A black background with white text

Description automatically generated

**Partial Facial Recognition**

Final Project Report

Julia Szostakiewicz

2210559

Supervisor: Dr Naseer Al-Jawad

University of Buckingham

School of Computing

Autumn 2024

**Declaration**

This declaration is to confirm that all of the work in this project and report is entirely mine and has been written by me and has not been submitted for any other degree of professional qualification.

**Acknowledgement**

I wanted to express my deepest appreciation and gratitude to Dr Naseer Al-Jawad for being a brilliant supervisor throughout the duration of this project. His guidance, support and patience have made a world of a difference, and I could not have completed this project to the extent I have done without his support, and definitely his patience. My greatest thanks go to him.

**Abstract**

This project explores the enhancement of facial recognition systems using a pre-trained VGG16 model fine-tuned on a modified Labelled Faces in the Wild (LFW) dataset. The primary objective was to identify faces that were only partially visible, due to various occlusions such as lighting, posing and coverings. The approach involved careful preprocessing, training of the model, and repeated fine-tuning. This project highlights the complexity of facial recognition under constrained environments and provides valuable insights for future improvements in partial facial recognition models.

Table of Contents

**Glossary of Terms** …………………………………………………………………………. 5

**Figure Table** ………..……………………………………………………………………… 6

**Chapter 1 – Introduction** …………………………………………………………………. 7

* 1. Report Structure ………………………………………………………………... 7
  2. Background ……………………………………………………………………. 8
  3. Aims and Objectives …………………………………………………………… 9

**Chapter 2 – Literature Review** …………………………………………………………. 11

2.1 Historical Development ……………………………………………………….. 11

2.1.1 Algorithms ……………………………………………………………11

2.1.2 Datasets ………………………………………………………………13

2.2 Feature Extraction ……………………………………………………………. 15

2.3 Artificial Neural Networks ……………………………………………………. 18

2.3.1 Convolutional Neural Networks ……………………………………. 19

2.3.2 Layers & Functions …………………………………………………. 20

2.4 Reviewed Algorithms …………………………………………………………. 22

2.4.1 FaceNET ……………………………………………………………. 22

2.4.2 VGG-16 ……………………………………………………………. 23

2.4.3 OpenFace ……………………………………………………………. 24

2.5 Vera Lettovska’s Project ………………………………………………………. 25

2.6 Real-World Applications ……………………………………………………… 25

2.6.1 Security Enhancement ………………………………………………. 26

2.6.2 Use Cases …………………………………………………………… 27

**Chapter 3 – Design** ……………………………………………………………………… 29

3.1 VGG-16 Algorithm …………………………………………………………. 29

3.2 Frameworks & Libraries ……………………………………………………. 30

3.2.1 Anaconda …………………………………………………………. 30

3.2.2 TensorFlow …………………………………………………………..30

3.2.3 Keras ……………………………………………………………….. 31

3.2.4 Dlib …………………………………………………………………. 31

3.2.5 OpenCV ……………………………………………………………. 31

3.2.6 NumPy ……………………………………………………………. 32

3.2.7 Scikit-learn …………………………………………………………. 32

3.3 Datasets ………………………………………………………………………. 32

3.3.1 Labelled Faces in the Wild (LFW) …………………………………. 32

3.3.2 Partial Labelled Faces in the Wild (PartialLFW) …………………. 33

**Chapter 4 – Implementation & Testing** …………………………………………………34

4.1 Data Preprocessing …………………………………………………………… 34

4.2 Transfer Learning …………………………………………………………….. 35

4.3 Model Compilation ……………………………………………………………36

4.4 Model Training ………………………………………………………………. 37

4.5 Problems Encountered ………………………………………………………. 37

4.5.1 Hardware Limitations & Environment Changes …………………… 37

4.5.2 Debugging Challenges ……………………………………………… 38

**Chapter 5 – Results** ……………………………………………………………………… 40

5.1 Training & Validation Performance …………………………………………. 40

5.2 Fine-Tuning and Enhanced Model Performance ……………………………. 41

**Chapter 6 – Evaluation & Conclusion** ………………………………………………. 43

6.1 Evaluation ……………………………………………………………………. 43

6.2 Conclusion ……………………………………………………………………. 44

**Glossary of Terms**

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| LFW | Labelled Faces in the Wild |
| PLFW | Partial Labelled Faces in the Wild |
| LBP | Local Binary Patterns |
| ANN | Artificial Neural Networks |
| CNN | Convolutional Neural Network |
| VGG-16 | Visual Geometry Group 16-layer model, a convolutional neural network architecture |
| ReLu | Rectified Linear Unit, an activation function used in neural networks |
| PCA | Principal Component Analysis |
| LDA | Principal Component Analysis |
| GPU | Graphics Processing Unit |
| RMSProp | Root Mean Square Propagation, an optimizer used in training neural networks |
| Adam | An optimization algorithm used for training deep learning models |
| Keras | A deep learning API written in Python, running on top of TensorFlow |
| TensorFlow | An open-source software library for dataflow and differentiable programming |
| OpenCV | Open Source Computer Vision Library |
| Dlib | A toolkit for making real-world machine learning and data analysis software in C++ and Python |
| ImageNet | A large visual database designed for use in visual object recognition research |

**Figure Table**

|  |  |  |
| --- | --- | --- |
| Figure | Label | Page |
| 2.1 | (a) Eigenfaces vs. (b) Fisherfaces | 12 |
| 2.2 | Partial Labelled Faces in the Wild | 14 |
| 2.3 | Feature Extraction | 18 |
| 2.4 | Artificial Neural Network Diagram | 19 |
| 2.5 | Convolutional Neural Network Diagram | 20 |
| 5.1 | Graph representing the results of model training | 41 |
| 5.2 | Fine-tuning accuracy percentage | 42 |

**Chapter 1 – Introduction**

Facial recognition technology is growing as an essential component in a wide range of real-world applications, such as security services, user authentication, and social media. Nevertheless, the majority of existing models have difficulties when confronted with faces that are just partially visible, obscured, or taken in a variety of environments. The objective of this study is to address this challenge by developing a VGG-16 model, using deep learning architectures that is meant to improve its ability to identify partially visible faces.

* 1. **Report Structure**

This report it structured to provide a comprehensive analysis of partial facial recognition technology and its implementation. To achieve this in an organised way, the report has been split into six main chapters:

Chapter 1 - Introduction: The report is introduced in this chapter, covering its background, aims, and objectives. It establishes the subject and offers an outline of the report's organisation.

Chapter 2 - Literature Review: This chapter focuses on the literature review, providing an extensive analysis of how facial recognition algorithms and datasets have evolved over time, whilst also examining the role of Artificial Neural Networks in this context. There is also a focus in this chapter on analysing specific algorithms, including FaceNET, VGG-16, and OpenFace. Furthermore, it explores the prior research undertaken by Vera Lettovska and discusses real-world applications of partial facial recognition technology.

Chapter 3 – Design: This chapter focuses on the design decisions made throughout the project, including chosen elements such as the VGG-16 algorithm, as well as various frameworks, libraries, and datasets. The chapter also outlines how these components were integrated in order to successfully accomplish the objectives of the project.

Chapter 4 – Implementation & Testing: This chapter outlines the multiple stages of implementing the facial recognition model, such as data preprocessing, transfer learning, model compilation, model training. Additionally, this section also focuses on challenges encountered during testing as well.

Chapter 5 – Results: This penultimate chapter includes the results obtained from the testing of the model, with an emphasis on the accuracy percentage and the comparison of the implemented model's performance with other well-established methods.

Chapter 6 – Evaluation & Conclusion: The final chapter assesses the effectiveness of the model, considers the fulfilment of the project goals and objectives, and presents a comprehensive summary of the main results. Furthermore, it suggests various possibilities for future research and improvements.

* 1. **Background**

In recent years, facial recognition technology has gained significant prominence in several industries such as security, banking, and healthcare. The widespread adoption of the technology may be attributed to its capacity to precisely recognise persons by facial characteristics, particularly in sectors that demand strict safety precautions and access control. Conventional facial recognition algorithms depend on whole facial pictures to achieve precise identification. However, real-life situations sometimes show partial or obscured views of people, such as faces covered by masks, sunglasses, or images taken from various angles. In response to this limitation, partial face recognition methods have been established, capable of identifying people using only partial facial data.

The use of partial face recognition technology is especially beneficial in surveillance, forensic investigations, and other security-related applications when the full view of the face is not guaranteed.  By utilising complex algorithms and deep learning methods, partial recognition systems can adjust to different conditions and uphold a high degree of precision even when only a small portion of the face is visible.  This adaptability overcomes the limitations of traditional methods and improves the robustness of face recognition systems in real-world applications.

The objective of this research is to investigate and implement advanced algorithms and deep learning frameworks for partial face recognition. This work intends to enhance existing models for partial face identification in order to contribute to the ongoing research in this field and provide practical answers to the problems presented by real-world limitations.

* 1. **Aims and Objectives**

The primary aim of this project is to develop and assess a face recognition model capable of precisely identifying people utilising just a selected portion of their facial data. The main incentive behind this objective is the increasing demand for strong recognition systems that can operate successfully in varied contexts where total facial view cannot be guaranteed.   
  
This project aims to solve the issues associated with partial facial recognition by employing advanced deep learning algorithms. By building a model capable of reliable identification under various conditions, the project hopes to contribute to the larger area of facial recognition technology and its practical applications in security, surveillance, and user authentication.  
  
In order to accomplish this general objective, the project will concentrate on the following primary goals:  
  
- Enhance Understanding and Knowledge: Conduct comprehensive research on the current state of face recognition, with a particular focus on partial recognition approaches. An extensive evaluation of current methods, datasets, and frameworks will be conducted to establish a strong conceptual foundation for the project.

- Develop an Adaptable Model: Design and develop an adaptable face recognition system capable of adjusting to different real-world situations in which faces are partially occluded. The suggested method would employ convolutional neural networks (CNNs) and transfer learning methods to enhance performance in various environments.  
  
- Promote innovation in Machine Learning Applications: This project will not only produce a new model but also encourage innovation through researching how modern machine learning techniques might be applied for challenging recognition tasks. The results are intended to give insights and direction for future study in this expanding sector.

The success of this research will be judged by the model's capacity to execute correct identification in situations of partial face visibility, demonstrating its potential to strengthen security measures in many practical applications.

**Chapter 2 – Literature Review**

This chapter will focus on the literature review.

**2.1 Historical Development**

Facial recognition technology has advanced massively since it was first introduced, progressing from simple geometric models to powerful deep learning approaches that can evaluate complex facial characteristics. The development of this technology can be traced back to the 1960s when early attempts focused on manual feature extraction and basic statistical analysis. Over the course of several decades, these techniques have been improved by integrating more advanced algorithms and datasets, thus establishing the foundation for present-day high-accuracy systems.

**2.1.1 Algorithms**

The earliest face recognition systems were mainly geometric in nature, depending on manually defined points and simple measurements to compare facial characteristics. For instance, the "man-machine" project in the 1960s needed a human operator to manually select precise facial coordinates, such as the distance between the eyes or the width of the mouth, which were then used to match faces against a database (Bledsoe & Chan, 1965). Although this technique marked an important milestone, it was restricted by the level of manual labour necessary and the subjective aspect of identifying features.  
  
During the 1970s and 1980s, algorithms were developed to automate the process of extracting face traits, therefore decreasing the need for human involvement. Eigenfaces, which originated in the late 1980s, significantly advanced the field by employing principal component analysis (PCA) to decrease the number of dimensions in facial data and enhance the accuracy of identification (Turk & Pentland, 1991). Eigenfaces works by converting facial photographs into a collection of key characteristics, which can then be compared to recognised faces registered in a database. This method marks the beginning of holistic approaches to facial recognition, which analyse the entire face as a set of connected characteristics rather than evaluating individual points in isolation.  
  
The development of more complex models, such as Fisherfaces and local binary patterns (LBP), significantly improved identification accuracy by accounting for differences in illumination, facial expressions, and orientations (Belhumeur et al., 1997). Fisherfaces paired PCA with linear discriminant analysis (LDA) to create more discriminative feature spaces, whereas LBP focused on local texture patterns that are more resilient to changes in appearance.

A collage of images of a person

Description automatically generated

Figure 2.1 - (a) Eigenfaces vs. (b) Fisherfaces

The introduction of deep learning in the 2010s changed the area by allowing models to learn sophisticated, non-linear representations of face data straight from massive datasets. Convolutional Neural Networks (CNNs) have emerged as the fundamental component in modern facial recognition systems. CNNs employ a hierarchical framework of convolutional layers to automatically extract features at various levels of abstraction, from simple edges to complex textures and forms (Krizhevsky et al., 2012). This transition from handcrafted features to learnt features constituted a paradigm change, leading to major increases in accuracy and robustness.

**2.1.2 Datasets**

The advancement of facial recognition algorithms has been directly related to the availability of more diverse and complex data. Early datasets, such as the AT&T and Yale Face Databases, featured just a small number of controlled photos, frequently captured under identical lighting and posing conditions. These datasets were suitable for testing early algorithms but lacked the variety needed for real-world applications.  
  
The FERET database, built in the 1990s, was one of the first large-scale efforts to provide the starting point for face recognition research (Phillips et al., 2000). With nearly 14,000 photos of 1,199 individuals recorded under varied situations, FERET provided a more rigorous testing environment that exposed the limits of previous algorithms in managing differences in position, expression, and lighting.  
  
The Labelled Faces in the Wild (LFW) dataset was created in 2007 to address the demand for more representative data. It comprises more than 13,000 face photos gathered from the internet (Huang et al., 2007). In contrast to previous datasets, LFW photos were taken in uncontrolled environments, showing significant variation in terms of age, gender, ethnicity, lighting, and background. This dataset established a new standard for evaluating modern facial recognition algorithms, enabling the creation of more advanced models capable of tackling increasingly complex tasks.   
  
To address specific issues with identifying partially visible faces, the Partial Labelled Faces in the Wild (PartialLFW) dataset was developed, expanding upon the original LFW by concentrating on conditions where faces are partially occluded. Accessible on GitHub (Hoermann, 2021), this collection comprises photos including multiple occlusions, including hats, glasses, and masks, which replicate scenarios often seen in surveillance, security, and other practical contexts.

A crucial resource for strengthening the adaptability and resilience of face recognition systems is the PartialLFW dataset, which provides researchers with a significant resource for training and testing models that must operate under circumstances of partial face visibility.

A collage of a person's face

Description automatically generated

Figure 2.2 - Partial Labelled Faces in the Wild

The use of PartialLFW signifies a notable advancement in the development of face recognition datasets, as it explicitly targets the increasing demand for models capable of efficiently processing obstructed data. The provision of a specialised dataset for partial recognition by PartialLFW facilitates the creation of algorithms that can sustain a high level of accuracy even in cases when only certain sections of the face are visible. This makes PartialLFW an essential tool for the advancement of the field of partial recognition.   
  
The continuous development and expansion of datasets continually accelerates advancements in facial recognition, providing various training data necessary to enhance model generalisation across many settings and target groups. Several publicly accessible datasets, like CelebA, MS-Celeb-1M, and VGGFace, include millions of photos, facilitating the creation of deep learning models capable of learning from significant face data.

**2.2 Feature extraction**

Feature extraction is a vital process in facial recognition that involves the detection and isolation of unique characteristics of a face, therefore enabling the system to achieve precise matching and identification. This process turns raw picture data into a collection of useful features that capture critical information about the facial structure, texture, and spatial connections among facial components such as the eyes, nose, and mouth. The efficiency of feature extraction directly improves the performance of recognition models, improving their capacity to identify between persons under varying environments.   
  
Feature extraction generally follows multiple phases meant to decrease data complexity while keeping the most important characteristics of the face:  
  
Preprocessing: Before feature extraction, photos are pre-processed to standardize properties like size, illumination, and orientation. Techniques such as histogram equalization, scaling, and alignment help to increase reliability and precision in the feature extraction process.

Detection of Key Facial Landmarks: Key points on the face, such as the eyes, nose, mouth, and jawline, are recognised using landmark detection algorithms like Dlib’s facial landmark detector. These landmarks offer the fundamental points from which characteristics are extracted.

Extraction of Local and Global Features:  
Local Features: Local feature extraction focuses on obtaining small information from specific face areas, such as the texture patterns around the eyes or lips. Methods like Local Binary Patterns (LBP) evaluate pixel intensity changes within tiny regions, providing descriptors that capture these localized properties.

Global Features: Global feature extraction takes into account the complete visual characteristics of the face, including elements such as shape and structure. Techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) minimise the data’s dimensionality and emphasise the most unique traits that separate faces.

The extraction of hierarchical features from data is a common use of Convolutional Neural Networks (CNNs) in contemporary deep learning methods. CNNs extract features across multiple layers, starting from basic edges and continuing to complex textures and forms. The hierarchical extraction procedure allows deep learning models to effectively capture the key high-level abstractions necessary for reliable face recognition.

Feature Vector Creation: The retrieved characteristics are combined into a feature vector, a numerical representation that uniquely describes the face. These vectors are then utilised for comparison against stored vectors from known persons, simplifying recognition or verification.  
  
Feature extraction is crucial to the performance of facial recognition systems for various reasons:  
  
Reduction of Complexity: Raw photos include significant amounts of data, most of which is unneeded for recognition tasks. Feature extraction distills all of this data into a more manageable form, decreasing computational requirements and enabling faster processing.

Improvement in identification Accuracy: By prioritising the most distinctive aspects of a face, the process of extracting features boosts the capacity of the identification model to precisely recognise persons, even in difficult situations such as changing lighting, postures, or partial occlusions.

Faces are susceptible to fluctuations in posture, expression, age, and environmental conditions, which affect their adaptability to variation. Effective feature extraction approaches capture invariant aspects of faces, making recognition systems more tolerant to these changes.

Enhancement of Model Generalisation: Accurately extracted features enhance the ability of models to generalise more effectively to new data, hence assuring that trained models exhibit strong performance on previously viewed pictures. This is particularly crucial for deep learning models that rely on substantial and diverse datasets.

Facilitation of Partial Recognition: In instances when just a piece of the face is visible, feature extraction can nevertheless find relevant traits from the available data. This functionality is essential in applications such as security and surveillance, where complete face visibility is not always possible.

A close-up of several faces

Description automatically generated

Figure 2.3 - Feature Extraction

**2.3 Artificial Neural Networks**

Facial recognition systems today rely heavily on Artificial Neural Networks (ANNs) as they offer computational support for advanced pattern detection. ANNs are designed to mimic the architecture and functioning of the human brain, which involves interconnected layers of neurons that analyse incoming data and make predictions based on it. With regards to facial recognition, ANNs support computers in acquiring knowledge about intricate patterns within facial data such as textures, shapes, spatial connections etc., ultimately enabling them to differentiate between various individuals with relative ease.

A diagram of a machine

Description automatically generated

Figure 2.4 - Artificial Neural Network Diagram

**2.3.1 Convolutional Neural Networks**

In the field of imagine data analysis, Convolutional Neural Networks (CNNs) are a special kind of Artificial Neural Networks (ANNs) that have been specifically developed to perform well in tasks such as facial recognition. CNNs function by applying convolutional filters to the input images, which aids in recognising spatial hierarchies of features – from low-level edges to high-level object parts.   
  
A typical CNN design has many layers: convolutional layers that identify features, pooling layers that down sample the input to minimise computing stress and fully connected layers that generate predictions based on the learnt features. One of the primary advantages of CNNs is their capacity to automatically learn which characteristics are most relevant for recognition, reducing the need for human feature extraction and enabling for end-to-end learning.  
  
For facial recognition, CNNs can be trained on enormous datasets, affecting millions of parameters to reach high accuracy. The convolutional layers are especially effective for facial recognition because they can learn to recognise small variations between faces, even under shifting lighting conditions, positions, or partial occlusions. Advanced approaches such as transfer learning and fine-tuning further increase the versatility of CNNs, enabling them to be reused for particular tasks with relatively little amounts of task-specific input.

A cat's head and a diagram

Description automatically generated with medium confidence

Figure 2.5 – Convolutional Neural Network Diagram

**2.3.2 Layers & Functions**

CNNs consist of many layers, each having a distinct purpose in the processing of picture data. I’ve explained the function of each layer below:   
  
Convolutional Layers: These are the layers that are the key feature extractors in CNNs. They use filters that slide across the input image to identify edges, textures, and various other fundamental data. As data traverses deeper levels, the retrieved features get more complicated, reflecting higher-level characteristics of the face.

Pooling Layers: Pooling layers are used to minimise the spatial dimensions of the data, maintaining the most relevant aspects while removing unnecessary data. Max pooling, the most widely used kind, chooses the maximum value within a set window, essentially summarizing the presence of a feature.

Activation Functions: Activation functions add non-linearity into the network, enabling CNNs to simulate complicated interactions. The Rectified Linear Unit (ReLU) is extensively used because it is computationally efficient and helps reduce any concerns associated to vanishing gradients during training.

Normalization Layers: These layers alter the outputs of the convolutional layers and increase training stability and speed. Batch normalization, for example, normalizes the output of a layer throughout the mini-batch, allowing for faster convergence and enhanced generalization.

Fully Connected Layers: Positioned at the end of the network, fully connected layers incorporate all the characteristics gathered by the preceding layers and make final judgements. In facial recognition, these layers provide probability scores that correlate to each individual identity stored in the database.

The SoftMax function is commonly used in the output layer of classification models to transform the final set of values into probabilities, therefore highlighting the most probable match.

The better performance of CNNs over conventional face recognition approaches can be attributed mainly to their capacity to autonomously extract significant information from pictures without the need for human interaction. The combination of this ability and the scalability of neural networks renders CNNs the most favoured approach for facial recognition applications.

**2.4 Reviewed Algorithms**

Facial recognition technology is dependent on complex algorithms that efficiently handle and evaluate facial data. Some of the most prominent among the approaches are deep learning-based models that use neural networks to accurately extract and compare face characteristics. This section provides a general overview of the major algorithms that have influenced the advancement of face recognition:

**2.4.1 FaceNET**

FaceNET, created by Google researchers in 2015, is a deep convolutional neural network (CNN) that connects facial pictures with a compact Euclidean space. In this space, the distance between two points represents the similarity of the faces (Schroff et al., 2015). This technique transformed the field of face recognition by prioritising the integration of faces into a spatial framework where similar faces are positioned in closer proximity, while dissimilar faces are positioned farther apart. FaceNET utilises a Convolutional Neural Network (CNN) structure that is trained using a triplet loss function. This loss function aims to minimise the distance between pairs of matching faces and maximise the distance between pairs of non-matching faces.  
  
The main innovation of FaceNET is its capacity to generate accurate and selective 128-dimensional feature vectors, referred to as embeddings, for every face picture. FaceNET's embeddings exhibit robustness to changes in posture, illumination, and occlusion, rendering it very successful in real-world situations. Without the need for further fine-tuning, the embeddings may be directly utilised for tasks such as face verification, identification, and clustering.  
  
FaceNET attained a record-breaking 99.63% accuracy on the Labelled Faces in the Wild (LFW) test, showcasing its remarkable superiority compared to earlier approaches. The success of this model may be attributed to its unique integration of deep learning and metric learning, which enables it to effectively generalise over a wide range of datasets and applications. Nevertheless, the processing requirements and need on extensive training data of FaceNET might present difficulties, especially in settings with limited information.

**2.4.2 VGG-16**

Developed by the Visual Geometry Group at the University of Oxford, VGG-16 is a deep convolutional neural network (CNN) architecture renowned for its simplicity and effectiveness in image identification tasks, particularly in the domain of facial recognition (Simonyan & Zisserman, 2014). The VGG-16 model comprises 16 layers that include 13 convolutional layers and three fully linked layers. It incorporates small 3x3 filters and maintains a consistent architecture throughout.  
  
An inherent advantage of VGG-16 is its capacity to acquire hierarchical representations of face characteristics by means of deep layers, encompassing both fundamental details like edges and textures, as well as general trends such as the arrangement of facial elements. The hierarchical feature extraction approach of VGG-16 renders it quite versatile for many face recognition situations, including those that include partial visibility or occlusion.  
  
The modular architecture of VGG-16 enables transfer learning by allowing the fine-tuning of pre-trained weights from large datasets such as ImageNet for particular face recognition applications. The ability for flexibility is especially advantageous for projects that have a scarcity of data, as it allows the model to utilise existing information and get a high level of accuracy with no further training.  
  
Although highly efficient, VGG-16 is computationally demanding, necessitating substantial system resources and memory. The depth and thick layer configurations of this architecture might result in extended training durations in comparison to more optimised designs. Nevertheless, VGG-16 continues to be a popular option for facial recognition because of its robust performance and adaptability in many implementations. In this project, I have chosen to utilise the VGG-16 algorithm in my model.

**2.4.3 OpenFace**

OpenFace is a face recognition system developed as an open-source project by scholars working with Carnegie Mellon University. This approach expands upon deep learning methods similar to those used by FaceNET, but with a focus on accessibility and seamless integration into other systems (Amos et al., 2016). OpenFace is specifically developed as a lightweight and ready to implement substitute for face recognition models. This ability makes it well-suited for academic research, commercial applications, and seamless interaction with customised systems.  
  
The OpenFace model use a deep Convolutional Neural Network (CNN) structure, modelled by FaceNET, which incorporates a triplet loss function to acquire embeddings that represent face similarity. OpenFace employs dlib to identify and align face landmarks, hence improving the accuracy of feature extraction through the standardisation of input photos ahead of processing.  
  
An unique characteristic of OpenFace is its ability to deliver real-time performance, even on typical consumer-grade hardware. This feature renders it a compelling choice for applications that prioritise performance and accessibility. The OpenFace framework has found applications in several domains such as security, human-computer interaction , and behavioural research.  
  
Although OpenFace provides extensive flexibility and simplicity, its accuracy is typically inferior to that of more intricate and processing-intensive models such as FaceNET and VGG-16. Nevertheless, in several real-world scenarios, the compromise between speed, resource utilisation, and precision is deemed acceptable, therefore establishing OpenFace as a useful asset within the expanding realm of face recognition technology.

**2.5 Vera Lettovska’s Project**

I was given access to Vera’s project, including her code and results, which provided a solid foundation for my research. By building upon her work, I was able to understand the strengths and limitations of the way she approached the project. This access allowed me to refine my methodology, expand on her findings, and provided me with a foundation to begin my research on.

**2.6 Real World Applications**

Beyond its early experimental stages, facial recognition technology has evolved into a crucial tool in a wide range of practical applications. The broad use of this technology in sectors such as security, banking, healthcare, and retail may be attributed to its capacity to offer precise, quick, and non-intrusive identification. The technology's adaptability in verifying identities, improving security measures, and simplifying user interactions has further established its significance in modern society.

**2.6.1 Security Enhancement**

A significant use of face recognition technology is in the field of security. In several domains such as law enforcement and personal device security, face recognition technologies offer a convenient and safe means of verification. During surveillance operations, face recognition technology is employed to oversee public areas, detect individuals of significance, and improve public safety by rapidly comparing collected photographs with criminal databases. This use is especially beneficial in locations with high security, such as airports, government buildings, and border control, where immediate identification is of the highest priority.   
  
Facial recognition technology is employed by financial organisations to enhance their security protocols, facilitating multi-factor authentication for banking transactions and granting access to confidential data. The integration of face recognition technology into mobile banking applications provides financial institutions with an additional biometric safeguard, therefore substantially mitigating the potential for unauthorised access and fraudulent activities.  
  
Furthermore, face recognition fulfils a crucial function in the process of verifying the identification of personal devices. Facial recognition technology has been incorporated by prominent technology firms into smartphones, laptops, and other devices, providing users with a smooth and reliable means of verification. This functionality not only improves user convenience but also reinforces device security by increasing the level of difficulty for unauthorised users to successfully access the device.

**2.6.2 Use Cases**

In addition to security, face recognition technology has been widely used in other sectors, all capitalising on its unique capability to precisely recognise individuals in different circumstances:  
  
Within the retail industry, facial recognition technology is employed to customise client experiences by customising in-store advertising according to demographic information or previous purchasing habits. Furthermore, this technology contributes to the improvement of security via the identification of recognised shoplifters and the reduction of retail theft.

In the healthcare industry, facial recognition technology facilitates efficient patient check-ins, ensures safe access to medical information, and enables comprehensive monitoring of patient status. Furthermore, it may be used to authenticate the identification of healthcare practitioners in restricted zones, therefore guaranteeing that only authorised individuals are granted entry to sensitive environments.

Attendance tracking and campus security are two applications in which face recognition technology is employed by educational institutions. The implementation of this technology streamlines the procedure of monitoring student attendance, therefore improving the effectiveness of administrative duties and mitigating instances of absenteeism fraud.

The use of face recognition technology by airports and hotels serves to accelerate check-in processes and improve the quality of customer service. Facial recognition technology is employed in airports to identify airline passengers, therefore aiding speedier boarding procedures and minimising the necessity for manual inspections. An example possible use involves the recognition of women who wear religious coverings like the niqab. Traditional identification techniques sometimes need the removal of face coverings, a process that can be culturally sensitive and uncomfortable. Advanced face recognition technology capable of precisely identifying persons based on visible facial characteristics like the eyes and eyebrows can resolve this problem. This measure would enable women who wear the niqab to keep their coverings while being reliably recognised, therefore improving their travel experience and showing respect for cultural customs.

The application of facial recognition technology is becoming more prevalent in stadiums and music venues as a means of providing ticketless entrance. This technology guarantees accurate identification of attendees and thereby minimises the risk of ticket fraud. Furthermore, this technology has the capability of increasing fan involvement by offering tailored experiences based on attendee data.

A wide array of applications highlights the significant impact of face recognition technology on everyday life, demonstrating its capacity to enhance security, maximise operational effectiveness, and deliver tailored services. With the ongoing evolution of technology, its applications are anticipated to broaden, generating new opportunities and addressing emerging challenges across diverse sectors.

**Chapter 3 – Design**

In this chapter, an overview of the design of the facial recognition model is provided.

For this project, I choose to utilise Python 3.8 to construct my model, integrating several separate frameworks and libraries to enable the development of a complete and functional version of my model.

**3.1 VGG-16 Algorithm**

The VGG-16 algorithm was selected as the starting point for this project because of its established effectiveness in image recognition problems, particularly in the domain of face recognition. A deep convolutional network design, with 13 convolutional layers and three fully linked layers, is the signature characteristic of VGG-16.   
  
An important advantage of VGG-16 is in its simplicity and the regular use of compact 3x3 convolutional filters, enabling the network to develop hierarchical representations of visual characteristics. This modular architecture allows the network to effectively capture both low-level  details, such as edges and textures, as well as more abstract characteristics, such as face structures and complex patterns. The capacity of VGG-16 to extract complex characteristics at several levels renders it very appropriate for partial face recognition, especially in cases when only a fraction of the face is visible.  
  
In order to use the pre-learnt weights of VGG-16, which were originally trained on the extensive ImageNet dataset, transfer learning was used. The use of this method allows the model to leverage existing knowledge of visual characteristics, therefore significantly reducing the training duration needed for the specific goal of partial face recognition. The upper layers of the network were trained using new data tailored to the project's needs, utilising the LFW dataset, helping the model to modify its learnt features to precisely identify faces even in situations when they are partially occluded.  
  
Despite its processing requirements, VGG-16 was chosen for its balance between accuracy, flexibility, and resilience, making it a perfect option for applications that need high precision, such as security and identification in settings with changing face visibility.

**3.2 Frameworks & Libraries**

The following frameworks and libraries were chosen based on their compatibility, extensive features, and widespread application in the field of machine learning:

**3.2.1 Anaconda**

Anaconda was chosen as the main environment manager for this project because of its excellent package management capabilities and capacity for creating isolated virtual environments, in addition to the fact it was recommended by my supervisor. This feature enables the efficient installation and administration of essential libraries, therefore guaranteeing consistency and replicability in the development process. Anaconda's easy integration with widely used deep learning frameworks such as TensorFlow and Keras rendered it the best selection for establishing the project environment.

**3.2.2 TensorFlow**

Developed by Google, the open-source deep learning framework TensorFlow offers comprehensive support for a wide range of machine learning applications including image classification, natural language processing, and facial recognition. The choice of TensorFlow was made with in mind of its capacity to scale, extensive community support, and efficient handling of large-scale data. The architecture provides a flexible structure for building and training neural networks, allowing the integration of tailored layers and loss functions to improve the model's effectiveness in partial face recognition.

**3.2.3 Keras**

Keras, an advanced neural networks application programming interface (API) implemented on TensorFlow, was utilised to construct and train the VGG-16 model. The straightforward syntax of Keras streamlines the process of model construction, facilitating experimentation with various designs and enhancing the performance of the model. The fast prototyping capability of the software is facilitated by its user-friendly interface, while its connection with TensorFlow offers the essential backend support for extensive training.

**3.2.4 Dlib**

Dlib is an open-source collection of machine learning algorithms and tools, known for its exceptional precision in detecting face landmarks. The present study employed Dlib to identify and align face characteristics, therefore assuring the uniform formatting of pictures prior to their transmission to the recognition model. This preprocessing stage improves the model's capacity to precisely extract pertinent characteristics, particularly in situations when visibility is limited.

**3.2.5 OpenCV**

The Open Source Computer Vision Library (OpenCV) was used to perform picture preprocessing operations including scaling, normalisation, and data augmentation. OpenCV's comprehensive range of functions for image modification facilitated effective management of input data, guaranteeing that the model obtained standardised pictures that were precisely tailored for extracting features.

**3.2.6 NumPy**

NumPy is a foundational Python library designed for scientific computing, with powerful capabilities for handling extensive, multi-dimensional arrays and matrices. NumPy was widely employed in this research for data manipulation and numerical operations, enabling the systematic pretreatment and augmentation of picture data.

**3.2.7 Scikit-learn**

A commonly used Python package for machine learning, Scikit-learn, was employed for data partitioning, evaluation metrics, and some preprocessing operations. Complementing the deep learning frameworks, its straightforward and effective tools for predictive data analysis offered supplementary features such model evaluation and performance benchmarking.

**3.3 Datasets**

**3.3.1 Labelled Faces in the Wild (LFW)**

An established standard in the field of facial recognition, the Labelled Faces in the Wild (LFW) collection comprises more than 13,000 photos of faces gathered from the internet. This dataset comprises faces that were recorded in unrestricted settings, exhibiting differences in lighting, posture, and expression. Consequently, it serves as a valuable asset for training algorithms to accurately identify faces in real-life situations.  
  
The LFW dataset was used as a fundamental dataset for preliminary model training, enabling the VGG-16 architecture to acquire a diverse range of face characteristics. The varied collection of photos facilitated the development of a strong feature extraction procedure for the model, which is crucial for applications involving partial recognition.

**3.3.2 Partial Labelled Faces in the Wild (PartialLFW)**

This study employed the Partial Labelled Faces in the Wild (PartialLFW) dataset to specifically tackle the problem of partial face recognition. This dataset expands upon the LFW by include photos that portray several forms of occlusions, such as masks, sunglasses, and other barriers. These images replicate real-life situations where complete face visibility is not always feasible.  
  
The PartialLFW dataset supplied the essential training samples to optimise the VGG-16 model, guaranteeing its ability to precisely recognise individuals even when only certain parts of their face were visible. The model was trained to extract and optimise the most informative elements of incomplete face data, such as eyes and eyebrows, hence enhancing its effectiveness in tasks such as detecting persons wearing religious covers or surgical masks.

**Chapter 4 – Implementation & Testing**

This chapter provides an description of the implementation and testing stages of the project, with a specific emphasis on the training and evaluation of the VGG-16 model for partial face recognition.

**4.1 Data preprocessing**

Data preprocessing is a crucial stage in the preparation of images for use into the model. To ensure consistent data quality, different preprocessing protocols were used to improve the model's capacity to correctly recognise faces from incomplete data.  
  
In order to comply with the input criteria of the VGG-16 model, the face pictures from the Labelled Faces in the Wild (LFW) and Partial Labelled Faces in the Wild (PartialLFW) datasets were downsized to a standard input size of 224x224 pixels. A normalisation technique was used to standardise the pixel values to a range of 0 to 1. This ensures that the model receives input data in a uniform way, therefore stabilising the training process.

Dlib's facial landmark detector was used to detect and align prominent face characteristics, including the eyes, nose, and mouth. The alignment process is crucial for the PartialLFW dataset, as different levels of occlusion might disturb the expected placement of face characteristics. By adjusting the alignment of the faces, the model achieved more precision in identifying the main characteristics, therefore improving its capability to recognise faces even in cases when only some sections were visible.

In order to enhance the model's generalisation capabilities and increase the variety of the training data, data augmentation techniques were implemented. These included rotations, displacements, magnifications, and horizontal inversions. Augmentation not only improves the model's capacity to identify faces in diverse orientations but also replicates real-life situations where faces may be partially hidden or visible from various perspectives.

The pre-processed data was divided into sets for training, validation, and testing. This employed a training set to instruct the model, a validation set to refine the model and reduce overfitting, and a test set to assess the ultimate performance of the model. This method of splitting guarantees that the correctness of the model is evaluated on data that has not been previously viewed, therefore offering a true evaluation of its capabilities.

**4.2 Transfer Learning**

Transfer learning played a crucial role in this effort by enabling the VGG-16 model to utilise the information acquired from training on the ImageNet dataset and apply it to the objective of partial face recognition.   
  
The pre-trained VGG-16 model, constructed using the ImageNet dataset, was imported together with its acquired weights. The weights correspond to a wide range of visual information, involving the capacity to identify key features such as edges, textures, and patterns that are essential for facial identification.

In addition, a modified version of the VGG-16 model was developed for partial face recognition by substituting the last completely linked levels with additional layers designed for this particular task. In order to maintain the useful low-level characteristics that had been previously learnt, the weights of the convolutional layers were not modified during training. The supplementary layers were further trained using the LFW and PartialLFW datasets, enabling the model to refine its higher-level characteristics for improved performance in the identification of partially visible faces.

The model underwent training using a variety of both whole and partial face photos, with the aim of enhancing its capacity to accurately recognise individuals using incomplete data. The use of transfer learning greatly decreased the duration of training and enhanced the performance of the model by using the existing fundamental information included in the VGG-16 architecture.

**4.3 Model Compilation**

Compiling the model required establishing the optimiser, loss function, and assessment criteria that would direct the training procedure.  
  
The choice of the Adam optimiser for this project was based on its high efficiency and versatility in effectively managing sparse gradients and noisy data. Adam uses the strengths of two well-known optimisers, AdaGrad and RMSProp, to enhance its performance in handling large datasets and complex models such as VGG-16.

The loss function employed was categorical cross-entropy, which is suitable for multi-class classification operations such as facial recognition. By measuring the difference between the expected probability distribution and the actual distribution, this function directs the model to modify its weights in order to minimise this difference over time.

The main evaluation metric I used was accuracy, which quantifies the ratio of accurate predictions generated by the model. In addition, careful monitoring of precision, recall, and F1-score was conducted to verify that the model not only achieved high overall accuracy but also effectively handled the intricacies of partial face recognition.

**4.4 Model Training**   
The model underwent training with a batch size of 32, which optimises the computational efficiency while ensuring the stability of gradient updates. Iterative training was carried out across 190 epochs to provide enough time for the model to learn and prevent overfitting.

I used early halting to mitigate potential overfitting. If the validation accuracy remained unchanged for five consecutive epochs, the training process was stopped. This method guarantees that the model shows strong generalisation when applied to new data, rather than just committing the training set to memory.

During the training process, I observed the performance of the model on the validation set. Refinements to the learning rate were implemented using a learning rate scheduler, which decreases the learning rate when the validation accuracy reaches a plateau, therefore enabling the convergence of the model towards an improved answer.

I then implemented model checkpointing to mitigate potential problems during training by saving model checkpoints at regular intervals. In the event of training interruption or deterioration in the model's performance, this enabled the reinstatement of the best-performing model.

**4.5 Problems Encountered**

**4.5.1 Hardware Limitations & Environment Changes**

The first obstacle I faced early on had to do with constraints in hardware capabilities. The first training of the model was conducted on a MacBook Pro, which, although equipped with a M1 Pro Chip, was not well-suited for the intense computational requirements of deep learning tasks. The training process was especially slow as a result of restricted GPU capabilities and frequent memory limitations, which had an impact on the model's performance and the duration of both training and testing.   
  
As a result of these constraints, it became clear to me that a transition to a different working environment that offered better computational power was necessary.  By changing to a more stable PC that allowed improved GPU processing, this change resulted in a significant decrease in training times and enabled extensive testing with the model. Even so, this change also brought out new challenges, such as compatibility concerns with software dependencies and the necessity to reinstall and reconfigure different libraries to ensure smooth operation of the code.

**4.5.3 Debugging Challenges**

Throughout the training phase, frequent issues kept occurring. They are detailed below:    
  
Mismatched input shapes created an ongoing problem in matching the input forms with the expected dimensions of the VGG-16 model. Errors were frequent when the data preparation was inconsistent with the anticipated input of the model, needing repetitive debugging and changes to the preprocessing process.   
  
Additionally, the migration between environments and the installation of important libraries often ended up in compatibility problems, such as inconsistencies in versions and a lack of certain modules, which caused delays and needed constant debugging.  
  
The process of debugging these problems was tedious and needed a solid understanding of both the data flow and the basic framework of the model. The iterative nature of these changes highlighted the need of adopting an adaptable approach and being ready to make further modifications to both the data and model settings.

**Chapter 5 – Results**

**5.1 Training & Validation Performance**

Thorough monitoring of the training process was carried out, including real-time display of accuracy and loss data for both the training and validation sets. The initial training phase, which employed the VGG-16 model without fine-tuning, demonstrated significant overfitting, marked by a high level of accuracy during training but a low level of accuracy during validation. The difference mostly was caused by the model committing  training data to memory without properly generalising to new validation data.

Beginning with an initial accuracy of around 0.00%, the accuracy steadily increased to a peak of 59.4% at the end of the training process.  
The validation accuracy demonstrated some fluctuations, starting as initially low but rising higher.

The training loss exhibited a continuous decline across epochs, suggesting the successful gathering of the training data.

By using data augmentation techniques such as rotation, zoom, and flips, the validation performance was improved. This was achieved by increasing the variety of training pictures, which allowed for the simulation of partial occlusions and different face positions.

**A screenshot of a graph

Description automatically generated**

Figure 5.1 - Graph representing the results of model training

**5.2 Fine-Tuning and Enhanced Model Performance**

In order for further re-training on our dataset, I performed fine-tuning by unfreezing additional layers of the VGG-16 model. The suggested approach utilised the pre-existing knowledge of VGG-16 and modified its deeper layers to enhance the recognition of partially visible faces.

**A close-up of a number

Description automatically generated**

Figure 5.2 - Fine-tuning accuracy percentage

Fine-tuning led to an improvement in the accuracy, with the final model achieving an accuracy of 50.16% on the test set. I believe I can refine and improve upon the accuracy of the model by the time of the project viva.

**Chapter 6 – Evaluation & Conclusion**

**6.1 Evaluation**

The VGG16 model, after fine-tuning, attained an accuracy of 50.16%, indicating an important improvement comparing to the basic model. This highlights the influence of fine-tuning the model using datasets customised for the particular task. This improvement demonstrates the potential of transfer learning, especially when developing a model for specific applications like partially visible face recognition. Despite the positive outcome, the degree of accuracy falls short of the normal requirements for deployment in real-life applications. The results suggest that although VGG16 is efficient, the identification of partially visible faces continues to be a challenging task because of issues such as alignment, different facial orientations, and uneven illumination.  
  
A significant constraint on the project was the imbalance in the dataset. Certain classes had a limited number of images, especially those with partial visibility, which had a negative impact on the model's capacity to make generalisations across all categories. The difference created a bias in the training process, where the model showed better results on classes with much information while displaying lower results on classes that were not well represented. Potential future improvements may involve the addition of advanced data augmentation methods, such as geometric transformations or synthetic data generation, to further enhance the dataset and provide the model a broader range of training examples.  
  
The project faced several technological challenges, including frequent runtime failures and debugging problems, such as mismatched input shapes, data preprocessing inconsistencies, and configuration difficulties during model training. The resolution of these challenges required thorough debugging, frequent modifications to the model design, and a substantial commitment of work towards problem-solving. In addition, the training process was further complicated by hardware factors such as GPU availability and RAM limits, which required a transition to another working environment. Despite these challenges, regular improvements and an organised approach to correcting errors ensured continued growth and success of the project.

**6.2 Conclusion**

The aim of this project was to create an effective facial recognition model that is capable of recognising partially occluded faces. This system was built around the VGG-16 architecture and trained using the Labelled Faces in the Wild (LFW) dataset. With an accuracy of 50.16%, the fine-tuned model highlights the effectiveness of transfer learning in adapting models for specialised and demanding tasks.  
  
Furthermore, the project offered important insights into the difficulties and possibilities of facial recognition under poor conditions, emphasising the importance of customised model modifications and the promise of careful refining. Whilst I believe that the model can further be refined to improve the accuracy, overall, I have achieved an accuracy rate that I am satisfied with.

**References**

[1] W. Bledsoe and H. Chan, *A Man-Machine Facial Recognition System-Some Preliminary Results*. Palo Alto, California: Panoramic Research, Inc. , 1965.

[2] M. Turk and A. Pentland, “Eigenfaces for Recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Jan. 2021, doi: <https://doi.org/10.1162/jocn.1991.3.1.71>.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul. 1997, doi: <https://doi.org/10.1109/34.598228>.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2012.

[5] P. J. Phillips, Hyeonjoon Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000, doi: <https://doi.org/10.1109/34.879790>.

[6] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments,” 2008. Available: <http://tamaraberg.com/papers/lfw.pdf>

[7] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, doi: <https://doi.org/10.1109/cvpr.2015.7298682>.

[8] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *Computer Science*, 2014, doi: <https://doi.org/10.48550/arXiv.1409.1556>.

[9] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, “OpenFace: A general-purpose face recognition library with mobile applications,” *undefined*, 2016. <https://www.semanticscholar.org/paper/OpenFace%3A-A-general-purpose-face-recognition-with-Amos-Ludwiczuk/82e66c4832386cafcec16b92ac88088ffd1a1bc9>