REFERENCES**

1. Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning*. Proceedings of the 34th International Conference on Machine Learning.
2. Wang, X., & Wu, X. (2018). *Cross-modal retrieval: A comprehensive survey*. IEEE Transactions on Big Data.
3. Guo, Y., Zhang, L., & Wu, Q. (2016). *Deep Learning for Face Recognition: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
4. Christoph Molnar (2020). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Available online.

**GITHUB LINK**