FACIAL STRUCTURE AND EMOTION ANALYSES

Chris Samuel Salcedo
2022-05055

In partial fulfillment of the requirements of CMSC 191: Image Processing.

Git Repository: https://github.com/chrisracha/face-analyzer
Demo Video: bit.ly/191FA-Demo

ABSTRACT

This project is an interactive desktop application developed using Python and various computer vision techniques to perform real-time analysis of facial features and emotions from a webcam feed. The program utilizes dlib for face landmark localization, OpenCV for image handling, and overlay/AR, as well as TensorFlow/Keras with an important Convolutional Neural Network (CNN) model for emotion recognition to conduct geometric analyses to classify face, lip, nose, and eye shapes with a Graphical User Interface (GUI) built with PyQt5. This project aimed to provide a comprehensive application of multiple computer vision concepts learned in CMSC 191 to a practical application.

Objectives

1. Develop a desktop application with a GUI for real-time computer vision analysis;
2. Apply at least three distinct computer vision applications learned in CMSC 191; and
3. Implement real-time processing of web camera input to analyze facial features and extract emotions based on facial landmarking, specifically:
    a. Use dlib landmark detection to perform geometric analysis to classify face, lip, nose, and eye shapes;

b. Deploy a pre-trained CNN model for real-time emotion recognition; and
c. Provide a user-friendly experience through a UI that allows toggling for visualizing different analysis results in the video feed.

Project Features

Real-time Webcam Input. The program allows real-time input from the user's device's web camera.

Face Detection. The program deploys dlib's pre-trained face detector to identify faces in the video feed.

Facial Landmark Detection. The program, through dlib, identifies 68 key facial landmarks.

Facial Feature Shape Analysis. Classifies the detected facial features into various shapes using geometric calculations, including primary features such as the lip, nose, and eye.

Real-time Emotion Recognition. The program analyzes the expression of the detected face using a pre-trained CNN model, classifying the face into 6 distinct facial features: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

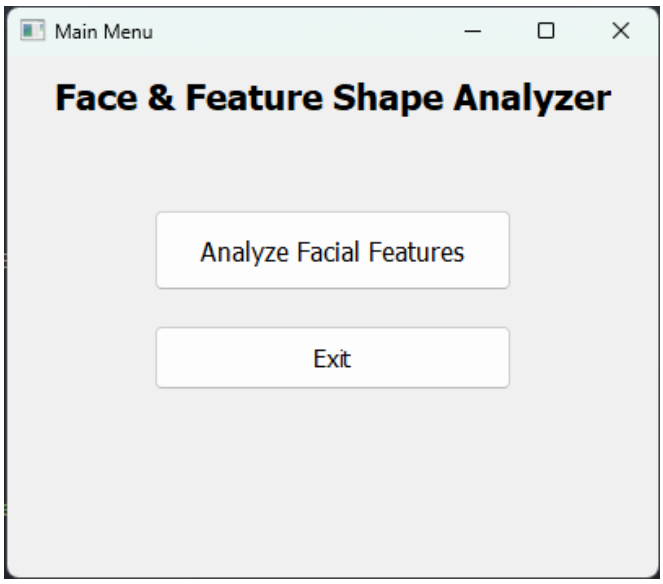
PyQt5 GUI. Provides an interactive graphical interface with a live video feed display, with checkboxes that allow users to select which visual overlays on the video feed, such as outlines and labels.

Relevant Techniques Used

- Python + OpenCV Image Handling
- Face and Facial Landmark Detection
- Neutral Networks (CNN) using TensorFlow/Keras
- Real-time Webcam Feed Processing

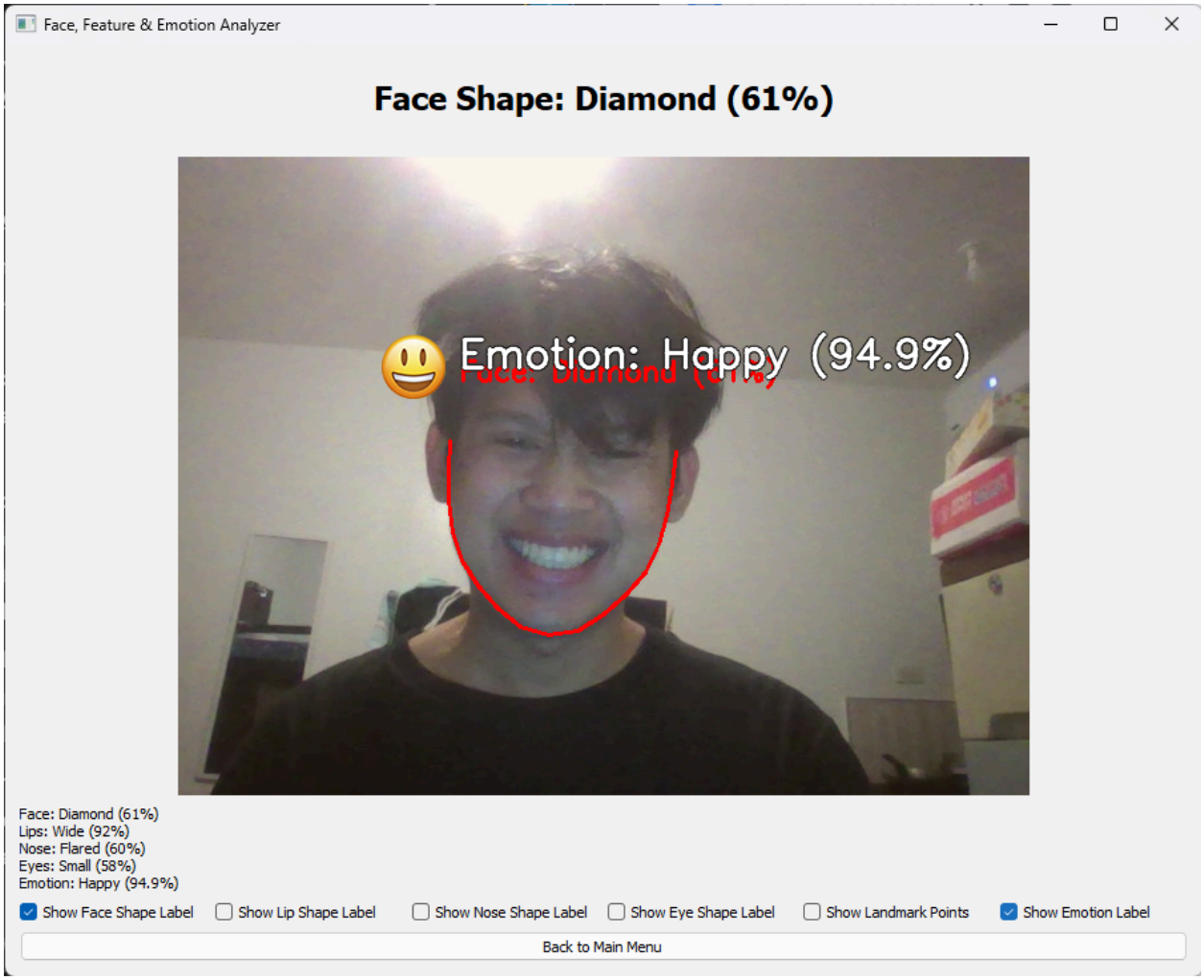
- Augmented Reality (AR) Functionality

SCREENSHOTS & WORKFLOW



Main menu window

Upon launching, the main menu window appears with options to “Analyze Facial Features” or “Exit”; clicking the first option opens the main analysis window and starts the webcam feed, which is then continuously captured by the program. If a face is detected, landmark detection is performed on the face ROI to calculate geometric features and classify, lips, nose, and eye shapes. Based on the enabled checkboxes, the application draws outlines of the specified feature and displays text for the detected face shape, feature shapes, and emotion on the feed. Clicking the “back to main menu” button hides the analysis window and returns to the main menu, which then, if followed by clicking the “Exit” button in the main menu, terminates the application.



Main analysis window

CHALLENGES AND LIMITATIONS

Developing this program has presented several challenges and has some limitations, which are as follows:

1. **Accuracy of Geometric Feature Analysis.** The methods deployed to classify facial features are based on geometric heuristics that came from landmark ratios and proportions, which are sensitive to head pose, facial expressions, lighting conditions, and individual variations that affect the accuracy scores of the classifications.
2. **Dependencies.** The program relies on a pre-trained dlib landmark prediction model as well as a pre-trained TensorFlow/Keras emotion model, which must always be present in the proper directory so that it works properly.
3. **Generalized Emotion Model**. The model's emotion detection may vary depending on the diversity of the dataset it was trained on, which may imply limitations on its application in different demographics, lighting, or with occlusions.
4. **Performance:** The operations deployed in the program are computationally intensive, which may affect user experience. Additionally, varying webcam resolution and powerful processing may impact the performance of the program, too.
5. **Single Face Processing:** The program cannot handle multiple faces simultaneously.

Future Enhancements

Several enhancements could be implemented to improve the program specifically:

1. Training machine learning models on datasets of labeled facial features; and
2. Implementation of analysis for more facial features;

3. Implementation of analysis for age and gender predictions;
4. Use of more advanced or larger emotion recognition models;
5. Extension to detect, track, and analyze multiple faces in the frame;
6. Use of more optimized face detections, landmarking, and GPU acceleration;
7. More controlled controls, such as for confidence thresholds, improve layout, and potentially, a “capture screenshot” button;
8. Implementation of logging to save results and images; and
9. Provide options for users to load different emotion models, adjust parameters for the geometric analysis.

REFLECTION

I've learned a lot in this course. I used to think that Image Processing is a highly complex process that requires very heavy technical skills to achieve. However, throughout the short months of 5 months, I've learned that, although it is still complex to some extent, it can be learned and I shouldn't be intimidated by its complexity. Before I was a computer science student, I was a graphic designer first, and it always fascinated me how things really work behind the scenes, and learning about this in this course made it more interesting for me. It was also fun learning, especially that there are accessible libraries out there, like OpenCV and TensorFlow, to work with real-time video streams to create your own image processing applications

Developing the application wasn't easy; the moment I tried to integrate all the libraries I needed proved to be a non-trivial task, as it required strategic dependency management and troubleshooting installation and DLL errors. Integrating these, getting dlib to seamlessly feed data into OpenCV for drawing and connecting the PyQT5 GUI, and tangling around the correct format of the facial ROIs for the TensorFlow CNN Model required a lot of patience and coordination to understand how everything works and how do I augment each of them. These challenges, however, led to valuable insights and challenged me to apply my theoretical knowledge to a practical problem and reflect on the capabilities of developing computer vision systems.

References

dlib: http://dlib.net/

OpenCV: https://opencv.org/

PyQt5: https://riverbankcomputing.com/software/pyqt/

TensorFlow/Keras: https://www.tensorflow.org/

King, D. E. (n.d.). *shape_predictor_68_face_landmarks.dat*. Dlib.net. Retrieved from http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2

Oarriaga, E. (n.d.). *emotions_model.hdf5*. GitHub: oarriaga/face_classification. Retrieved from http://github.com/oarriaga/face_classification/tree/master/trained_models/emotion_models

Borman, T. M., & Campbell, R. M. (2010). Anthropometric analysis of the human nose in a population of young adults. *Journal of Craniofacial Surgery*, 21(4), 1207-1210. https://doi.org/10.1097/SCS.0b013e3181e1c7e4

Ercan, I., Ozdemir, S. T., Etoz, A., et al. (2008). Reliability of facial soft tissue measurements on 3D images. *European Journal of Orthodontics*, 30(6), 649-656. https://doi.org/10.1093/ejo/cjn070

Ferrario, V. F., Sforza, C., Miani, A., et al. (1997). Three-dimensional analysis of lip form in adult humans. *Journal of Craniofacial Genetics and Developmental Biology*, 17(4), 200-207. https://pubmed.ncbi.nlm.nih.gov/9431532/

Satina, Armacheska M. (CMSC 191) Laboratory Resources. 1, 1.1, 1.2, 1.3, 1.5, 2, 2.6, 6.4, 7.2. Lecture 6.2. Lecture 12.

Sforza, C., Grandi, G., Catti, F., et al. (2010). Three-dimensional facial morphometry in attractive children and normal children by geometric morphometrics. *Angle Orthodontist*, 80(1), 6-13. https://doi.org/10.2319/010509-7.1

Xie, Y., & Lam, K. M. (2015). Automatic Face Shape Classification Using Geometric Features. *Pattern Recognition Letters*, 68, 31-37. https://doi.org/10.1016/j.patrec.2015.09.014