National University of Computer and Emerging Sciences, Lahore

SnapSense- Live Face Recognition

Sophia Razzaq 21L-5607
Idrees Arshad 21L-5632

Instructor: Dr. Asma Ahmad

FAST School of Computing
National University of Computer and Emerging Sciences
Lahore, Pakistan
February 2024

Contents

Introduction

The project aims to develop a face recognition/detection system using Python and a self-created dataset of our own faces. The motivation behind selecting this project lies in the wide range of applications where face detection plays a crucial role. Face detection is a fundamental task in computer vision with diverse applications in facial recognition systems, surveillance, biometrics, emotion analysis, and more. By building a face detection AI system from scratch, we gain a deep understanding of the underlying algorithms and techniques, enabling us to contribute to this field and enhance our knowledge in artificial intelligence and computer vision.

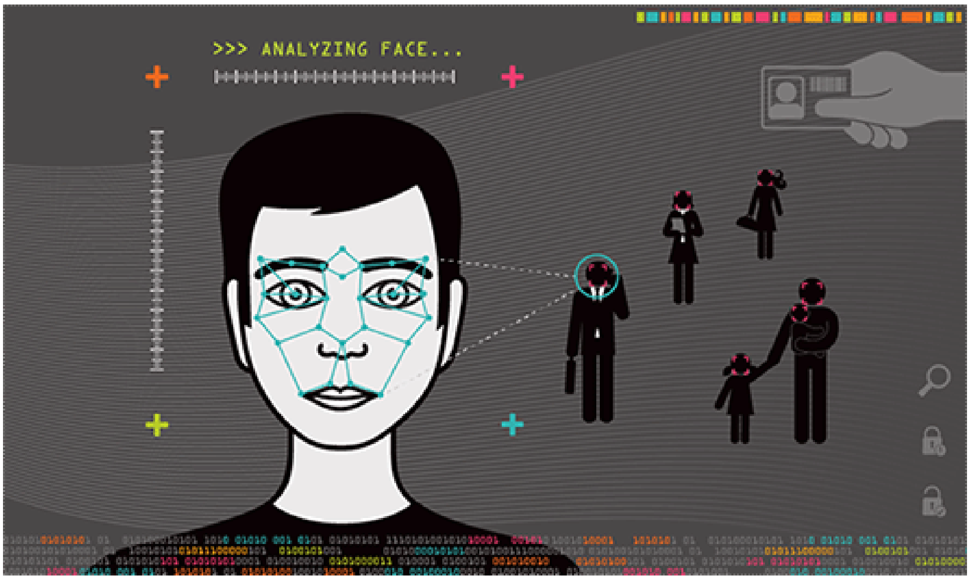

Figure 1: Face Detection.

Challenges

The project presents a set of challenges, including collecting a real-time dataset using our own faces, achieving high model accuracy, and building a face detection system from scratch. These challenges encompass capturing diverse face variations, optimizing the model's performance by addressing occlusion and varying lighting conditions, and implementing complex algorithms. Overcoming these hurdles not only enhances our understanding and expertise in artificial intelligence but also enables us to develop a customized and robust face detection solution tailored to our specific dataset.

Uniqueness of our project

While numerous state-of-the-art solutions exist in face detection, our project distinguishes itself by not relying on pre-existing models or architectures

Our Project differs from these already existing codes, is that these codes use pre-existing/ pre-trained models or libraries such as DeepFace or FER. Instead of using predefined convolutional neural network models like VGG-16, ResNet, ImageNet, etc., we will use the TensorFlow framework to build each layer of our model from scratch.

By creating our own face detection models from scratch, tailored to our unique dataset, we deepen our understanding of the underlying algorithms and techniques. This approach enables us to customize and fine-tune the models, potentially yielding improved accuracy and performance compared to generic solutions.

High level Architecture

The high-level architecture of our face detection project consists of the following components:

a) Data Collection: We capture a diverse dataset of our own faces, including variations in lighting conditions, poses, and expressions.

b) Preprocessing: The collected images undergo preprocessing steps such as resizing, normalization, and noise reduction to enhance the quality and consistency of the dataset.

c) Model Design: We design and implement a custom face detection model from scratch. This involves selecting appropriate neural network architectures, defining loss functions, and training the model on our dataset.

d) Training: The model is trained using the collected dataset, employing techniques like data augmentation, regularization, and optimization algorithms to improve its performance.

e) Evaluation: The trained model is evaluated on a separate test dataset to assess its accuracy, precision, recall, and other performance metrics.

f) Deployment: The final trained model is deployed in a real-time application, capable of detecting faces in images or videos.

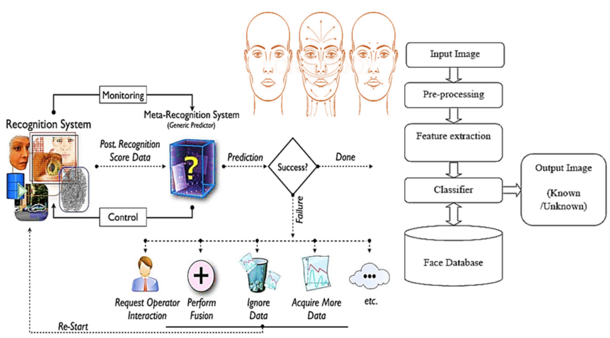

Figure 2: High level architecture.

Available Datasets

For this project, we have created our own dataset by capturing images of our own faces. This dataset contains variations in lighting conditions, poses, expressions, and backgrounds. While there are publicly available face detection datasets like the WIDER Face dataset, our project focuses on using a self-created dataset to showcase the effectiveness of our custom-built models.

Following are some datasets on kaggle: Click here.

References

During the research of our project, we have referred to various resources and research papers to gain insights into face detection algorithms and techniques. Some key references include:

Reference 1.

Reference 2.

Reference 3.

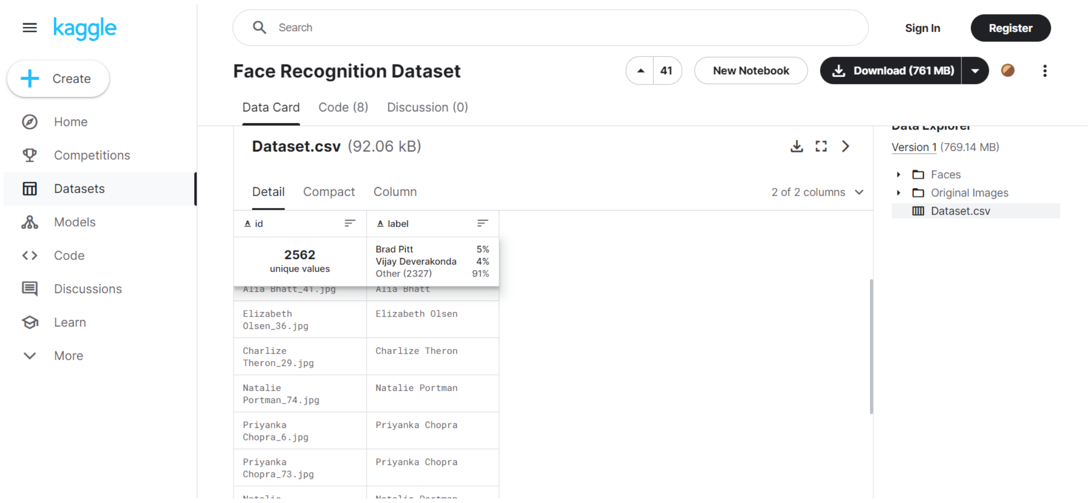

Figure 3: Kaggle dataset.

Conclusion

In summary, our project differentiates itself by building a custom face detection system using TensorFlow Keras. By leveraging these tools and tailoring the models to our specific dataset, we aim to enhance accuracy and performance. This approach allows us to deepen our understanding of face detection algorithms while creating a reliable solution for detecting faces in real-world scenarios.