



Student Name: Javohirmirzo Kosimov

Student ID: TM221-310  
Student Number: 24835743

Cover Page

Title: Face Detection in Python Using Deep Learning  
Author: Javohirmirzo Kosimov  
Course: [Advanced AI and Applications]  
Institution: [TMCi institute]  
Submission Date: [18/05/2025]

**Introduction**

Face detection is a core technique that finds extensive use in many areas of computer vision. The technology finds relevance in a wide array of use cases like facial recognition, emotion analysis, identity verification, video surveillance and human-computer interaction. State-of-the-art deep learning methods have led to the development of face detection systems that are able to overcome challenges such as occlusion, varying image quality or human faces covered by masks with high accuracy.

This project involves designing a deep learning system that can effectively detect faces and analyze the associated emotions of individuals, regardless of whether or not they're wearing masks. it's built using Python programming language along with popular libraries such as TensorFlow, Keras, OpenCV and Streamlit.

Both face detection and emotion classification are essential features of this project because they serve applications in fields such as mental health monitoring, customer analysis, and intelligent tutoring systems. The model is trained using data sets of masked faces divided into seven different categories. Angry, happy, fear, sad, neutral, disgusted and surprised. we've created a custom CNN in Keras and fine-tuned it to obtain excellent performance on both facial emotion datasets with or without masks.

Designing and training the model makes use of cnn\_emotion\_model.py while evaluation is performed in evaluate\_model.py. Automated data preprocessing steps, such as masking and organization of the dataset, are implemented in main.py. Streamlit-based interface in streamlit\_demo.py enables users to easily access and use the model with either their webcam or by uploading images.

The project emerged from the growing demand for face-aware systems in a world where wearing masks is prevalent following the pandemic. Most conventional face recognition or detection algorithms are challenged by masked faces, which calls for models trained explicitly on that type of data. This project directly overcomes that difficulty by incorporating simultaneous mask detection and masked emotion recognition into a unified system.

Thanks to its modular design, the entire system can be easily extended with functionality such as facial landmark detection, gender recognition and attention monitoring. This project provides a solid core for developing advanced face analytics systems.

The project effectively showcases how deep learning can be used effectively for face detection as well as how Python and open-source tools make it possible to develop robust, easy-to-use, and intelligent solutions. The rest of this report will explore the issue context, model design, training approach, performance assessment, extra features and lessons learned throughout the development process.

**Problem Analysis and Background Research**

Detecting faces and deciphering emotional states has been a major focus of research in the field of computer vision during the last two decades. Progress in computer hardware and neural network advances has enabled these systems to perform more accurately using modern deep learning methods.

The Problem

Face detection approaches utilizing Haar cascades and HOG features can work well when faces are evenly lit and the head is turned directly toward the camera. These techniques frequently fail in the presence of unusual circumstances such as changes in face angle, lighting, facial hair or obstructions such as glasses or masks.

Since the onset of the COVID-19 crisis, face masks have significantly increased the difficulty of processing face-related tasks. The presence of masks occluding much of the lower face is especially difficult for traditional detection and classification methods since they largely depend on the lower area of the face for emotion recognition. The main aim of this project is to solve the problem caused by this situation. Developing methods capable of accurately classifying facial expressions under different levels of face mask occlusion.

Many kinds of real-time applications demand fast, reliable and memory-efficient face detection systems. As a result, developing AI models grounded in the field of deep learning and specifically in the use of CNNs plays a crucial role when it comes to solving image recognition and recognition tasks.

**Research Background**

Much research has proven that CNNs excel at identifying and understanding facial expressions. In various studies, models based on these networks have obtained impressive results when identifying emotions such as joy, sadness and anger on the faces of people in images. They are capable of extracting more effective representations from visual input than those designed by humans in traditional methods.

Reducing errors in detecting emotions on faces obscured by surgical or cloth masks has become a major focus in recent research. Recognizing masked expressions has sparked the development of new datasets that incorporate faces with surgical or cloth masks. Architectures designed for analyzing emotions have started concentrating more on the regions around the eyes and eyebrows. Several methods, including transfer learning, attention mechanisms and facial landmark localization, have been investigated to overcome the challenge.

## We build a custom CNN that handles both masked and unmasked facial data while distinguishing between different emotional expressions. This hybrid approach aims to accommodate the changing masked scenarios while delivering powerful results.

## Moreover, this work pertains to many practical industrial scenarios. Emotion detection is becoming more common in e-learning platforms, customer service AI, security systems and healthcare interfaces because timely identification of emotions benefits these applications.

## Objective Relevance

## This project offers a targeted solution to a pressing and current challenge in the field of computer vision by overcoming the weaknesses existing approaches encounter in situations involving masked images.

## **Building the Deep Learning Network**

We constructed the model using the Keras deep learning library, with TensorFlow as a backend and adapted a modular workflow integrating iterative steps. The objective was to design a network that could recognize expressions effectively, despite them usually being partially covered by a mask.

Dataset

Seven emotions were represented in the dataset by images labeled as angry, happy, fear, neutral, sad, disgusted and surprised. Angry, happy, fearful, neutral, sad, disgusted and surprised. The images were divided among the masked\_train and masked\_test subdirectories, each including masked instances of facial expressions.

Masks were applied to the original dataset using an implementation written in main.py using OpenCV. Using realistic images with partial face masks helped the network deal better with scenarios involving COVID-era facial images.

**Network Structure**

A sequential model structure was implemented using the Keras API. The core structure included:

• Multiple Conv2D layers for feature extraction

• ReLU activation functions for non-linearity

• MaxPooling layers for downsampling

• Dropout layers to prevent overfitting

A fully connected layer was added to perform the final prediction for the corresponding emotion.

Dense layer with 7 nodes producing a single value during each prediction (one class prediction per emotion).

A carefully designed architecture was selected to achieve both good results and fast training times.

Loss Function and Optimizer

A categorical cross-entropy loss function was chosen because the problem is a multi-class classification problem. Adam was chosen since it automatically updates the learning rates, reducing training time and promoting better results.

Training and Evaluation

A real-time image augmentation strategy (including rotation, zoom and flip) was applied to enhance generalization during training. To ensure the preservation of the best model performance, a checkpoint callback was implemented. Evaluation revealed reliable and dependable performance over the whole masked test dataset.

## **Testing**

To assess the model’s performance, its ability to classify emotions in real-world scenarios and how well it performs despite partial occlusion due to masks, testing was conducted using unseen data. We needed to test the model to ensure it would accurately represent emotions in situations like photos with partially masked faces.

Test Dataset and Environment

Evaluation was conducted using a separate masked\_test folder containing images classified into seven separate facial expressions. Expressions such as anger, happiness, fear, neutrality, sadness, disgust and surprise were included in the model’s output. These test images varied widely in the emotions displayed, lighting and face variations.

Structured evaluation was performed using the evaluate\_model.py script. It uses evaluating the trained CNN model on the test dataset and generates additional metrics through Keras’ evaluate() function and information from sklearn classifiers.

**Metrics Used**

A wide range of metrics was chosen to assess the model’s performance.

• Accuracy: The percentage of emotions predicted correctly out of all test samples.

• Precision & Recall: We evaluated how reliably the model is able to discern emotions that are often confused (such as sad and neutral).

• F1-Score: An average of the two metrics that considers class imbalance.

• Confusion Matrix: We analyzed where the model made errors across different emotions.

Results

Accuracy was high across the test set and the model performed best on commonly seen emotions such as happy and neutral. Models struggled somewhat with distinguishing emotions from underrepresented classes like disgust and fear. Misclassifications between sad and neutral were common due to the limited information available from facial expressions when only the upper half of the face is used.

A demo application was created with the model running in real time using streamlit\_demo.py. As a result, users could submit images or use their webcam to perform real-time prediction. The model’s performance showed it was ready for use in real-time applications or large-scale deployments.

Testing the CNN model showed it is capable of accurately discriminating a wide range of emotions in images with people wearing masks. Its performance is consistent under different circumstances and also holds up during real-time predictions.

**Summary of Additional Features**

The project offers a number of valuable extensions that increase the model’s ease of use, automation and flexibility for both software engineers and users of the app.

A major addition is a user-friendly interface built with Streamlit. The built-in web application (streamlit\_demo.py) lets users demo and trial the emotion recognition model instantly online. People can load their own photos or directly capture faces using the live webcam, after which the app detects and classifies emotions from the displayed images. Users instantly receive predictions and their associated scores, making the app ideal for presentations, tests or actual implementations.

A major improvement is the automatic masking pipeline available in main.py. The pipeline uses synthesized masks to cover faces, mimicking conditions found in the real world. OpenCV is employed to apply masks to faces in large numbers of photos, allowing the generation of a dataset ideal for training models capable of processing occluded images.

A feature of the system is model checkpointing which saves the best-performing model as it is trained. Automatic selection of the best model eliminates the need for researchers to choose and save the ideal results manually.

As a result, the project has become an integrated end-to-end deep learning solution for face recognition with occluded regions.

## **Discussion and Conclusion**

### **Discussion**

We explored the development of a deep learning system that recognizes facial emotions even when the bottom half of the face is covered by a mask. The presence of masks creates numerous difficulties because vital expressions involving the mouth become unreadable. Most standard face detection approaches and emotion recognition algorithms require clear views of the entire face which limits their performance when crucial parts of the face are covered.

A custom CNN was developed by training it on a dataset of masked facial expressions. An appropriate model design was ensured by employing Conv2D layers together with ReLU activations and dropout layers. Data augmentation was performed while training to improve both the variety and overall performance of the model.

The training curve sustained consistent improvement and the final model achieved excellent results on data featuring masked faces. Detailed analysis of model performance was made possible by employing the evaluate\_model.py script alongside produced confusion matrices and classification reports. The model showed impressive results overall, although it struggled somewhat to distinguish between neutral and sad expressions because of the reduced visibility of facial expressions.

Building a Streamlit web app to interact with the model directly from an image or webcam camera served as a crucial element of this project. Using Streamlit not only demonstrates how the model is implemented in practice but also makes it easy to interact with and explore by users.

An innovative strategy for dealing with increasingly commonly used facial coverings was introduced by implementing the automation of mask application to datasets via the main.py script.

**Lessons Learned**

Working on this project helped me improve my theoretical understanding of deep learning as well as my ability to apply deep learning techniques in practice. I gained hands-on experience in:

Creating and training sequential convolutional neural networks for the purpose of image classification

Preprocessing and augmenting large datasets

I integrated TensorFlow/Keras, OpenCV and Streamlit into the algorithm.

Organizing data workflows for storage, loading and evaluation.

Addressing real-life image problems such as partial blockage of faces.

Testing with high comparability and crafting easy-to-read scripts was crucial, especially when combining several tools and scripts.

Limitations

The project exhibits several areas for improvement.

Performance of the model can degrade in scenarios of poor illumination or image blurriness captured by the camera.

A lack of images representing feelings like disgust and fear may be a reason why the model struggled to classify those emotions.

The present model is able to identify emotions in individual faces but cannot simultaneously detect multiple faces or follow facial movements in busy scenarios.

The current interface is not yet designed with complete consideration for light mode accessibility and mobile responsiveness.

Expanding the dataset to include faces of varying ethnicity, age and lighting could make the system more inclusive and resilient.

**Conclusion**

Overall, the project shows that a deep learning system is capable of precisely recognizing emotions from masked faces in images. The pipeline integrates CNN architecture, automated preprocessing and an interactive interface allowing for efficient steps from dataset preparation to real-time inference.

The adaptable, reliable and easy-to-use nature of the system has the potential to be used as a foundation for many advanced applications in fields such as education, surveillance and human-computer interaction.

Advancements such as mobile compatibility, the ability to recognize multiple faces and a larger training dataset will enable this project to develop into a highly practical tool for intelligent systems that understand faces.

## **References:**

Abadi, M., et al. (2016). TensorFlow: A scalable framework for performing machine learning. Retrieved from https://www.tensorflow.org

OpenCV. (n.d.). OpenCV Documentation. Retrieved from https://docs.opencv.org/

Brownlee, J. (2019). Deep Learning for Computer Vision. Machine Learning Mastery.

Deep Learning by Goodfellow et al. Deep Learning. MIT Press.

PyImageSearch. (2023). Detecting Face Masks in Real Time Using Python, OpenCV and Deep Learning. Retrieved from <https://pyimagesearch.com>

TensorFlow. (n.d.). TensorFlow API Documentation. Retrieved from https://www.tensorflow.org/api\_docs

Kingma, Diederik P., & Ba, Jimmy. Adam: An algorithm for efficiently optimizing functions when data is sampled. arXiv preprint arXiv:1412.6980.