

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**A Project Report**

**On**

**Building HTTP Server in C++**

**Submitted By:**

Amrit Sapkota (THA079BCT003)

Deep Dhamala (THA079BCT010)

Gyanendra Silwal (THA079BCT012)

Kushal Shah (THA079BCT021)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

November, 2024

### COPYRIGHT

The author has agreed that the Library, Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering and author’s written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head

Department of Electronics and Computer Engineering

Thapathali Campus, Institute of Engineering

Lalitpur, Kathmandu

Nepal

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our project supervisor, Er. Saroj Shakya, for his invaluable guidance and support throughout this project. His expertise was instrumental in shaping our understanding of network programming and web server architecture. Er. Shakya's ability to simplify complex concepts greatly enhanced our learning experience and motivated us to strive for excellence in our work.

We also extend our heartfelt thanks to our peers and faculty members in the Department of Electronics and Computer Engineering for their encouragement and assistance. The collaborative environment in our department allowed us to engage in meaningful discussions that enriched our project. We are grateful for the support from our classmates, whose enthusiasm inspired us to work harder.

Finally, we appreciate our families for their unwavering support and motivation during this endeavor. Their understanding provided us with the strength to overcome challenges and stay focused on our objectives. We dedicate this project to them, as it is a testament to their love and support. We are proud to have completed this project and look forward to applying the knowledge we have gained in our future endeavors.

Amrit Sapkota (THA079BCT003)

Deep Dhamala (THA079BCT010)

Gyanendra Silwal (THA079BCT012)

Kushal Shah (THA079BCT021)

i

**ABSTRACT**

This project focuses on the development of a basic HTTP server using C++. The server is designed to handle HTTP requests and responses, supporting essential methods such as GET and POST, which are fundamental to web communication. By implementing these operations, we aim to create a robust foundation for understanding how web servers function and interact with clients. The implementation utilizes multithreading to efficiently manage multiple client connections, allowing the server to handle concurrent requests without significant delays, thereby enhancing performance and responsiveness. In addition to deepening our understanding of the HTTP protocol, this project provides practical experience in network programming and server design, exploring key aspects of socket programming, connection management, and error handling. The server is capable of serving static files, such as HTML and images, and is designed to be extensible for dynamic content in future iterations, laying the groundwork for more complex web applications. Through this endeavor, we have gained valuable insights into the challenges of building a functional web server and the importance of efficient resource management in networked environments.

*Keywords: C++, HTTP, Server, Multithreading, Network Programming*

ii

**TABLE OF CONTENTS**

[**ACKNOWLEDGEMENT**](#page2)[**i**](#page2)

[**ABSTRACT**](#page3)[**ii**](#page3)

[**List of Figures**](#page5)[**v**](#page5)

[**List of Tables**](#page7)[**vi**](#page7)

[**List of Abbreviations**](#page8)[**vii**](#page8)

[**1. INTRODUCTION**](#page9)[**1**](#page9)

[1.1 Background Introduction](#page9) [1](#page9)

[1.2 Motivation](#page9) [1](#page9)

[1.3 Problem Definition](#page10) [2](#page10)

[1.4 Objectives](#page10) [2](#page10)

[1.5 Project Scope and Applications](#page10) [2](#page10)

[1.5.1 Project Scope](#page10) [2](#page10)

[1.5.2 Application(s)](#page10) [2](#page10)

[**2. LITERATURE REVIEW**](#page11)[**3**](#page11)

[2.1 Facial Feature Detection Techniques](#page11) [3](#page11)

[2.2 Emotion Analysis Advances](#page11) [3](#page11)

[2.3 Advancements in Deep Learning for Image Processing](#page12) [4](#page12)

[2.4 Ethical and Privacy Implications](#page12) [4](#page12)

[2.5 Critique and Relevance to Our Project Motivation](#page13) [5](#page13)

[**3. METHODOLOGY AND SYSTEM DESCRIPTION**](#page14)[**6**](#page14)

[3.1 Image Processing](#page14) [6](#page14)

[3.2 System Architecture](#page14) [6](#page14)

[3.2.1 Facial Feature Detection](#page15) [7](#page15)

[3.2.2 Emotion Analysis](#page16) [8](#page16)

[3.3 Programming Languages and Libraries](#page16) [8](#page16)

[3.4 Testing and Evaluation](#page16) [8](#page16)

iii

[3.5 Integration and Deployment](#page16) [8](#page16)

[3.6 Tools and Environment](#page17) [9](#page17)

[3.6.1 Productivity Tools Used](#page17) [9](#page17)

[3.6.2 Development Tools Used](#page18) [10](#page18)

[**4. RESULT AND ANALYSIS**](#page19)[**11**](#page19)

[5.1 Implementation and Detection Performance](#page19) [11](#page19)

[5.2 Emotion Analysis Accuracy](#page20) [12](#page20)

[5.3 Conclusion from Results and Analysis](#page21) [13](#page21)

[**5. CONCLUSION AND FUTURE ENHANCEMENT**](#page22)[**14**](#page22)

[6.1 Conclusion](#page22) [14](#page22)

[6.2 Limitations](#page22) [14](#page22)

[6.3 Future Enhancement](#page22) [14](#page22)

[**6. APPENDICES**](#page24)[**16**](#page24)

[Appendix A: Gantt Chart](#page24) [16](#page24)

[Appendix B: Code Snippets](#page24) [16](#page24)

[**References**](#page27)[**19**](#page27)

iv

**List of Figures**

[Figure 3 1: General Representation of System](#page14) [6](#page14)

[Figure 3 2: System Architecture Diagram](#page15) [7](#page15)

[Figure 3 3: Haar Cascade Detection Process](#page15) [7](#page15)

[Figure 3 4: Emotion Analysis Process](#page16) [8](#page16)

[Figure 5 1: Facial Features Detection without Glasses](#page19) [11](#page19)

[Figure 5 2: Facial Features Detection with Glasses](#page19) [11](#page19)

[Figure 5 3: Emotion Analysis (I)](#page20) [12](#page20)

[Figure 5 4: Emotion Analysis (II)](#page20) [12](#page20)

[Figure 6 1: Gantt Chart](#page24) [16](#page24)

[Figure 6 2: Main Function](#page24) [16](#page24)

[Figure 6 3: Face Detection Class](#page25) [17](#page25)

[Figure 6 4: Image Capture and Save](#page25) [17](#page25)

[Figure 6 5: Conversion of binary image to base64.](#page26) [18](#page26)

v

**List of Tables**

[Table 5 1: Analysis of Detected Emotion](#page21) [13](#page21)

vi

**List of Abbreviations**

HTTP: Hypertext Transfer Protocol

URL: Uniform Resource Locator

TCP/IP: Transmission Control Protocol/Internet Protocol

I/O: Input/Output

GET: HTTP method for requesting data from a server

POST: HTTP method for sending data to a server

JSON: JavaScript Object Notation

XML: extensible Markup Language

DNS: Domain Name System

vii

**1. INTRODUCTION**

The rapid expansion of the internet and the increasing demand for efficient, scalable, and secure web services have propelled the development of web server technologies. Our project, "Building HTTP Server from Scratch in C++," positions itself at the forefront of this technological advancement, leveraging the robust capabilities of C++ to create a high-performance web server from the ground up. This project aims to delve into the fundamental aspects of web server architecture, focusing on core functionalities such as handling HTTP requests, managing connections, and ensuring data security.

**1.1 Background Introduction**

Facial feature detection and emotion analysis have become pivotal in the evolution of interactive technology. The ability to recognize and interpret facial expressions opens new approach for human-computer interaction. Historically, this field has seen significant developments, from basic image processing techniques to the incorporation of neural networks for more precise emotion recognition. The growing interest in creating more intuitive and responsive systems has fueled research and development in this domain, making it a rich field for exploration and innovation.

**1.2 Motivation**

After learning the concepts of object-oriented programming, we decided to implement the concepts learned in our lectures and practical classes into a working project. As C++ is a widely used programming language, there were a huge number of potential ideas to work on. Instead of just writing code to learn, we sought to solve a real-world problem. Millions of web searches, scanning through GitHub repositories, and YouTube videos helped us find a purpose. As AI is being implemented everywhere and for feasible purposes, we too chose to include it in our project. Our motivation stems from the desire to enhance human-computer interaction by making it more natural and intuitive. Recognizing and responding to user’s emotions can significantly improve the user experience in applications such as virtual meetings, online learning, and interactive gaming. Additionally, the project offers a platform for us to apply theoretical knowledge to solve practical problems, bridging the gap between academia and real-world applications.

1

**1.3 Problem Definition**

The project aims to develop a system that can accurately detect facial features and analyze emotions in real-time from a video feed. The challenge lies in achieving high accuracy and low latency in diverse lighting conditions and various facial orientations. The system must be robust, efficient, and capable of processing live video streams without significant delays.

**1.4 Objectives**

* To create a real-time system for accurate facial feature detection.
* To analyze and interpret emotions based on detected facial expressions.

**1.5 Project Scope and Applications**

**1.5.1 Project Scope**

The project's scope includes developing an efficient algorithm for facial feature detection and emotion analysis that can be applied in real-time video processing. It involves utilizing OpenCV for image processing, creating a user-friendly interface, and ensuring the system's compatibility with different devices and platforms.

**1.5.2 Application(s)**

1. **Security Systems:** Enhancing surveillance and access control systems with facial recognition capabilities.
2. **Interactive Media:** Creating more engaging content by adapting to the viewer's emotional responses.
3. **Educational Tools**: Offering a new dimension in online learning by responding to students' engagement and emotions.
4. **Psychological Research:** Facilitating studies on emotional responses and human-computer interaction.

2

**2. LITERATURE REVIEW**

The exploration of facial feature detection and emotion analysis represents a significant area of interest within the realms of computer vision and artificial intelligence. The development of these technologies has seen a convergence of various methodologies, from traditional image processing techniques to advanced machine learning algorithms. This literature review aims to present a concise yet comprehensive examination of key developments in this field, providing insights into the existing work, methodologies employed, their importance, limitations, and criticisms, thereby linking these aspects to the motivation behind our project.

**2.1 Facial Feature Detection Techniques**

In "Real-time facial feature detection using Conditional Regression Forests," Kazemi and Sullivan [1] introduce an approach leveraging Conditional Regression Forests for rapid and accurate facial feature detection. This method stands out for its efficiency in real-time applications, emphasizing the importance of speed and precision in interactive systems.

* **Methodology:** The technique employs an ensemble of decision trees to regress facial landmark positions directly from a sparse subset of pixel intensities, offering robustness against variations in expression and illumination.
* **Importance:** Its application extends to real-time tracking and analysis in video streams, crucial for dynamic interaction systems.
* **Limitations:** While effective, it relies heavily on initial face detection accuracy and may struggle with extreme poses and occlusions.
* **Criticism:** The reliance on accurate face detection as a precursor for feature detection points to a potential area for improvement, particularly in handling partial facial visibility.

**2.2 Emotion Analysis Advances**

"Deep Convolutional Neural Networks for Emotion Recognition," by Kim et al. [2], showcases the application of deep learning to analyze complex emotional states from facial expressions.

3

* **Methodology:** Utilizing CNN architectures, this work processes facial images to classify emotional expressions, benefiting from the model's ability to learn hierarchical feature representations.
* **Importance:** The approach enhances the understanding of nuanced emotional expressions, contributing to more empathetic human-computer interactions.
* **Limitations:** Deep learning models require substantial training data and computational resources, potentially limiting their accessibility.
* **Criticism:** The model's performance is contingent on the diversity and quality of the training dataset, raising concerns about bias and generalizability.

**2.3 Advancements in Deep Learning for Image Processing**

The revolution in image processing technologies has been significantly driven by advancements in deep learning. A landmark in this journey was the introduction of deep convolutional neural networks (CNNs) by Krizhevsky, Sutskever, and Hinton in 2012 [3], which significantly enhanced image analysis and recognition capabilities. This breakthrough laid the groundwork for the development of sophisticated neural network architectures such as ResNet, Inception, and YOLO (You Only Look Once), each setting new standards in accuracy and processing speed for image recognition tasks. These models have facilitated a leap in computer vision applications, demonstrating the transformative impact of deep learning on image processing technology.

**2.4 Ethical and Privacy Implications**

The widespread adoption of facial recognition technology has raised significant ethical and privacy concerns. Garvie's 2016 study [4] underscores the potential for mass surveillance and privacy erosion, highlighting the urgent need for regulatory frameworks and ethical guidelines to govern technology use in public spaces. In response, the research community has explored privacy-preserving techniques, such as homomorphic encryption and differential privacy, aiming to safeguard individual data while maintaining the utility of facial recognition technologies. These efforts emphasize the critical balance between technological advancement and the protection of privacy rights, advocating for responsible deployment and usage of these powerful tools.

4

**2.5 Critique and Relevance to Our Project Motivation**

The reviewed works underscore the rapid advancements in facial feature detection and emotion analysis, highlighting the shift towards more sophisticated and computationally intensive models. However, the dependence on extensive datasets for training deep learning models and the challenges in processing under varied conditions reveal critical limitations. These insights directly inform our project's motivation to develop a system that is both efficient and robust, capable of operating in real-time with minimal computational demands. By addressing these limitations, our project seeks to contribute to this field by applying object-oriented programming (OOP) principles to enhance system design and implementation efficiency and aims to make significant contributions to the field, enhancing the practicality and applicability of facial feature detection and emotion analysis technologies in everyday applications.

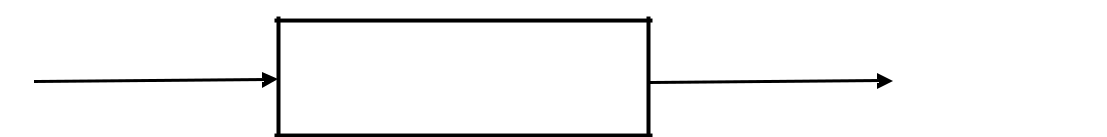
5

**3. METHODOLOGY AND SYSTEM DESCRIPTION**

Our project's methodology integrates computer vision techniques and machine learning algorithms to develop a real-time system for facial feature detection and emotion analysis. The process is defined into distinct phases, each contributing to the system's overall functionality.

**3.1 Image Processing**

Image processing is a field that involves manipulating and analyzing images to extract useful information or enhance their visual quality. In the context of our project, focused on face features detection, and emotion analysis, image processing plays a crucial role in transforming raw visual data into meaningful insights.



Input Image

Image Processing

System

Output Data (Emotion, etc.)

Figure 3 1: General Representation of System

**3.2 System Architecture**

The system architecture is designed to seamlessly integrate facial detection with emotion analysis in a real-time application. This integration is facilitated using OpenCV for image processing.

The system architecture depicted in Figure 3-2 illustrates a streamlined process for real-time facial feature detection and subsequent emotion analysis. Using the robust image processing capabilities of OpenCV, the system initially loads a pre-trained cascade classifier to identify facial features within a live video feed. Detected features such as the eyes, face, and smile are then processed and subjected to an API call, which communicates with a local server equipped to analyze emotions. The server responds with the identified emotion and its confidence score, which are then displayed alongside the video feed. Additionally, the system is designed to handle user interactions, such as an 'On Click' event, and incorporates local storage to manage the image data efficiently. This architecture ensures a cohesive operation, from image acquisition to emotion display, facilitating an intuitive user experience.

6

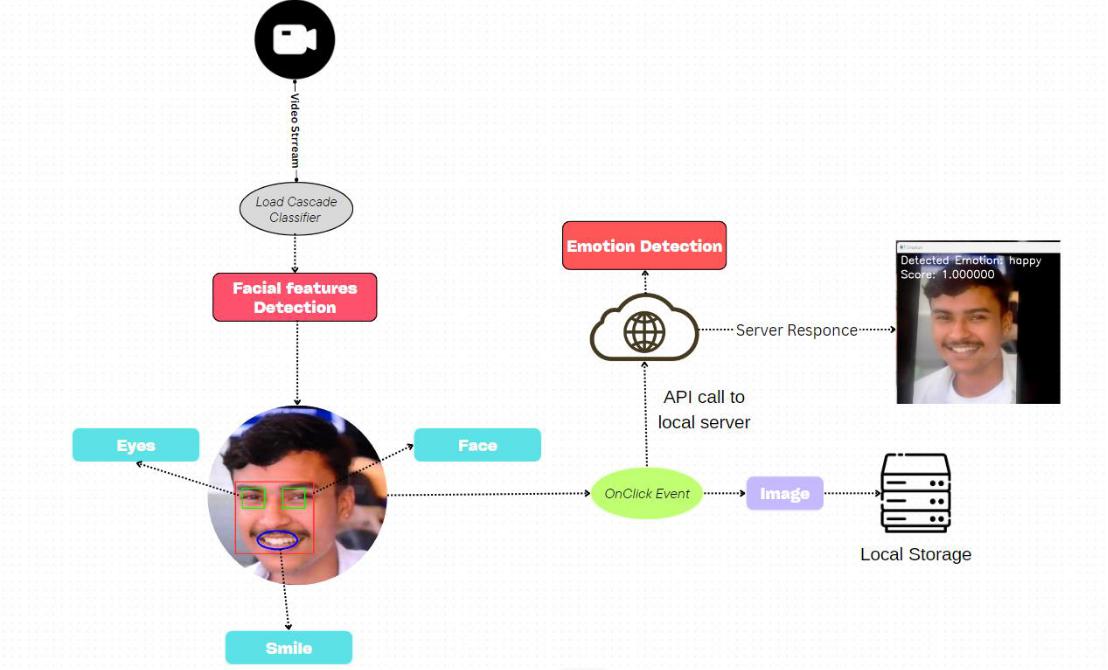
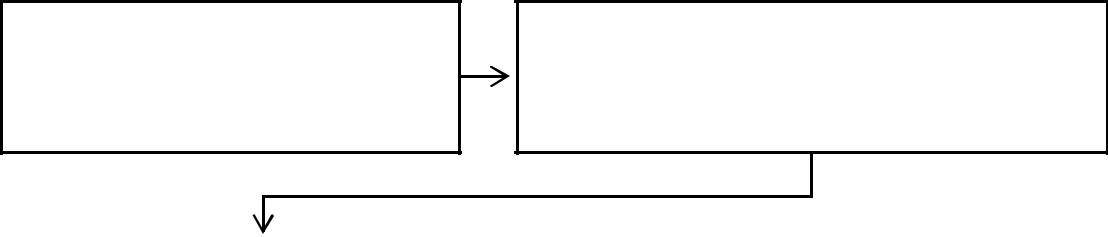


Figure 3 2: System Architecture Diagram

**3.2.1 Facial Feature Detection**

Facial feature detection is the first critical step in our methodology, leveraging Haar cascade classifiers provided by OpenCV. This process involves capturing live video feed from the camera, converting the feed into grayscale (to reduce computational complexity), and applying Haar cascades to detect facial features.



Capture live video feed using

OpenCV



Convert each frame into grayscale

Apply Haar cascade classifiers to

identify facial regions.

Mark detected features with

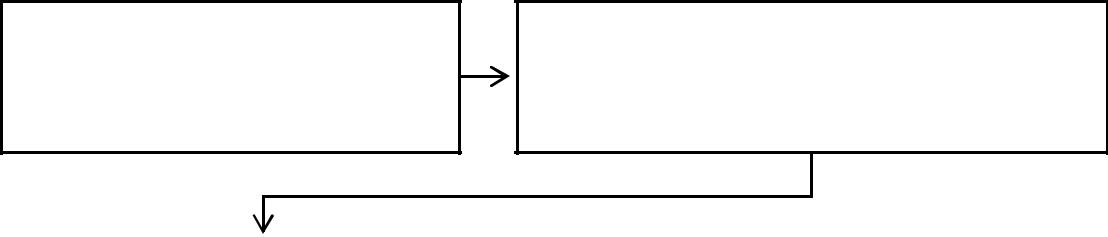
bounding boxes for further analysis.

Figure 3 3: Haar Cascade Detection Process

7

**3.2.2 Emotion Analysis**

Upon successful detection of facial features using Haar cascades, the system transitions to the emotion analysis phase. This phase involves processing the detected facial regions through a pre-trained convolutional neural network (CNN) model to classify the emotion exhibited by the facial expressions. The methodology for this phase encompasses the following steps:



Crop Detected Facial Regions



Preprocess Cropped Images

Emotion Prediction with Pre-trained

CNN Model

Display Predicted Emotion

Figure 3 4: Emotion Analysis Process

**3.3 Programming Languages and Libraries**

* **C++:** Using C++ for its efficiency and performance, especially in real-time applications.
* **OpenCV:** Leveraging the OpenCV library for its rich set of tools and functions tailored for computer vision and image processing. [5]

**3.4 Testing and Evaluation**

In our project, testing and evaluation were primarily focused on the visual verification of the system's ability to detect facial features and analyze emotions in real-time, rather than relying on traditional dataset-driven methods. This approach allowed us to assess the system's performance in dynamic, real-world conditions where variability in facial expressions, lighting, and angles can be more challenging than static dataset images.

**3.5 Integration and Deployment**

* **System Integration:** Combining the various modules (face detection, features detection, emotion analysis) into a unit system.

8

* **Real-World Testing:** Evaluating the system's performance in real-world scenarios to validate its effectiveness. Feedback from third-party testers was instrumental in identifying strengths and potential areas for improvement. Testers were asked to interact with the system.

**3.6 Tools and Environment**

**3.6.1 Productivity Tools Used**

**Jira:** Jira served as a pivotal tool for our project, providing a centralized platform for task organization, project progress tracking, and fostering seamless team collaboration. With Jira's versatile features, our team efficiently managed the complexities of the Smart Camera project.

The Jira board structure allowed us to visually represent our workflow, providing a clear overview of tasks through issues and customizable workflows. The breakdown of tasks into epics and user stories facilitated the organization of project components. Features such as checklists were instrumental in dissecting intricate tasks into more manageable subtasks, promoting a systematic approach to development.

Deadlines and sprints in Jira played a crucial role in ensuring the timely completion of project milestones. By setting and tracking deadlines, we maintained a focused and time-sensitive development process. The agile methodologies supported by Jira, including Scrum and Kanban, allowed us to adapt to evolving project requirements efficiently.

The use of labels in Jira allowed us to categorize and prioritize tasks effectively. This feature enabled the team to identify critical aspects of the project and allocate resources accordingly. Through the Jira dashboard, we could easily monitor progress, identify bottlenecks, and make data-driven decisions to keep the project on track.

**Google Meet and Microsoft Word:** For virtual meetings and collaborative document editing, we leveraged Google Meet and Microsoft Word. Google Meet provided a seamless platform for remote discussions, enabling real-time interaction and project coordination. Microsoft Word, on the other hand, served as our go-to tool for collaborative document creation and editing, ensuring a centralized and accessible repository for our project proposal.

9

**3.6.2 Development Tools Used**

**Git & GitHub:** GitHub played a pivotal role in our development process, offering robust version control and collaboration capabilities. The use of pull requests streamlined the code review process, facilitating efficient collaboration among team members. Issue tracking on GitHub helped us identify, prioritize, and resolve issues effectively, ensuring a smoother development workflow. With Git, we maintained a reliable history of our codebase, facilitating collaboration, bug tracking, and the implementation of new features in a well-organized manner.

**VS Code Editor:** Visual Studio Code (VS Code) emerged as a powerful code editor that significantly enhanced our coding efficiency. Features like debugging tools allowed us to identify and address issues swiftly. Syntax highlighting and code completion features improved code readability and reduced the likelihood of errors. The versatility of VS Code, with its support for various programming languages and extensions, contributed to a seamless development experience, making it an integral part of our coding environment.

**GCC:** The GNU Compiler Collection (GCC) played a crucial role in compiling our C and C++ code. It ensured compatibility and optimized the performance of our algorithms. By using GCC, we could generate executable files from our source code, making our software runnable on different platforms. This compiler's efficiency contributed to the overall functionality and performance of our image processing algorithms in the Smart Camera project.

**CMake:** While CMake is primarily a build system, it is an integral part of the development toolchain. It significantly contributes to the efficiency and portability of the development process, ensuring that the project can be built consistently across different environments. Including CMake in the development toolset is recommended, especially when working on C++ projects where build configuration and dependency management are critical aspects of the development process.

10

**4. RESULT AND ANALYSIS**

This section presents the outcomes of implementing the real-time facial feature detection and emotion analysis system. Our project leverages Haar cascade classifiers within the OpenCV framework to detect facial features in live video feeds and applies pre-trained models to analyze emotions based on these detections.

**5.1 Implementation and Detection Performance**

Our system successfully utilizes OpenCV and Haar cascade classifiers to open the camera, capture images, and detect facial features in real time. The integration of these classifiers into our application allows for efficient identification of key facial regions, crucial for subsequent emotion analysis.

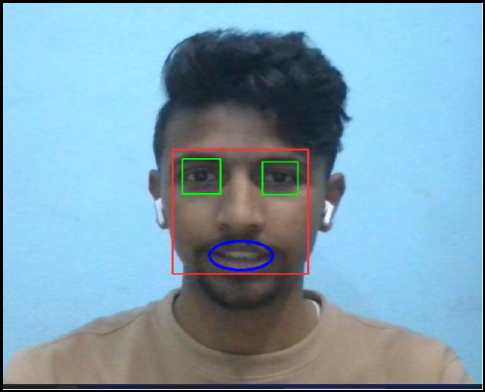


Figure 5 1: Facial Features Detection without Glasses

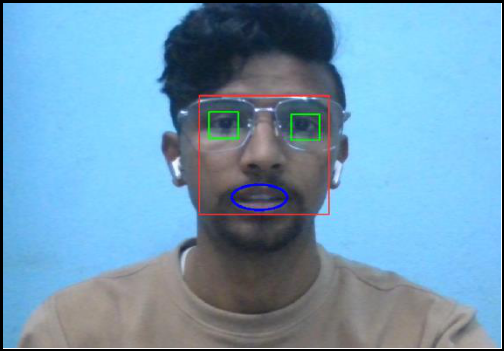


Figure 5 2: Facial Features Detection with Glasses

11

The real-time performance of our system demonstrates the practical applicability of Haar cascades for facial feature detection in live video feeds. Despite the reliance on pre-trained models, the system exhibits a commendable ability to identify facial features promptly, ensuring a seamless user experience.

**5.2 Emotion Analysis Accuracy**

The emotion analysis component of our system, powered by pre-trained models, categorizes detected emotions with a notable degree of accuracy. The accuracy rates for emotion recognition are as follows:

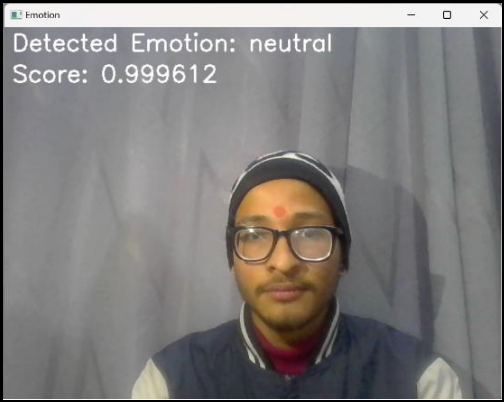


Figure 5 3: Emotion Analysis (I)

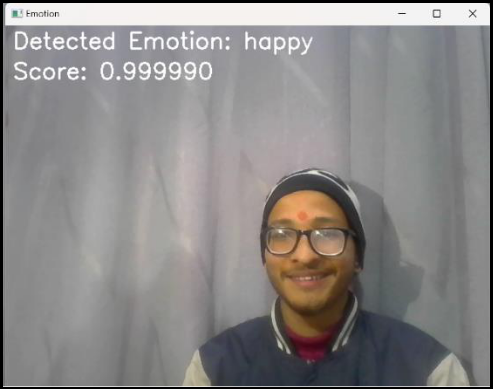


Figure 5 4: Emotion Analysis (II)

12

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy Rate** | |  |
| **Emotion** |  |  |  |
| **Image 1** | **Image 2** |  |
|  |  |
|  |  |  |  |
| Neutral | 0.9996123909950256 | 0.0000065115973484 |  |
|  |  |  |  |
| Happy | 0.0002212307881563 | 0.9999897480010986 |  |
|  |  |  |  |
| Sad | 0.0000294023102469 | 1.6416592529822083 |  |
|  |  |  |  |
| Angry | 0.0000297436181426 | 2.584645528713736 |  |
|  |  |  |  |
| Fearful | 1.9957363406319928 | 4.314558665896584 |  |
|  |  |  |  |
| Disgusted | 0.0000051362644626 | 2.316395040224961 |  |
|  |  |  |  |
| Surprised | 0.0001019060582621 | 0.000003158688969 |  |
|  |  |  |  |

Table 5 1: Analysis of Detected Emotion

**5.3 Conclusion from Results and Analysis**

The results underscore the potential of leveraging Haar cascades combined with pre-trained models for real-time facial feature detection and emotion analysis. Our project highlights the efficiency and applicability of these technologies in creating interactive applications that can operate in diverse environments. The system's performance, particularly in emotion analysis accuracy, showcases the capabilities of existing models to provide meaningful insights into human emotions in real time.

13

**5. CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 Conclusion**

This project has successfully demonstrated the implementation of a real-time facial feature detection and emotion analysis system using object-oriented programming (OOP) principles, leveraging the capabilities of C++ and the OpenCV library. Through the development and testing phases, we have addressed key challenges in accurately detecting facial features and analyzing emotions in real-time, highlighting the system's potential in various applications from security to interactive media. Our approach has shown that combining advanced image processing algorithms with OOP principles can result in a robust and efficient system capable of operating in diverse environments and conditions. The project not only achieved its primary objectives but also provided insights into the practical applications and implications of computer vision technologies.

**6.2 Limitations**

* **Lighting Conditions:** The system's performance can vary under different lighting conditions, affecting the accuracy of facial feature detection and emotion analysis.
* **Facial Occlusions:** Obstructions such as glasses or face coverings can hinder the system's ability to detect facial features accurately.
* **Real-time Processing:** While efficient, there are still challenges in achieving ultra-low latency during real-time video analysis, particularly on lower-end hardware.
* **Data Diversity:** The training data used for emotion analysis may not fully represent the diversity of facial expressions across different cultures and demographics, potentially affecting the system's universality.

**6.3 Future Enhancement**

* **Development of Custom Models:** A key direction for future work involves developing our proprietary deep learning models for facial feature detection and emotion analysis. By training these models on a diverse and expansive dataset, we aim to achieve higher accuracy and robustness compared to the currently utilized pre-trained models.

14

* **Enhancing Model Accuracy:** Continuous efforts will be dedicated to refining the accuracy of our models through advanced training techniques, including transfer learning and data augmentation. This will ensure that our system remains effective across a broad spectrum of facial expressions and environmental conditions.
* **Real-world Application Integration:** Future enhancements will focus on integrating our system into practical applications, such as interactive advertising, personalized content delivery, and enhanced security systems. This involves developing APIs and plugins that facilitate easy integration with existing digital platforms and services.
* **Robustness Against Occlusions and Lighting Variability:** We plan to tackle the challenges posed by facial occlusions and varying lighting conditions through the implementation of more sophisticated image preprocessing and augmentation techniques. This will enable our system to maintain high performance even in suboptimal conditions.
* **Custom Model Training:** In response to the limitations of pre-trained models, future efforts will include the development and training of our models tailored specifically to the nuances of our application requirements. This will allow for finer control over the features and characteristics the models prioritize, potentially leading to significant improvements in system performance.

15

1. **APPENDICES Appendix A: Gantt Chart**



Figure 6 1: Gantt Chart

**Appendix B: Code Snippets**

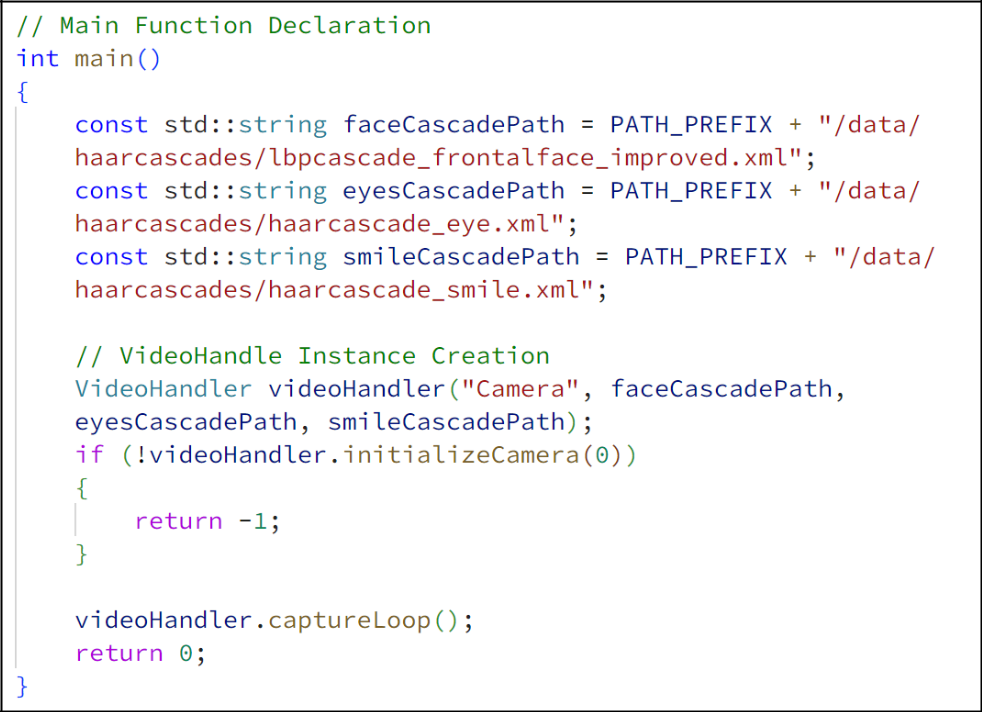


Figure 6 2: Main Function

16



Figure 6 3: Face Detection Class

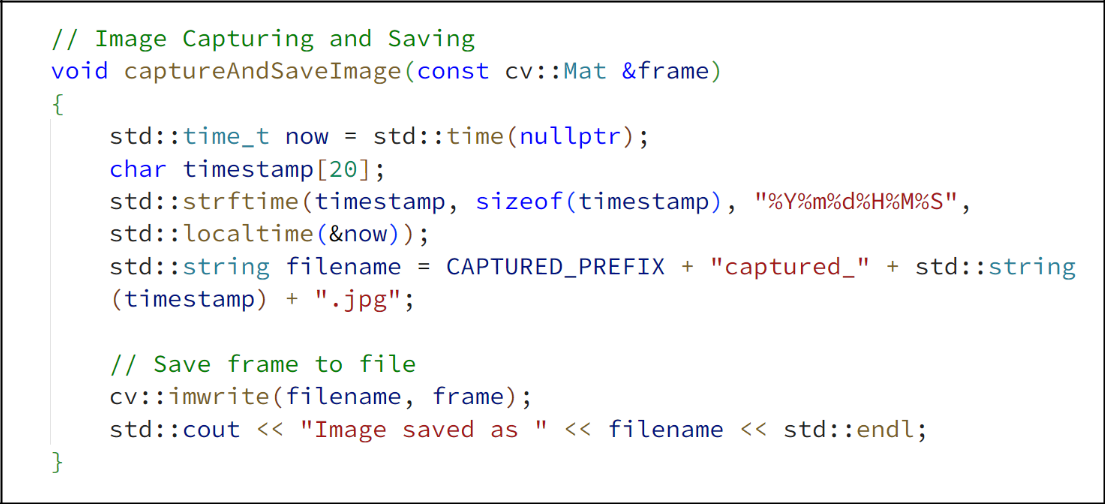


Figure 6 4: Image Capture and Save

17

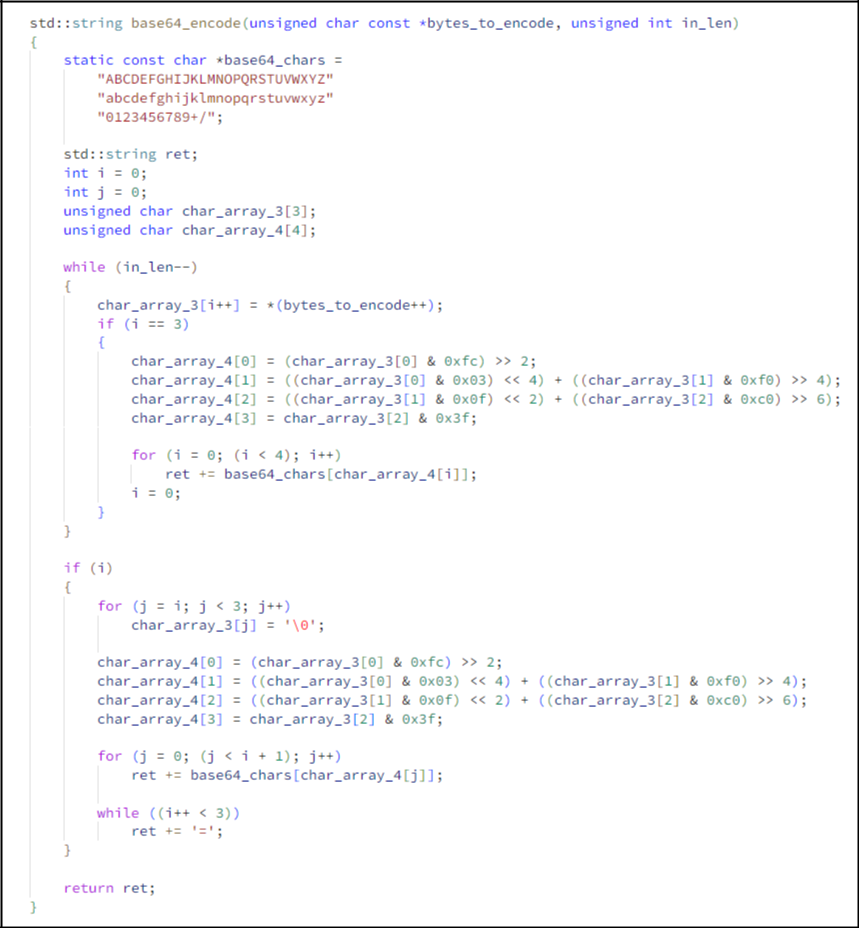


Figure 6 5: Conversion of binary image to base64.

18

**References**

1. Kazemi, V., & Sullivan, J. (2014). "Real-time facial feature detection using Conditional Regression Forests." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Kim, E., Lee, H., & Provost, E. M. (2016). "Deep Convolutional Neural Networks for Emotion Recognition." IEEE Transactions on Affective Computing.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012.
4. C. Garvie, "The Perpetual Line-Up: Unregulated Police Face Recognition in America," Georgetown Law Privacy & Technology Center, 2016.
5. OpenCV Contributors, "OpenCV: Open-Source Computer Vision Library," 2021. [Online]. Available: https://opencv.org/

19