People's Democratic Republic of Algeria Ministry of Higher Education and Scientific Research

University of 20 août 1955-Skikda
Faculty of Sciences
Department of Computer Science

Master thesis
For obtaining the diploma of Master degree in Computer Science
Option: Artificial Intelligence- AI

Subject

TRANSFER LEARNING FOR PERSON IDENTIFICATION BASED ON FACIAL FEATURES

Presented by
BOUTAGHANE Imane
&
TADJER Yousra

Chairman	Dr. CHIKH Ramdane	University of 20 août 1955-Skikda
Reviewer	Dr. TAZIR Khaira	University of 20 août 1955-Skikda
Supervisor	Dr. HAZMOUNE Samira	University of 20 août 1955-Skikda

2023/2024

Acknowledgments

First and foremost, we express our deepest gratitude to Allah for His boundless blessings and for granting us the strength and perseverance to embark on this academic journey. We are profoundly grateful to our supervisor, Dr. HAZMOUNE Samira, whose unwavering support and guidance have been instrumental in shaping this thesis. Her expertise and encouragement have been invaluable throughout this endeavor. Additionally, we extend our heartfelt appreciation to the faculty members of the Computer Science department, whose dedication to teaching and scholarly guidance have enriched our academic experience over the past years. Their profound insights have broadened our understanding and laid a solid foundation for our future endeavors. We also extend our thanks to the esteemed members of the thesis committee for their time and expertise in evaluating our work. Their constructive feedback will undoubtedly contribute to the refinement of our research. Finally, we express our deepest gratitude to our families, especially our parents, and friends for their unwavering support and encouragement. Their love and encouragement have been our source of strength throughout this academic journey.

Dedication

To my loving parents, whose unwavering support and encouragement have been the foundation of my academic journey. Your sacrifices and belief in my abilities have propelled me forward, and I *am forever grateful.*

To my dear sisters and brothers, your constant encouragement and understanding have been a source of strength throughout this endeavor. Your presence in my life is a blessing that I cherish deeply.

To my dedicated binome, tadjer yousra, your collaboration and friendship have made this journey both rewarding and enjoyable. Your insights and shared experiences have enriched my understanding and contributed significantly to the completion of this thesis.

To all my friends, near and far, your support and camaraderie have lifted my spirits during the challenging times and made the joyful moments even brighter.

Finally, I dedicate this thesis to the resilient people of Palestine, whose courage and perseverance in the face of adversity inspire me deeply. May their struggle for justice and freedom be acknowledged and may peace prevail in their homeland."

.....BOUTAGHANE Imane

Dedication

To my beloved parents, whose unwavering love and unwavering support have formed the bedrock of my academic journey. Your enduring sacrifices and steadfast belief in my capabilities have propelled me forward, shaping the person I am today. I am eternally grateful for your guidance and encouragement.

To my cherished siblings, your unwavering support and understanding have been a constant source of strength throughout this arduous journey. Your presence in my life fills it with joy and warmth, and I am profoundly grateful for the bond we share.

To my dedicated partner, boutaghane imane, your unwavering collaboration and unwavering friendship have made this journey both enriching and enjoyable. Your profound insights and shared experiences have broadened my perspective and played an integral role in the completion of this thesis.

To all my dear friends, near and far, your unwavering support and camaraderie have illuminated my path during the darkest moments and amplified the joyous occasions. Your presence in my life is a blessing I cherish deeply.

Lastly, I dedicate this thesis to the resilient people of Palestine, whose unwavering courage and indomitable spirit in the face of adversity serve as an enduring source of inspiration. May their ongoing struggle for justice and freedom be recognized, and may peace reign in their homeland, illuminating their path towards a brighter future. Ya Allah, grant the people of Palestine strength, patience, and ultimate liberation.

.....TADJER Yousra

Abstract

In this master's thesis, our primary focus revolves around person identification from facial images, a field deemed immensely significant across various sectors. This technology is particularly crucial in law enforcement for criminal identification, in healthcare for patient verification and personalized treatment, and in financial institutions for secure transactions and access control.

The precision and efficiency of facial recognition systems hold paramount importance in averting security breaches and ensuring dependable identification processes. Harnessing the strides made in artificial intelligence, particularly in the realm of computer vision, serves to augment the efficacy of these systems.

Our study embarks on an experimental journey, meticulously comparing a spectrum of pre-trained models through transfer learning, encompassing diverse architectures of Convolutional Neural Networks (CNNs) and Vision Transformer (ViT) models, with the aim of pinpointing the most optimal model for facial recognition task. Specifically, we investigate models such as EfficientNet, DenseNet, ResNet-50, VGG16, and MobileNet among the CNNs, alongside ViT-B/8, ViT-S/16, and ViT-L/16 among the ViT models.

Our methodology hinges on the utilization of two pivotal datasets: The widely recognized Labeled Faces in the Wild (LFW) dataset, a staple in facial recognition research, and the Pins dataset, housing images of renowned personalities. The process of fine-tuning these pre-trained models on these datasets acts as a catalyst in optimizing their performance. The findings are immensely encouraging, with the EfficientNet model exhibiting an unparalleled classification accuracy of 100% on the LFW dataset and 94.12% on the Pins-FR dataset. These results underscore the exceptional performance prowess of EfficientNet in the field of facial recognition, eclipsing both the other CNN models and the ViT models subjected to testing.

Keywords: Facial recognition, Person identification, Transfer learning, Pre-trained models, CNN, ViT, EfficientNet, DenseNet, ResNet-50, VGG16, MobileNet.

Résumé

Dans cette thèse de master, notre objectif principal porte sur l'identification des personnes à partir d'images faciales, un domaine considéré comme extrêmement important dans divers secteurs. Cette technologie est particulièrement cruciale dans l'application de la loi pour l'identification des criminels, dans les soins de santé pour la vérification des patients et les traitements personnalisés, ainsi que dans les institutions financières pour les transactions sécurisées et le contrôle d'accès.

La précision et l'efficacité des systèmes de reconnaissance faciale revêtent une importance capitale pour prévenir les violations de sécurité et assurer des processus d'identification fiables. Exploiter les avancées réalisées en intelligence artificielle, en particulier dans le domaine de la vision par ordinateur, permet d'améliorer l'efficacité de ces systèmes.

Notre étude entreprend une étude expérimentale, comparant méticuleusement un éventail de modèles pré-entraînés à travers l'apprentissage par transfert (Transfer Learning), englobant diverses architectures de réseaux de neurones convolutifs (CNN) et de modèles Vision Transformer (ViT), dans le but de déterminer le modèle le plus optimal pour la tâche de reconnaissance faciale. Plus précisément, nous examinons des modèles tels que EfficientNet, DenseNet, ResNet-50, VGG16, et MobileNet parmi les CNN, ainsi que ViT-B/8, ViT-S/16, et ViT-L/16 parmi les modèles ViT.

Notre méthodologie repose sur l'utilisation de deux ensembles de données essentiels : Le célèbre ensemble de données Labeled Faces in the Wild (LFW), une référence dans la recherche en reconnaissance faciale, et l'ensemble de données Pins, contenant des images de personnalités renommées. Le processus de fine-tuning de ces modèles pré-entraînés sur ces ensembles de données agit comme un catalyseur pour optimiser leurs performances. Les résultats sont extrêmement encourageants, avec le modèle EfficientNet affichant une précision de classification inégalée de 100% sur l'ensemble de données LFW et de 94.12% sur l'ensemble de données Pins-FR. Ces résultats soulignent la performance exceptionnelle d'EfficientNet dans le domaine de la reconnaissance faciale, surpassant à la fois les autres modèles CNN et les modèles ViT testés.

Mots-clés: Reconnaissance faciale, Identification des personnes, Apprentissage par transfert, Modèles pré-entraînés, CNN, ViT, EfficientNet, DenseNet, ResNet-50, VGG16, MobileNet.

ملخص

في هذه المذكرة، يتركز اهتمامنا الأساسي حول التعرف على الأشخاص من صور الوجه، وهو مجال يُعتبر ذو أهمية بالغة عبر مختلف القطاعات. هذه التكنولوجيا مهمة بشكل خاص في تطبيق القانون للتعرف على المجرمين، وفي الرعاية الصحية للتحقق من هوية المرضى والعلاج الشخصي، وفي المؤسسات المالية للمعاملات الآمنة والتحكم في الوصول.

تُعتبر دقة وكفاءة أنظمة التعرف على الوجوه ذات أهمية قصوى في منع الخروقات الأمنية وضمان عمليات تحديد الهوية الموثوقة. إن الاستفادة من التقدم في الذكاء الاصطناعي، وخاصة في مجال رؤية الكمبيوتر، يساهم في تعزيز فعالية هذه الأنظمة.

تبدأ دراستنا برحلة تجريبية، تقوم بمقارنة منهجية بين مجموعة من النماذج المدربة مسبقاً من خلال التعلم الانتقالي، وتشمل مختلف معماريات الشبكات العصبية الالتفافية (CNN) ونماذج محولات الرؤية (ViT)، بهدف تحديد النموذج الأمثل لمهمة التعرف على الوجوه. على وجه الخصوص، نحقق في نماذج الـCNN مثل EfficientNet وDenseNet وResNet-50 وVGG16 وMobileNet، بالإضافة إلى نماذج الـ ViT مثل ViT-B/8 وViT-S/16 وViT-L/16.

تعتمد منهجيتنا على استخدام مجموعتي بيانات محوريتين: مجموعة (LFW) المعترف بها على نطاق واسع في أبحاث التعرف على الوجوه، ومجموعة بيانات Pins التي تحتوي على صور لشخصيات مشهورة. إن عملية تحسين أداء هذه النماذج المدربة مسبقاً على هذه المجموعات تعتبر بمثابة محفز لتحسين أدائها. النتائج مشجعة للغاية، حيث أظهر نموذج EfficientNet دقة تصنيف لا مثيل لها بلغت 100% على مجموعة بيانات LFW و94.12% على مجموعة بيانات Pins-FR. هذه النتائج تؤكد الأداء الاستثنائي لنموذج EfficientNet في مجال التعرف على الوجوه، متفوقاً على كل من نماذج الـCNN الأخرى ونماذج الـViT التي خضعت للاختبار.

الكلمات المفتاحية: التعرف على الوجوه، تحديد الهوية، التعلم الانتقالي، النماذج المدربة مسبقاً، الشبكات العصبية الالتفافية، محولات الرؤية، EfficientNet, DenseNet, ResNet-50, VGG16, MobileNet.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence.
ML	Machine Learning.
LR	Logistic Regression.
SVM	Support Vector Machine.
RBF	Radial Basis Function
DT	Decision Tree.
RF	Random Forest.
PCA	Principal Component Analysis.
KNN	k-Nearest Neighbors.
DL	Deep Learning.
CPU	Central Processing Units.
GPU	Graphics Processing Units.
ANN	Artificial Neural Networks.
ReLU	Rectified Linear Unit.
PReLU	Parametric ReLU.
MSE	Mean Square Error.
BCE	Binary Cross-Entropy.
RNN	Recurrent Neural Network.
LSTM	Long Short-Term Memory.
GPU	Gated Recurrent Unit.
CNN	Convolutional Neural Network.
NLP	Natural Language Processing.
ViT	Vision Transformers.
RGB	Red, Green, Blue.
LRN	Local response normalization.
DeiT	Data-efficient Image Transformers.
BEiT	BERT pre-training of Image Transformers.
MAE	Masked Auto Encoders.
SR	Speaker Recognition.
DNA	Deoxyribonucleic Acid.
STR	Short Tandem Repeat.

PCR	Polymerase Chain Reaction.
PC	Personal Computers.
DARPA	Defense Advanced Research Projects Agency.
FRGC	Face Recognition Grand Challenge.
MTCNN	Multi-Task Cascaded Convolutional Networks.
FRT	Facial Recognition Technology.
IBM	International Business Machines.
ID	Identification.
JSON	JavaScript Object Notation.
XML	Extensible Markup Language.
MS-Celeb-1M	Microsoft Celeb.
MS1M	Microsoft Celeb.
CelebA	CelebFaces Attributes.
IJB-A	IARPA Janus Benchmark A.
LFW	Labeled Faces in the Wild.
CVSAN	Convolutional Visual Self-Attention Network.
DS	Depthwise Separable (DS).
Cllr	Loglikelihood Ratio Cost.

GENERAL INTRODUCTION

Facial recognition technology has seen significant advancements over the past few decades, driven by improvements in computational power, artificial intelligence, and machine learning algorithms. This technology identifies or verifies individuals based on their facial features, making it applicable in various fields such as security, law enforcement, access control, and user authentication. Facial recognition systems involve several stages: image acquisition, face detection, feature extraction, and face recognition. The effectiveness of these systems is often evaluated based on their accuracy, speed, and robustness against variations in lighting, expression, and occlusion.

Despite the progress in facial recognition technology, achieving high accuracy across diverse and challenging datasets remains a critical issue. Traditional approaches to facial recognition often relied on hand-crafted feature extraction techniques and shallow learning models, which may not effectively capture the complex patterns and nuances present in facial data. With the advent of deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Transformers, significant advancements have been made in the field of facial recognition. These techniques demonstrated remarkable success in learning hierarchical representations from raw data, enabling accurate and robust feature extraction for various computer vision tasks, including facial recognition. However, training deep neural networks from scratch requires a substantial amount of labeled data and computational resources, which can be prohibitive in many real-world scenarios.

Transfer learning techniques offer a promising solution to these challenges by leveraging pre-trained models on large datasets, which are then fine-tuned for specific facial recognition tasks with smaller datasets. This approach overcomes the challenge of limited annotated data. Convolutional neural networks (CNNs) and, more recently, Vision Transformers (ViT), pre-trained on general image datasets, are often used. By reusing their lower layers to extract general features and adjusting the upper layers for specific tasks, these models achieve high accuracy even with limited data. This method improves robustness against variations in lighting, facial expressions, and occlusions, demonstrating the effectiveness and flexibility of transfer learning in facial recognition.

The objective of this study is to develop robust and accurate facial recognition system by investigating transfer learning with various pre-trained models, including different architectures of CNNs and ViTs. By doing so, the models can benefit from the learned features of the pre-trained models while adapting to the specific characteristics of the new data. The main contributions of our study are:

- Experimental tuning of the training hyper-parameters to identify the optimal settings for achieving the best accuracy in our facial recognition models on two different datasets: LFW and PINS-FR.
- Comparative study of different architectures of CNN pre-trained models (such as EfficientNet, DenseNet, ResNet-50, VGG16, and MobileNet) and ViT pre-trained models (such as ViT-B/8, ViT-S/16, and ViT-L/16) to determine the optimal model for our system.

This thesis is comprised of a general introduction, three chapters, and a general conclusion:

- Chapter 1: In this chapter, we will explore the foundations of traditional machine learning methods and their evolution towards transfer learning approaches. We will discuss the limitations of traditional methods and how transfer learning provides enhanced performance and efficiency.

- Chapter 2: This chapter focuses on the methods and techniques for person identification using facial features. We will present different modalities in facial recognition and analyze their effectiveness. Additionally, we will review the state-of-the-art datasets and recent works in the field, highlighting key advancements and challenges.
- Chapter 3: In this final chapter, we will detail our experimental study for facial recognition using various pre-trained models, including different architectures of CNNs and ViTs. We will describe the experimental setup, including the datasets used (LFW and PINS-FR), and the hyper-parameter tuning process. The chapter will conclude with a presentation of our experimental results, showcasing the performance improvements and the optimal settings that led to the highest accuracy.

CHAPTER 1

FROM TRADITIONAL MACHINE LEARNING TO TRANSFER LEARNING

1.1. INTRODUCTION

The transition from traditional machine learning to transfer learning represents a pivotal shift in AI methodologies. While traditional algorithms like decision trees and SVMs excel in well-defined tasks with abundant labeled data, they face limitations in adapting to complex, domain-specific challenges. Transfer learning introduces a transformative approach by leveraging knowledge across domains, optimizing learning in scenarios with limited data or resources. In this chapter, we explore this evolution, uncovering the principles and applications that place transfer learning at the forefront of modern AI solutions.

1.2. ARTIFICIAL INTELLIGENCE (AI)

Artificial Intelligence (AI), is a term coined by emeritus Stanford Professor John McCarthy in 1955, was defined by him as "the science and engineering of making intelligent machines". Much research has humans program machines to behave in a clever way, like playing chess, but, today, we emphasize machines that can learn, at least somewhat like human being do (Manning et al, 2020).

AI is the branch of computer science, which make the computers to mimic the human behavior to assist humans for better performance in the field of science and technology. Replicating human intelligence, solving knowledge-intensive tasks, building machines, which can perform tasks that require human intelligence, creating some system which can learn by itself are the few specific goals of AI. Machine learning and deep learning are two subsets of AI (**Erreur ! Référence non valide pour un signet.**) which are used to solve problems using high performance algorithms and multilayer neural network (Thirugnanam et al , 2021).

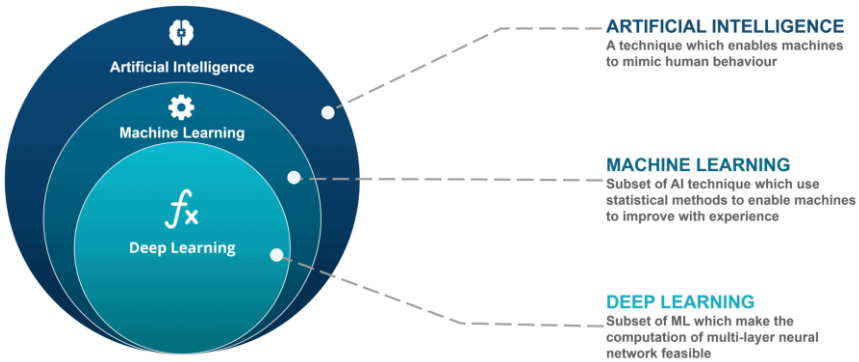

Figure 1. 1 : Relationship among artificial intelligence, machine learning and deep learning [Web-1].

1.3. MACHINE LEARNING (ML)

As humans can think, improve by self-improvement cycle, and learn from the past experiences, AI machines can also learn from the past experiences with the help of the concept known as Machine Learning (ML). The machine learning deals with the development of algorithms that enables the computer to learn from its data and past experiences on their own. In this method, the machine analyzes the available dataset which is also known as training data and with the help of the algorithms predict the

possible output over the given input. More the data (information) is provided, more perfect is the performance or prediction. In other words, the relationship between the data and efficiency is that the machine can improve its efficiency by gaining more and more data. It can learn from the data and improve automatically. This is very much helpful for dealing with huge data of complex problems, which are difficult for humans to deal with and they also consume more time in solving. In this process of computing the machine receives data as input and provides result using an appropriate algorithm (Thirugnanam et al, 2021).

1.3.1. TYPES OF MACHINE LEARNING

Based on the nature of the learning signal or response that the machine gets; machine learning can be classified into following categories:

1.3.1.1. SUPERVISED LEARNING

Supervised learning is typically the task of machine learning to learn a function that maps an input to an output based on sample input-output pairs. It uses labeled training data and a collection of training examples to infer a function. There are two main tasks of supervised learning: Classification, and Regression (Sarker et al, 2021).

- **Classification**

In classification tasks, the goal is to predict a discrete label or category for the input data. For example, classifying emails as spam or not spam. Many classification algorithms have been proposed in the machine learning and data science literature. In the following, we summarize the most common and popular methods that are used widely in various application areas (Sarker et al, 2021).

Logistic Regression (LR): Another common probabilistic based statistical model used to solve classification issues in machine learning is Logistic regression (LR). Logistic regression typically uses a logistic function to estimate the probabilities. It can overfit high-dimensional datasets and works well when the dataset can be separated linearly. The regularization (L1 and L2) techniques can be used to avoid over-fitting in such scenarios. The assumption of linearity between the dependent and independent variables is considered as a major drawback of Logistic Regression (Figure 1. 2).It can be used for both classification and regression problems, but it is more commonly used for classification (Sarker et al, 2021).

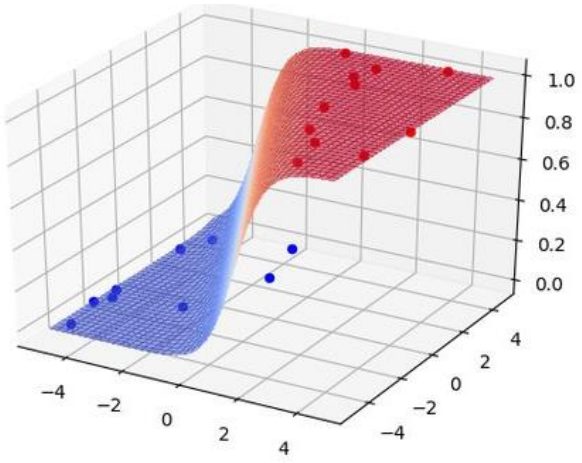

Figure 1. 2 : Logistic regression [Web-2].

Support Vector Machine (SVM): In machine learning, another common technique that can be used for classification, regression, or other tasks is a support vector machine. In high- or infinite-dimensional space, a support vector machine constructs a hyper-plane or set of hyper-planes. Intuitively, the hyper-plane, which has the greatest distance from the nearest training data points in any class, achieves a strong separation since, in general, the greater the margin, the lower the classifier's generalization error. It is effective in high-dimensional spaces and can behave differently based on different mathematical functions (Figure 1. 3) known as the kernel. Linear, polynomial, radial basis function (RBF), sigmoid, etc., are the popular kernel functions used in SVM classifier. However, when the data set contains more noise, such as overlapping target classes, SVM does not perform well (Sarker et al, 2021).

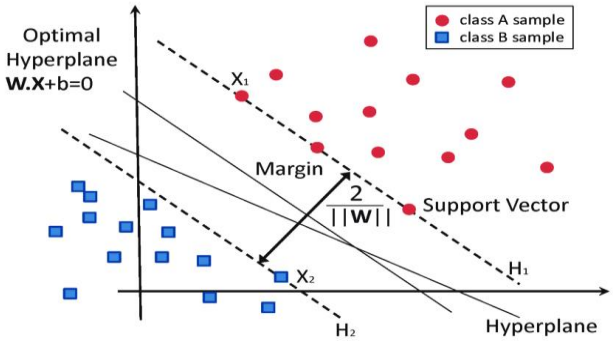

Figure 1. 3 : mathematics behind SVMs [Web-3].

Decision Tree (DT): Decision tree (DT) is a well-known non-parametric supervised learning method. DT learning methods are used for both the classification and regression tasks. ID3, C4.5, and CART are well known for DT algorithms. DT classifies the instances. Instances are classified by checking the attribute defined by that node, starting at the root node of the tree, and then moving down the tree branch corresponding to the attribute value (Figure 1. 4). For splitting, the most popular criteria are "gini" for the Gini impurity and "entropy" for the information gain that can be expressed mathematically as (Sarker et al, 2021).

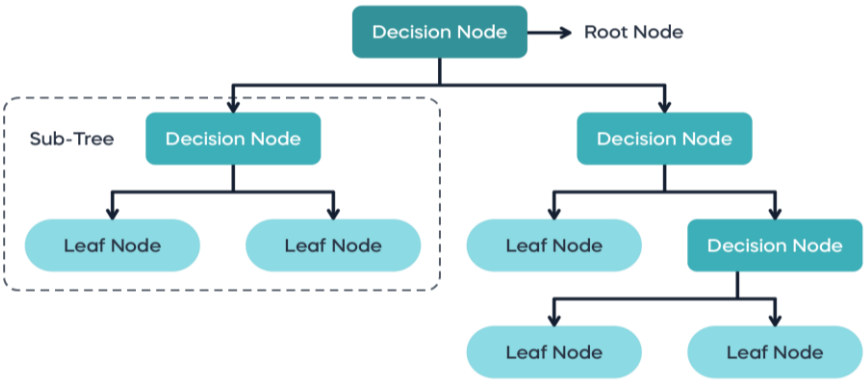

Figure 1. 4 : Decision tree [Web-4].

Random Forest (RF): A random forest classifier is well known as an ensemble classification technique that is used in the field of machine learning and data science in various application areas. This method uses "parallel ensembling" which fits several decision tree classifiers in parallel, on different data set sub-samples and uses majority voting or averages for the outcome or final result. It thus minimizes the over-fitting problem and increases the prediction accuracy and control. Therefore, the RF learning model with multiple decision trees is typically more accurate than a single decision tree based model (Figure 1. 5). To build a series of decision trees with controlled variation, it combines bootstrap aggregation (bagging) and random feature selection. It is adaptable to both classification and regression problems and fits well for both categorical and continuous values (Sarker et al, 2021).

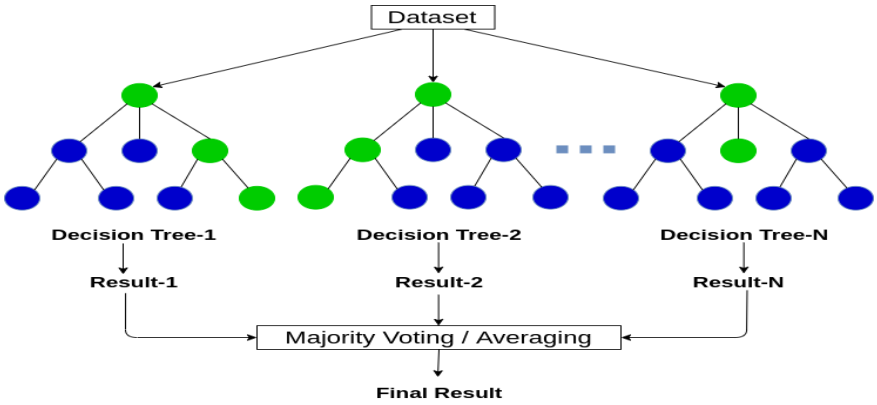

Figure 1. 5 : Random forest [Web-5].

Perceptron: Perceptron was introduced by Frank Rosenblatt in 1957. He proposed a Perceptron learning rule based on the original MCP neuron. A Perceptron is an algorithm for supervised learning of binary classifiers. This algorithm enables neurons to learn and processes elements in the training set one at a time. Perceptrons come in various types, each with its unique capabilities. A single-layer perceptron is limited to learning linearly separable patterns, meaning it can only effectively handle data that can be divided into two categories by a straight line or plane. On the other hand, multilayer perceptrons (MLPs) are more versatile. They consist of two or more layers of neurons, allowing them to learn complex patterns by processing information through multiple stages of transformation. This greater processing power enables MLPs to tackle nonlinear problems and learn intricate relationships within data that single-layer perceptrons cannot handle effectively. The perceptron is considered a fundamental building block in the development of neural networks, which we will further expand upon in the section on deep learning (Banoula et al, 2023).

- **Regression**

Regression analysis includes several methods of machine learning that allow to predict a continuous (y) result variable based on the value of one or more (x) predictor variables. Some of the familiar types of regression algorithms are linear, polynomial, lasso and ridge regression, etc., which are explained briefly in the following (Sarker et al, 2021).

Simple and multiple linear regression: This is one of the most popular ML modeling techniques as well as a well-known regression technique. In this technique, the dependent variable is continuous, the independent variable(s) can be continuous or discrete, and the form of the regression line is linear. Linear regression creates a relationship between the dependent variable Y and one or more independent

variables X (Figure 1. 6) (also known as regression line) using the best fit straight line (Sarker et al, 2021).

Polynomial regression: Polynomial regression is a form of regression analysis in which the relationship between the independent variable *x* and the dependent variable *y* is not linear (Figure 1. 6), but is thepolynomial degree of n^{th} (Sarker et al, 2021) .

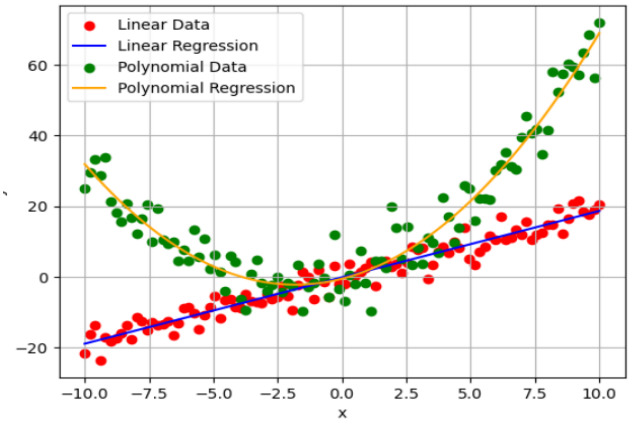

Figure 1. 6 : Polynomial regression Vs Linear regression [Web-6].

1.3.1.2. UNSUPERVISED MACHINE LEARNING

Unsupervised learning analyzes unlabeled datasets without the need for human interference. The most common unsupervised learning tasks are clustering, dimensionality reduction.

- **Clustering**

Clustering is an unsupervised machine learning technique for identifying and grouping related data points in large datasets without concern for the specific outcome (Figure 1. 7). It does grouping a collection of objects in such a way that objects in the same category, called a cluster, are in some sense more similar to each other than objects in other groups. It is often used as a data analysis technique to discover interesting trends or patterns in data. Clustering used four main algorithms including: K-means, fuzzy k-means, hierarchical clustering and mixture of Gaussians (Sarker et al, 2021).

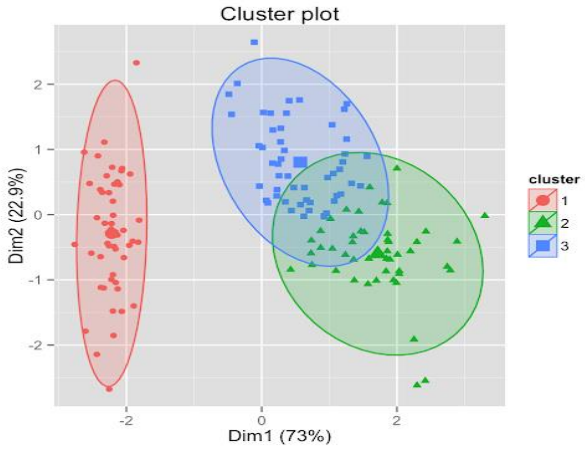

Figure 1. 7 : Concept of clustering [Web-7].

- **Dimensionality reduction and feature learning**

In machine learning and data science, high-dimensional data processing is a challenging task for both researchers and application developers. Thus, dimensionality reduction which is an unsupervised learning technique, is important because it leads to better human interpretations, lower computational costs, and avoids overfitting and redundancy by simplifying models.

1.3.1.3. SEMI-SUPERVISED LEARNING

Semi-supervised learning can be defined as a hybridization of the above-mentioned supervised and unsupervised methods, as it operates on both labeled and unlabeled data. Thus, it falls between learning "without supervision" and learning "with supervision". In the real world, labeled data could be rare in several contexts, and unlabeled data are numerous, where semi-supervised learning is useful. The ultimate goal of a semi-supervised learning model is to provide a better outcome for prediction than that produced using the labeled data alone from the model (Sarker et al, 2021).

1.3.1.4. REINFORCEMENT LEARNING

Reinforcement learning is a type of machine learning algorithm that enables software agents and machines to automatically evaluate the optimal behavior in a particular context or environment to improve its efficiency (Figure 1. 8).This type of learning is based on reward or penalty, and its ultimate goal is to use insights obtained from environmental activists to take action to increase the reward or minimize the risk (Sarker et al, 2021).

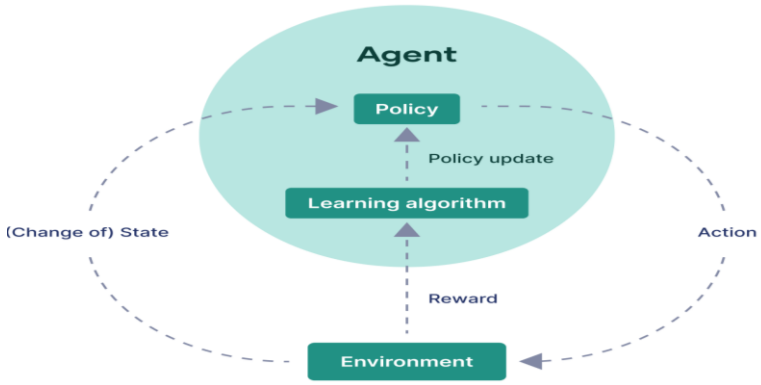

Figure 1. 8 : Reinforcement Learning Model [Web-8].

1.3.1.5. SUPERVISED LEARNING VS UNSUPERVISED LEARNING

Supervised and Unsupervised learning are two primary paradigms in machine learning, each with distinct approaches and goals in different scenario and with different datasets (Table 1. 1).

Table 1. 1 : Comparison between Supervised and unsupervised learning [Web-9].

Supervised learning	Unsupervised learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Model takes direct feedback to check if it is predicting correct output or not.	Model does not take any feedback.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, k-Nearest Neighbors (KNN), K-means Clustering, Principal Component Analysis (PCA), and Apriori algorithm
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in Clustering and Associations problems.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Is a simple method for machine learning, have more accuracy.	Need powerful tools for working with large amounts of unclassified data. Unsupervised learning models are computationally complex and have less accuracy, because they need a large training set to produce intended outcomes.
Ideal for spam detection, sentiment analysis, weather forecasting, pricing predictions and Optical Character Recognition.	Ideal for anomaly detection, recommendation engines, customer personas, medical imaging and Find a face in an image.

1.3.2. LIMITATIONS OF TRADITIONAL MACHINE LEARNING

Traditional machine learning techniques have been foundational in shaping the field of artificial intelligence. However, they come with their own set of challenges. Below are several hurdles faced by these classical methods:

Limited representation: Classical machine learning techniques often require handcrafted feature engineering, which can be time-consuming, labor-intensive, and may not capture the full complexity of the data, especially in tasks involving unstructured data such as images, text, or audio.

Scalability: Some classical algorithms struggle to scale efficiently with increasing data sizes or high-dimensional feature spaces. This limitation can lead to longer training times and higher resource requirements, making them less suitable for large-scale datasets.

Non-linearity: Many real-world datasets exhibit complex, non-linear relationships that classical algorithms may struggle to capture effectively. Linear classifiers, in particular, may fail to model intricate decision boundaries present in such data.

Generalization: Overfitting is a common challenge in classical machine learning, where models trained on specific datasets may fail to generalize well to unseen data. This issue can arise due to model complexity, inadequate regularization, or insufficient training data.

Interpretability: While some classical algorithms offer interpretability to some extent, understanding the reasoning behind model predictions can be challenging, especially for complex models or when dealing with high-dimensional data. This lack of interpretability can hinder trust and adoption in certain applications.

Feature engineering: Classical machine learning often relies on human expertise to manually design relevant features. However, identifying and selecting informative features can be subjective and may overlook important patterns present in the data, leading to suboptimal performance.

Imbalanced data: Class imbalance, where one class is significantly more prevalent than others, can pose challenges for classical algorithms. Biased models may favor the majority class and perform poorly on minority classes, leading to inaccurate predictions and reduced model effectiveness.

Noise sensitivity: Classical algorithms may be sensitive to noise or outliers in the data, leading to degraded performance or inaccurate predictions, especially in datasets with significant noise or data quality issues.

Addressing these challenges has been a driving force behind the development of more advanced techniques, such as deep learning, which will be explained in the next section.

1.4. DEEP LEARNING (DL)

In the last few years, the Deep Learning (DL) computing paradigm has been deemed he Gold Standard in the Machine Learning (ML) community. Moreover, it has gradually become the most widely used computational approach in the field of ML, thus achieving outstanding results on several complex cognitive tasks, matching or even beating those provided by human performance. One of the benefits of DL is the ability to learn massive amounts of data. The DL field has grown fast in the last few years and it has been extensively used to successfully address a wide range of traditional applications. More importantly, DL has outperformed well-known ML techniques in many domains such as cyber security, natural language processing, bioinformatics, robotics and control, and medical information processing, among many others (Alzubaidi et al, 2021).

1.4.1. DEEP LEARNING VS. MACHINE LEARNING

Deep learning and machine learning are two interrelated fields within artificial intelligence (AI) that have revolutionized various industries. There are many differences between DL and ML, as shown in the following (Table 1. 2).

Table 1. 2: Deep Learning vs. Machine Learning (Mohamed taye et al, 2023).

Deep Learning	Machine Learning
Uses artificial neural network architecture to learn the hidden patterns and relationships in the dataset.	Apply statistical algorithms to learn the hidden patterns and relationships in the dataset.
Requires the larger volume of dataset compared to machine learning	Can work on the smaller amount of dataset
Better for complex task like image processing, natural language processing, etc.	Better for the low-label task.
Takes more time to train the model.	Takes less time to train the model.
Relevant features are automatically extracted from images. It is an end-to-end learning process.	A model is created by relevant features which are manually extracted from images to detect an object in the image.
More complex and less interpretable, because of involving multiple layers of nonlinear transformations.	Provide more interpretable results because they rely on explicit rules or models that can be easily understood.
It requires a high-performance computer with GPU.	It can work on the CPU or requires less computing power as compared to deep learning.

Considering the fact that Deep learning is the next step in the evolution of Machine learning instilling machines how to make their decisions accurately without the intervention of the human expert.

1.4.2. BENEFITS OF DEEP LEARNING

Deep learning has enabled the creation of intelligent systems capable of perceiving, comprehending, and interacting with the world in ways that were previously considered impossible. Applications include self-driving cars, virtual assistants, personalized recommendations, and advanced robotics. This is clarified in the following factors [Web-10].

Automatic feature learning: Deep learning algorithms can automatically learn features from the data, which means that they don't require the features to be hand-engineered. This is particularly useful for tasks where the features are difficult to define, such as image recognition.

Handling large and complex data: Deep learning algorithms can handle large and complex datasets that would be difficult for traditional machine learning algorithms to process. This makes it a useful tool for extracting insights from big data.

Improved performance: Deep learning algorithms have been shown to achieve state-of-the-art performance on a wide range of problems, including image and speech recognition, natural language processing, and computer vision.

Handling non-linear relationships: Deep learning can uncover non-linear relationships in data that would be difficult to detect through traditional methods.

Handling structured and unstructured data: Deep learning algorithms can handle both structured and unstructured data such as images, text, and audio.

Predictive modeling: Deep learning can be used to make predictions about future events or trends, which can help organizations plan for the future and make strategic decisions.

Handling missing data: Deep learning algorithms can handle missing data and still make predictions, which is useful in real-world applications where data is often incomplete.

Handling sequential data: Deep learning algorithms such as Recurrent Neural Networks (RNNs) and Long Short-term Memory (LSTM) networks are particularly suited to handle sequential data such as time series, speech, and text. These algorithms have the ability to maintain context and memory over time, which allows them to make predictions or decisions based on past inputs.

Scalability: Deep learning models can be easily scaled to handle an increasing amount of data and can be deployed on cloud platforms and edge devices.

Generalization: Deep learning models can generalize well to new situations or contexts, as they are able to learn abstract and hierarchical representations of the data.

1.4.3. ARTIFICIAL NEURAL NETWORKS (ANN)

Virtually every deep learning algorithm, which relies on Artificial Neural Networks (ANN), ANN are based on the structure and function of human neurons (Figure 1. 9). It's also referred to as neural networks or neural nets. The input layer, or first layer, of an artificial neural network gets data from external sources and forwards it to the hidden layer, or second layer. Each neuron in the hidden layer gets information from the neurons in the previous layer, computes the weighted total, and then transfers it to the neurons in the next layer. These connections are weighted, which means that the impacts of the inputs from the preceding layer are more or less optimized by giving each input a distinct weight. These weights are then adjusted during the training process to enhance the performance of the model. The ANN work is based on three layers: input layer, hidden layer, and output layer [Web-11].

Input layer: This layer accepts input features. It provides information from the outside world to the network, no computation is performed at this layer, nodes here just pass on the information (features) to the hidden layer.

Hidden layer: Nodes of this layer are not exposed to the outer world, they are part of the abstraction provided by any neural network. The hidden layer performs all sorts of computation on the features entered through the input layer and transfers the result to the output layer.

Output layer: This layer bring up the information learned by the network to the outer world.

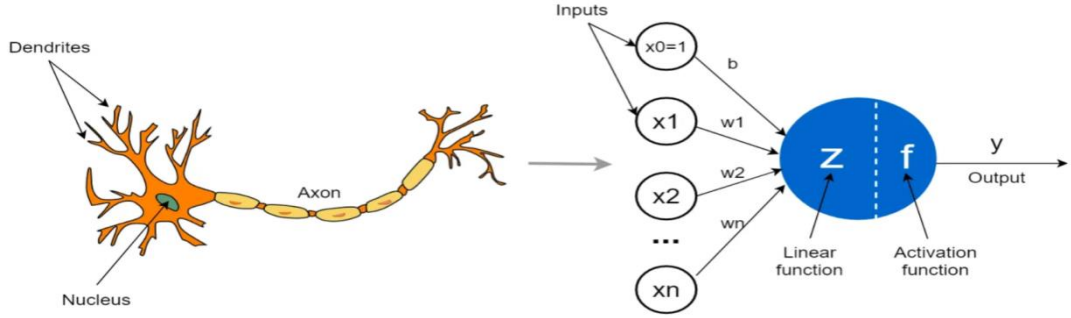

Figure 1. 9 : The concept of ANN [Web-11].

In the working of an ANN, there are two fundamental steps, which, forward-propagation and Back-propagation.

Step 1: Forward-propagation

Forward propagation is a crucial process in neural networks where input data is passed through the network to generate predictions or outputs. It begins with the input layer, where each feature is represented by a node that receives input data. The connections between nodes are assigned weights, indicating the strength of the connection. As the network undergoes training, these weights are adjusted. Moving through the hidden layers, each neuron processes inputs by multiplying them by weights, summing them up, and then applying an activation function. This introduces non-linearity, allowing the network to learn intricate patterns in the data. This process is repeated until the output layer is reached, where the final result or prediction is generated based on the transformed input data. Through forward propagation, neural networks can effectively learn and make predictions on complex datasets.

Step 2: Back-propagation

In the back-propagation phase of neural network training, several key steps are undertaken to refine the network's performance and improve its accuracy. Firstly, the network's output is compared to the desired target values, and a loss function is employed to quantify the disparity between them. This loss calculation guides the network in understanding its errors and areas for improvement. Following this, gradient descent optimization is employed to minimize the loss. By computing the gradient of the loss function with respect to each weight, the network can iteratively adjust its weights to move towards a configuration that minimizes the error. This adjustment of weights occurs across the network through an iterative process known as back-propagation, wherein the gradients are propagated backward from the output layer to the input layer. Throughout training, which involves exposing the network to various data samples, this cycle of forward propagation, loss calculation, and back-propagation is repeated iteratively. This iterative training process enables the network to adapt and learn intricate patterns inherent in the data. Additionally, activation functions play a crucial role in introducing non-linearity to the model. Functions like the rectified linear unit (ReLU) or sigmoid determine whether a neuron should activate based on the weighted input it receives, thus enhancing the network's ability to capture complex relationships within the data [Web-12].

1.4.3.1. ACTIVATION FUNCTION

An Activation Function decides whether a neuron should be activated or not. This means that it will decide whether the neuron’s input to the network is important or not in the process of prediction using simpler mathematical operations [Web-13].

The Rectified Linear Unit (ReLU) function: commonly known as ReLU, is a fundamental activation function used in neural networks. Despite its name suggesting linearity, ReLU possesses a derivative function, enabling back-propagation and computational efficiency. A notable characteristic of ReLU is its selective activation of neurons, where neurons are deactivated if the output of the linear transformation is less than 0. This feature contributes to the sparsity of activations and aids in addressing the vanishing gradient problem often encountered in deep learning models. The mathematical equation is as follows:

$$f(x) = \max(0, x) \tag{1.2}$$

Leaky ReLUFunction: LeakyReLU is an improved version of ReLU function to solve the Dying ReLU problem as it has a small positive slope in the negative area. The mathematical equation is as follows:

$$f(x) = \max(0.1x, x) \tag{1.3}$$

Parametric ReLU: PReLU is another variant of ReLU that aims to solve the problem of gradient's becoming zero for the left half of the axis. This function provides the slope of the negative part of the function as an argument *a*. By performing back-propagation, the most appropriate value of *a*is learnt. The mathematical equation is as follows:

$$f(x) = \max(ax, x) \tag{1.3}$$

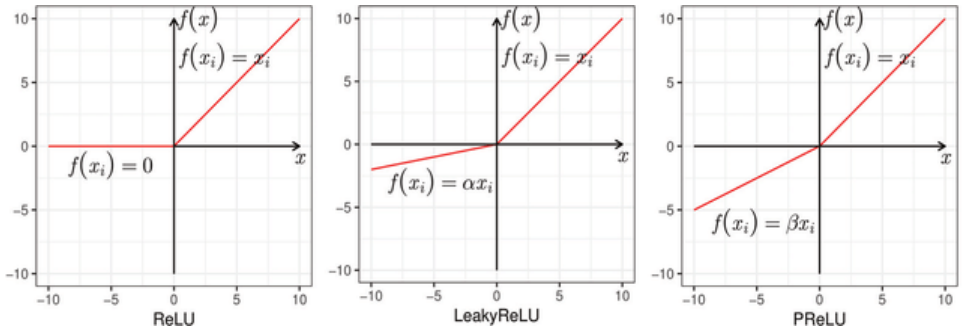

Figure 1. 10 : ReLU, PReLU, and Leaky ReLUActivation [Web-14].

1.4.3.2. LOSS FUNCTION

The loss function, also referred to as the error function, is a crucial component in machine learning that quantifies the difference between the predicted outputs of a machine learning algorithm and the actual target values. Loss functions in machine learning can be categorized based on the machine learning tasks to which they are applicable. Most loss functions apply to regression and classification machine learning problems. There are many loss functions but we'll focus on two crucial ones [Web-15]:

The Mean Square Error (MSE): Also known as L2 loss, is a loss function commonly used for regression problems. It measures the error magnitude between a machine learning model's prediction and the actual output. The MSE calculates the average of squared differences between predictions and target values. The mathematical equation for Mean Square Error (MSE) is as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2 \tag{1.4}$$

Where: n is the number of samples, y_i is the predicted value,$\overline{y}$ is the target value for the i-th sample.

Binary Cross-Entropy loss (BCE): Is a performance measure for classification models that outputs a prediction with a probability value typical between 0 and 1, and this prediction value corresponds to the likelihood of a data sample belonging to class or category. In the case of Binary Cross-Entropy Loss, there are two distinct classes. But notably, a variant of cross-entropy loss, Categorical Cross-Entropy applies to multiclass classification scenarios [Web-15]. The mathematical equation for Binary Cross-Entropy loss (BCE) is as follows:

$$L(y, f(x)) = -[y * \log(f(x)) + (1 - y) * \log(1 - f(x))] \tag{1.5}$$

Where: L is the Binary Cross-Entropy Loss function, y is the true binary label, f(x) is the predicted probability of the positive class.

1.4.4. RECURRENT NEURAL NETWORK (RNN)

RNNs are a commonly employed and familiar algorithm in the discipline of DL, RNN is mainly applied in the area of speech processing and NLP contexts unlike conventional networks, RNN uses sequential data in the network. Since the embedded structure in the sequence of the data delivers valuable information, this feature is fundamental to a range of different applications. For instance, it is important to understand the context of the sentence in order to determine the meaning of a specific word in it. It is possible to consider the RNN as a unit of short-term memory, where x represents the input layer, y is the output layer and s represents the state (hidden) layer (Figure 1. 11). RNN's sensitivity to the exploding gradient and vanishing problems represent one of the main issues with this approach. More specifically, during the training process the reduplications of several large or small derivatives may cause the gradients to exponentially explode or decay. With the entrance of new inputs the network stops thinking about the initial ones, therefore this sensitivity decays over time. This issue can be handled using Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) (Alzubaidi et al, 2021).

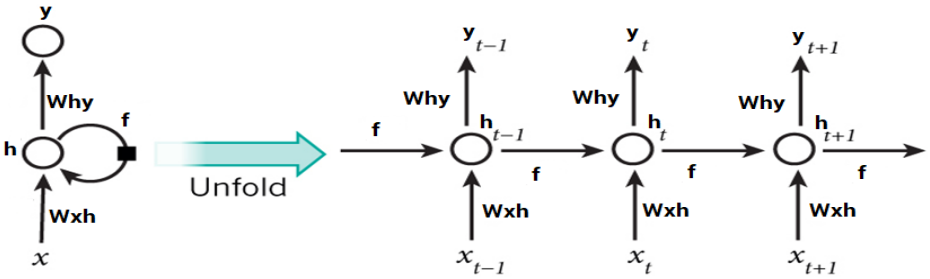

Figure 1. 11 : RNN basic architecture [Web-16].

Long Short-Term Memory (LSTM): Long Short-Term Memory works on the read-write-and-forget principle where given the input information network reads and writes the most useful information from the data and it forgets about the information which is not important in predicting the output. For doing this three new gates are introduced in the RNN. In this way, only the selected information is passed through the network. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates (Figure 1. 12). LSTM has three primary gates: the forget gate, input gate, and output gate. The forget gate is responsible for removing information from the cell state. It decides what information from the previous cell state should be discarded. The input gate is responsible for the addition of information to the cell state, regulating the inflow of new information that needs to be stored. The output gate takes the current input, the previous short-term memory, and the newly computed long-term memory to produce new short-term memory, which will be passed on to the cell in the next time step [Web-17].

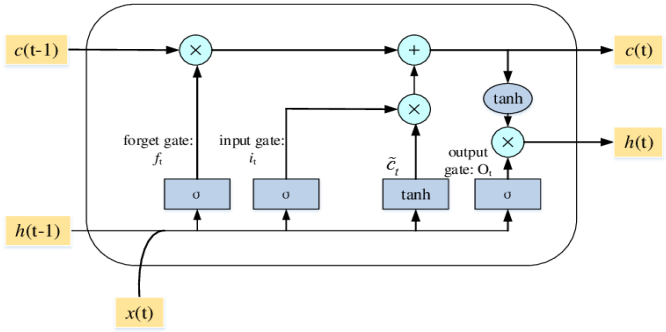

Figure 1. 12 : LSTM architecture [Wweb-18].

Gated Recurrent Unit (GRU): GRUs are improved version of standard recurrent neural network. To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called, update gate and reset gate. Basically, these are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction. The two principle gates in this context are the update gate and the reset gate (Figure 1.13). The update gate is responsible for determining the amount of previous information that needs to pass along to the next state. On the other hand, the reset gate is used by the model to decide how much of the past information needs to be neglected; in short, it decides whether the previous cell state is important or not. [Web-19].

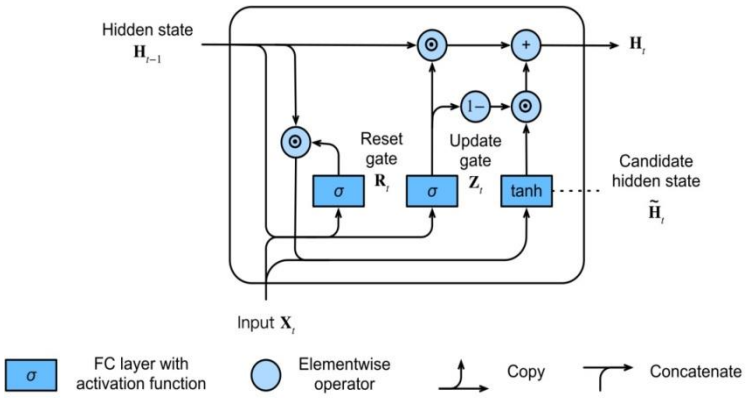

Figure 1. 13 : GRU architecture [Web-20].

1.4.5. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network (CNN) is an extension of artificial neural networks (ANN) that is mostly used to extract features from grid-like matrix datasets. For example, consider visual datasets such as images or videos, in which data patterns play a significant role. CNN comprises of several layers (Figure 1. 14), including the input layer, the convolutional layer, the pooling layer, and the fully connected layer.

Input layers: It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3(example of image of dimension 32 x 32 x 3).

Convolutional layers: This is the layer, which is used to extract the feature from the input dataset .It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices. It slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred ad feature maps.

Activation layer: By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. It will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max (0, x), Tanh, Leaky RELU, etc.

Pooling layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling.

Flattening: The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.

Fully Connected Layers: It takes the input from the previous layer and computes the final classification or regression task.

Output Layer: The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

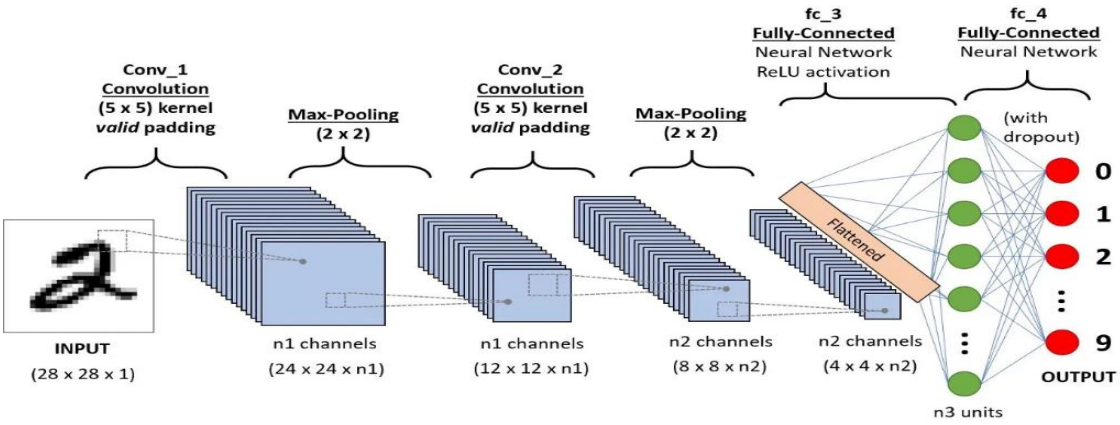

Figure 1. 14 : CNN architecture [Web-21].

1.4.6. TRANSFORMERS

In deep learning, transformers refer to a specific type of architecture that has gained immense popularity, primarily in natural language processing (NLP) tasks. The transformer architecture, introduced in the "Attention is All You Need" paper by Vaswani, in 2017, revolutionized how sequential data [Web-22].

Transformers represent a breakthrough in deep learning architecture, particularly in the realm of natural language processing (NLP). Their success stems from several key components.

Self-Attention mechanism: The core innovation of transformers is the self-attention mechanism, which allows the model to weigh the importance of different elements in the input sequence when encoding or decoding. Self-attention computes attention scores between all pairs of elements in the sequence, enabling the model to focus on relevant information dynamically.

Positional encoding: Since transformers do not inherently understand the order of elements in a sequence, positional encodings are added to the input embeddings to provide information about the positions of elements.

Encoder-Decoder architecture: Transformers typically consist of an encoder and a decoder. The encoder processes input sequences, while the decoder generates output sequences. Each encoder and decoder layer contains multiple self-attention mechanisms and feed-forward neural networks.

Feed-forward neural networks: In addition to self-attention layers, transformers contain feed-forward neural networks that process the output of the attention mechanisms independently for each position in the sequence [22].

1.4.6.1. APPLICATION OF TRANSFORMERS IN DEEP LEARNING

Transformers have found numerous applications in the field of deep learning.

Natural Language Processing (NLP): Transformers are extensively utilized across various Natural Language Processing (NLP) tasks. In language modeling, they play a pivotal role in tasks like next-word prediction and text generation, leveraging their ability to understand and generate coherent sequences

of text. Transformers have made amazing breakthroughs in machine translation, giving cutting-edge performance by effectively translating across multiple languages. They accomplish this by capturing nuanced linguistic nuances and context. Furthermore, Transformers are instrumental in question-answering systems, where they analyze both questions and contextual passages to produce accurate responses. In text classification tasks such as sentiment analysis, document categorization, and spam detection, Transformers excel in understanding and classifying textual data, thereby enhancing decision-making processes.

Audio processing: Transformers have been explored in the domain of speech recognition tasks, where they process audio spectrograms or other speech signal representations. Their adaptability beyond text-based applications offers promising avenues for enhanced accuracy and efficiency in transcribing and interpreting speech. Leveraging their robust architecture and attention mechanisms, Transformers enable the extraction of meaningful features from speech data. This capability positions them as valuable tools for addressing the challenges of speech recognition, with ongoing research poised to further advance their contributions to this field. We will explain an example of audio processing transformers.

Sequence to sequence tasks: Transformers demonstrate exceptional proficiency in sequence-to-sequence tasks, encompassing endeavors like text summarization, machine translation, and image captioning. Their adeptness lies in efficiently capturing long-range dependencies and contextual information inherent in sequences. With their robust architecture and attention mechanisms, Transformers adeptly handle the complexities of these tasks, enabling accurate and coherent generation of output sequences based on input sequences. This versatility positions Transformers as invaluable tools for various applications requiring the transformation or synthesis of sequential data across diverse domains.

Recommendation system: Transformers have found application in recommendation systems, particularly in modeling user-item interactions and discerning intricate patterns within user behavior data. Their utilization stems from their ability to efficiently process and understand vast amounts of interaction data, facilitating accurate predictions and personalized recommendations. By leveraging their robust architecture and attention mechanisms, Transformers excel at capturing the nuanced relationships between users and items, thereby enhancing the effectiveness of recommendation systems. This adaptation underscores the versatility of Transformers in addressing diverse challenges across recommendation tasks and optimizing user experiences in various domains.

Computer vision: In the realm of Computer Vision, vision transformers (ViT) have emerged as a potent tool for image classification tasks. This neural network architecture introduces a transformative approach by converting images into sequences of vectors. By doing so, it facilitates the exploration and capture of long-term relationships among various parts of the image. This innovative methodology harnesses the power of self-attention mechanisms to analyze intricate visual features, enabling the network to make robust and accurate classifications. Vision transformers represent a significant advancement in image understanding, offering a versatile solution for a wide range of computer vision applications [Web-23].

1.4.6.2. VISION TRANSFORMERS (VITS)

Transformers have been adapted for image classification tasks, replacing convolutional neural networks (CNNs) in some cases. The Vision Transformer, is a model for image classification that employs a Transformer-like architecture over patches of the image (Figure 1. 15). An image is split into fixed-size patches, each of them are then linearly embedded, position embeddings are added, and the resulting sequence of vectors is fed to a standard Transformer encoder. In order to perform classification, the

standard approach of adding an extra learnable "classification token" to the sequence is used (Dosovitskiy et al, 2020).

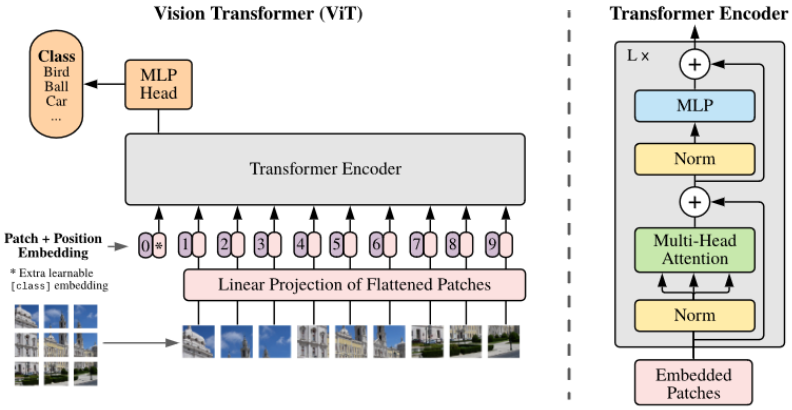

Figure 1. 15 : The working of ViT (Dosovitskiy et al, 2020).

The Vision Transformer (ViT) is a cutting-edge development in computer vision research, showcasing the power of transformer-based architectures in image understanding. It processes images by breaking them into fixed-size patches and using a transformer encoder to capture global dependencies among these patches. ViT leverages self-attention mechanisms to extract meaningful spatial relationships and context from images, enabling it to produce predictions for image classification tasks. ViT has achieved state-of-the-art performance in computer vision benchmarks, demonstrating its effectiveness and interpretability in analyzing visual data.

1.5. TRANSFER LEARNING

Conventional machine learning approaches make future predictions based on the statistical models, trained on previously collected, labeled, or unlabeled data. An approach that utilizes the labeled data in the process of model training is most commonly referred to as supervised learning, while the utilization of unlabeled data in the process of model training is commonly known as unsupervised or self-supervised training. When dealing with small, insufficient sets of labeled data, building a good classifier is a hard and burdensome task. Many studies have been conducted to tackle this issue, utilizing semi-supervised training or some variation of such an approach where the usage of a large unlabeled set of data and a small set of labeled data is combined. The most common issue with such approaches is the assumption that the labeled and unlabeled data distributions are the same. In contrast to semi-supervised approaches, transfer learning enables the domains, tasks and data distributions to be different.

Transfer learning also gained momentum, due to the requirements common to all deep neural networks (the need for large datasets). Additionally, the process of training deep neural network architectures requires a lot of computational power and thus is a time consuming task. In such cases, with the utilization of the transfer learning techniques, one could benefit significantly in terms of time complexity as well as in terms of the large, required dataset [Web-24].

In general, transfer learning techniques are used in two ways, one being the approach where the weights of the pre-trained model are preserved (frozen) on some of the layers and fine-tuned (trained) in the remaining layers, and the other being the approach where the pre-trained deep neural network is utilized as a feature extractor, while the extracted features are fed to the classifier of choice.

1.5.1. PRE-TRAINING

Pre-trained models are particularly valuable in machine learning because they offer a shortcut for leveraging the knowledge gained from extensive training on large datasets. Rather than starting from scratch, typically on a large-scale image-classification task. For example, in computer vision, models like VGG, ResNet or EfficientNet are often pre-trained on datasets like ImageNet, which contains millions of labeled images across thousands of categories.

Alongside these architectures, pre-trained models encapsulate learned weights, representing a repository of valuable knowledge and representations gleaned from the pre-training process. This pre-training typically involves exposure to large-scale datasets through supervised or self-supervised learning techniques, exemplified in computer vision by models like ResNet trained on the ImageNet dataset for image classification. Throughout pre-training, the model acquires the capability to extract hierarchical features from raw input data, refining its understanding of relevant patterns and structures pertinent to the task at hand.

A distinctive characteristic of pre-trained models lies in the transferability of the features they learn, transcending specific tasks and domains. The layers of a pre-trained model encode generalizable patterns and representations, offering potential utility across a spectrum of related tasks and domains. For instance, features extracted by a pre-trained convolutional neural network (CNN) for image classification possess applicability in diverse scenarios, including object detection or image segmentation tasks. This transferability underscores the versatility and efficiency of pre-trained models, allowing practitioners to leverage existing knowledge and representations to accelerate the development of solutions for new tasks or domains. Through the extraction and adaptation of these transferable features, pre-trained models facilitate streamlined model development, enabling practitioners to capitalize on the wealth of information encoded within pre-trained architectures and weights to address a wide array of machine learning challenges effectively [Web-24].

1.5.1.1. RNNS PRE-TRAINED MODELS

Transfer learning in recurrent neural networks (RNNs) involves integrating features extracted from images by pre-trained CNNs like VGG, ResNet, or ViT, particularly beneficial for tasks like time series analysis or natural language processing. For instance, in image captioning, features from VGG, ResNet, or ViT can complement RNNs such as LSTM or GRU, facilitating rich textual descriptions of images. Hybrid architectures combining CNNs and RNNs address tasks with both image and sequential data, where joint fine-tuning of pre-trained CNN components (e.g., VGG, ResNet, ViT) with RNNs on new datasets enhances performance by encapsulating visual and sequential patterns. This convergence improves task-specific representations, highlighting the effectiveness of transfer learning in RNNs.

1.5.1.2. CNNS PRE-TRAINED MODELS

Transfer learning with convolutional neural networks (CNNs) involves leveraging pre-trained models trained on large datasets to improve performance on new tasks or datasets. Initially, a pre-trained CNN model, such as VGG19, VGG16, DenseNet or ResNet, trained on a dataset like ImageNet, is selected. The learned representations from the last few layers of the CNN, before the fully connected layers, are extracted as features. These features serve as inputs for a new task, which may include classification, object detection, or segmentation. To adapt the pre-trained CNN to the new task, additional layers, such as fully connected or convolutional layers, are added on top of the pre-trained model. These layers are then trained on the new dataset using techniques like back-propagation and gradient descent. Fine-tuning, an optional step, involves adjusting the weights of both the pre-trained layers and the newly added layers on the new dataset. This process allows the model to learn task-specific features from the new data while still benefiting from the general features learned during pre-training. Overall, transfer

learning with CNNs offers a powerful approach to improving performance on a wide range of computer vision tasks, reducing training time and computational resources required for training from scratch [Web-25].Below, we outline the most prominent pre-trained models in CNNs:

VGG-16 model architecture: The VGG16 architecture (Figure 1. 16) is a convolutional neural network (CNN) model developed by the Visual Geometry Group (VGG) at the University of Oxford, renowned for its excellence in image classification tasks. It is a prominent member of the VGG family of models, distinguished by the following architectural characteristics: The input layer of VGG16 accepts input images of a consistent resolution, typically set at 224x224 pixels, which conventionally consist of three color channels, corresponding to the Red, Green, and Blue (RGB) color space. The convolutional layers consist of 13 convolutional layers, each consecutively followed by a Rectified Linear Unit (ReLU) activation function. These convolutional layers are responsible for the extraction of hierarchical features at various scales within the input image. Max-pooling layers are incorporated after each pair of consecutive convolutional layers, serving the purpose of down-sampling and reducing the spatial dimensions of the feature maps while retaining significant feature information. The fully connected layers in VGG16 comprise three fully connected layers, where the initial two layers are equipped with 4096 neurons each, while the final layer contains 1000 neurons. The ultimate layer aligns with the number of classes in the ImageNet dataset and undertakes the task of high-level feature abstraction and classification. The concluding softmax layer is responsible for assigning class probabilities to the various categories within the classification problem. Preceding the fully connected layers, a flatten layer is used to convert the feature maps into a one-dimensional vector, making them amenable for input into the fully connected layers.

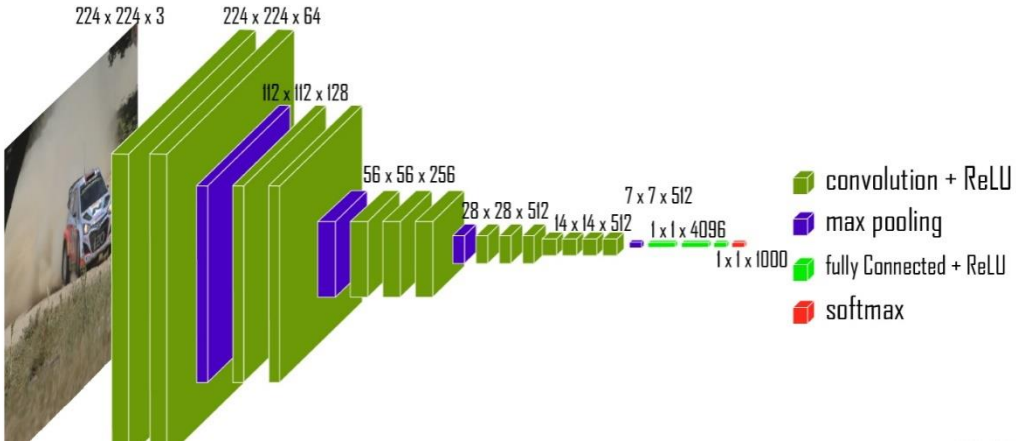

Figure 1. 16 : VGG-16 model architecture [Web-26].

VGG-19 model architecture: The VGG-19 architecture (Figure 1.17), an extension of the original VGG model, was developed by the Visual Geometry Group (VGG) at the University of Oxford and represents a significant advancement in the domain of deep convolutional neural networks (CNNs) for image recognition tasks. The input layer of VGG-19, like its precursor VGG-16, is designed to process input images of a standardized size, typically set at 224x224 pixels, utilizing the conventional three color channels (RGB). This standardization promotes compatibility with diverse image datasets. The convolutional layers of VGG-19 exhibit a profound depth with 16 successive convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function. These convolutional layers are instrumental in capturing intricate visual features, such as edges, textures, and patterns, across various spatial scales. Max-pooling layers are incorporated following each pair of consecutive convolutional layers, serving the dual purpose of spatial dimension reduction and feature preservation, which

facilitates the network's ability to generalize effectively. VGG-19's architectural design comprises three fully connected layers. The initial two fully connected layers encompass 4096 neurons each, while the ultimate layer culminates in 1000 neurons, aligning with the number of categories in the widely recognized ImageNet dataset. These fully connected layers empower the network to extract high-level features and execute classification tasks effectively. The final architectural element, the softmax layer, is responsible for assigning class probabilities to the network's output, transforming the model's raw predictions into a probability distribution across potential image categories. Preceding the connection with the fully connected layers, a flatten layer is employed to transform the feature maps from the preceding convolutional and pooling layers into a one-dimensional vector, facilitating the seamless integration of features for classification [Web-25].

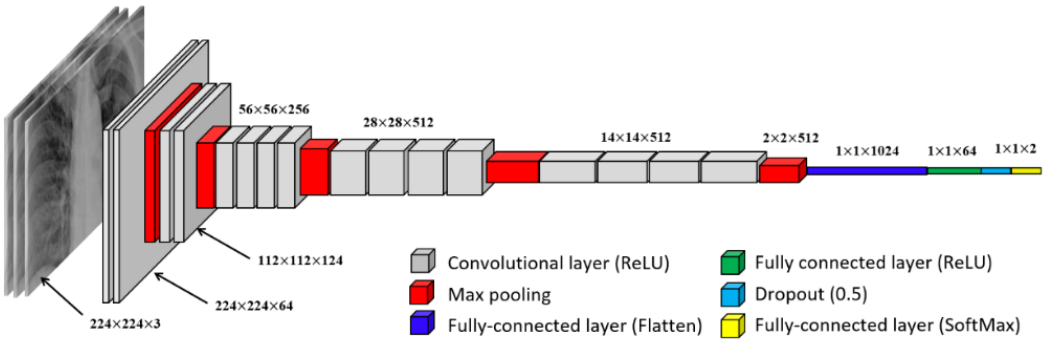

Figure 1. 17 : VGG-19 model architecture (Kamil et al, 2021).

ResNet model architecture: introduced as "Deep Residual Learning for Image Recognition" by researchers at Microsoft Research, represents a pioneering development in deep convolutional neural networks (CNNs). It addresses the vanishing gradient problem encountered in training deep neural networks. ResNet starts with an input layer for standard RGB images (224x224 pixels). Initial layers include a 7x7 convolutional layer with 64 filters, followed by batch normalization and ReLU activation, and a max-pooling layer to down-sample feature maps (Figure 1. 18). The architecture's core consists of residual blocks with two convolutional layers and a skip connection that adds the input directly to the output. In deeper networks, a bottleneck architecture with three convolutional layers (1x1, 3x3, 1x1) is used to balance efficiency and capacity. Multiple residual blocks are stacked to achieve depth, varying by ResNet variants (e.g., ResNet-18, ResNet-50). Traditional fully connected layers are replaced by global average pooling, producing a one-dimensional vector for classification, followed by a fully connected layer with softmax activation for output [Web-25].

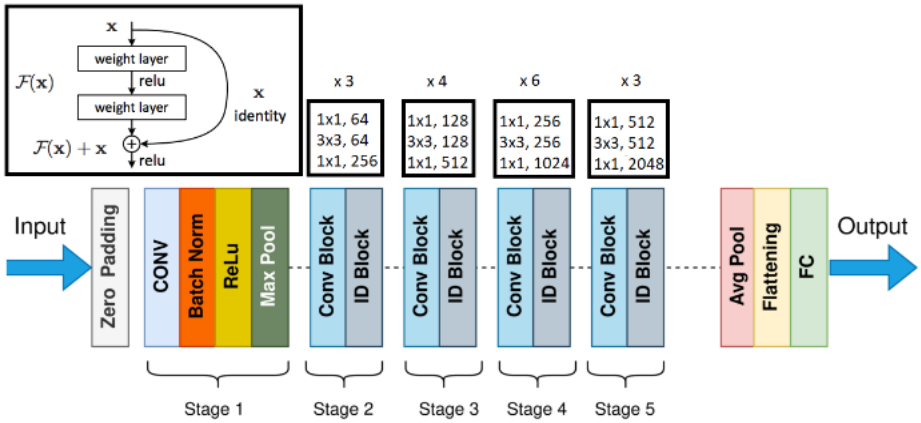

Figure 1. 18 : ResNet-50 model architecture (Rahul gomes et al, 2022).

AlexNet model architecture: AlexNet, devised by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, is a seminal CNN architecture that has transformed computer vision and image classification. It features a meticulously structured design (Figure 1.19) that serves as a precursor to modern deep learning models. AlexNet starts with an input layer for RGB images resized to 227x227 pixels. Its core consists of five convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function for non-linearity. Max-pooling layers are used after the first two convolutional layers for spatial down-sampling, promoting translation invariance and reducing parameter count. Local response normalization (LRN) is applied after the first and second convolutional layers to enhance the network's generalization capacity. Dropout is used within the fully connected layers during training to combat overfitting by randomly deactivating a fraction of neurons. The architecture includes three fully connected layers with 4096 neurons each, leading to a softmax layer for high-level feature extraction and classification. The softmax layer computes class probabilities, transforming model outputs into a probability distribution. Before reaching the fully connected layers, feature maps are flattened into a one-dimensional vector. Data augmentation strategies, such as random cropping and horizontal flipping, are used during training to enhance network resilience and reduce overfitting [Web-25].

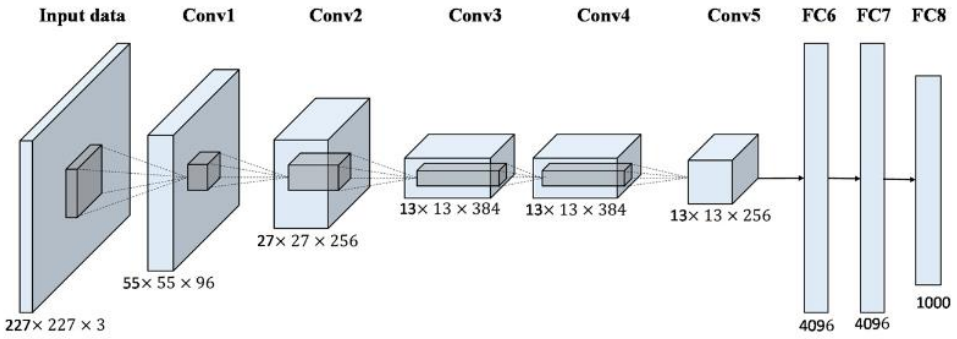

Figure 1. 19 : AlexNet model architecture (Han et al, 2017).

MobilNet model architecture: MobileNet, designed by Andrew G. Howard et al. from Google Research, is a convolutional neural network (CNN) architecture tailored for efficient and lightweight deep learning models. It addresses the challenge of running neural networks on resource-constrained devices like mobile phones and embedded systems (Figure 1. 20). MobileNet achieves efficiency through depthwise separable convolutions, which split the standard convolution operation into two components: depthwise convolution and pointwise convolution. The depthwise convolution applies a 3x3 filter independently to each input channel, reducing parameters and computational load. It is

followed by batch normalization and ReLU activation to enhance spatial information capture within channels. Pointwise convolution then combines outputs across channels using 1x1 filters, effectively mixing features and reducing dimensionality. The network begins with an input layer that receives raw pixel values and proceeds with a standard 3x3 convolutional layer for spatial reduction, followed by a series of depthwise separable convolutional layers. These layers stack depthwise convolutions, batch normalization, ReLU activation, and pointwise convolutions to optimize efficiency while maintaining feature extraction capability. MobileNet comprises multiple depthwise separable convolution blocks that progressively enhance feature complexity and abstraction. It incorporates downsampling layers like strided convolutions or max pooling for spatial dimension reduction and upsampling layers such as bilinear interpolation for spatial resolution recovery. At the end of the network, global average pooling condenses feature maps to a single vector by averaging values, preparing them for processing in fully connected layers. Typically concluding with a fully connected layer and softmax activation for classification, MobileNet outputs normalized class probabilities, enabling accurate predictions across defined classes (Singh et al., 2023).

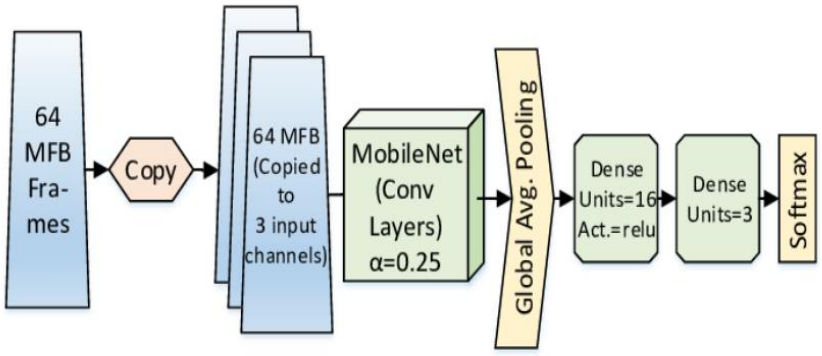

Figure 1.20 : MobileNet model architecture (singh et al, 2023).

DenseNet model architecture: DenseNet, or Densely Connected Convolutional Networks, introduced by Gao Huang, Zhuang Liu, and Kilian Q. Weinberger in their 2016 paper, represents a significant advancement in computer vision. This neural network architecture has been successfully applied to tasks such as image classification, object detection, and segmentation. DenseNet is structured around dense blocks as shown in (Figure 1. 21), each comprising layers where every layer is connected directly to every other layer within the block. This dense connectivity promotes feature reuse and facilitates gradient flow throughout the network, enhancing learning efficiency. Transition layers are employed between dense blocks to reduce the spatial dimensions of feature maps while increasing the number of channels. These layers typically integrate convolutional and pooling operations to manage network size effectively. Within each dense block, bottleneck layers precede 3x3 convolutional layers to decrease channel dimensions, optimizing computational efficiency while maintaining the model's representational power (Sridhar et al, 2023).

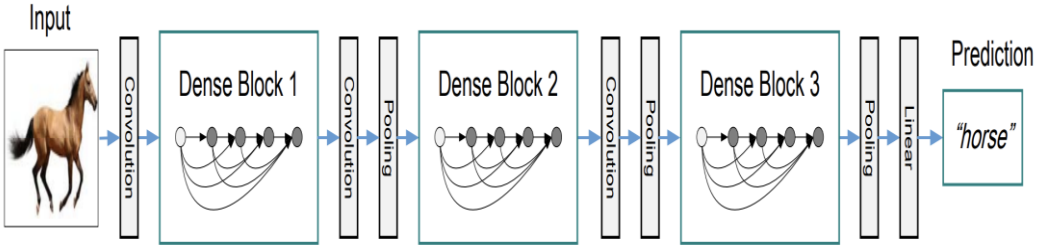

Figure 1. 21 : DenseNet model architecture (Sridhar et al. 2023).

EfficientNet model architecture: EfficientNet is a convolutional neural network architecture and scaling method that uniformly scales all the dimensions of depth, width and resolution using a compound coefficient (Figure 1. 22). EfficientNet, first introduced in Tan and Le, is among the most efficient models (requiring least FLOPS for inference) that reaches State-of-the-Art accuracy on both imagenet and common image classification transfer learning tasks. By introducing a heuristic way to scale the model, EfficientNet provides a family of models (B0 to B7) that represents a good combination of efficiency and accuracy on a variety of scales. Such a scaling allows the efficiency-oriented base model (B0), (our model is EfficientNet B0), to surpass models at every scale, while avoiding extensive grid-search of hyper-parameters (Tan et al, 2020).

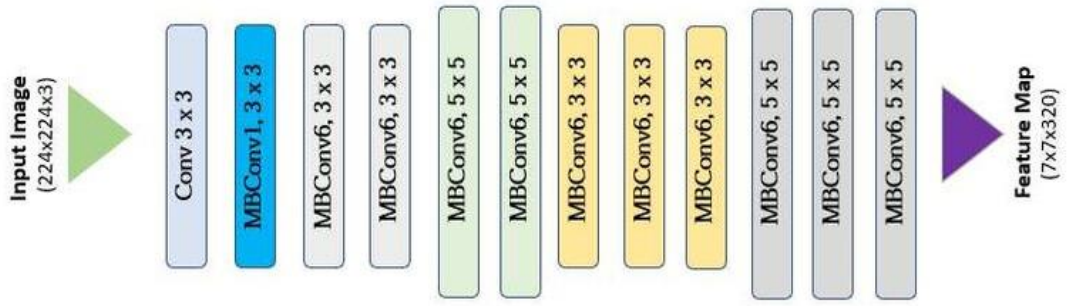

Figure 1.22 : EfficientNet model architecture (Hisaria et al, 2024)

1.5.1.3. TRANSFORMERS PRE-TRAINED MODELS

Transformers have emerged as indispensable tools in advancing the capabilities and performance of various NLP and audio processing, exemplified by models like BERT and Wav2Vec2, respectively.

BERT: The BERT model was proposed in BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by (Jacob Devlin et al). It's a bidirectional transformer pre-trained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising the Toronto Book Corpus and Wikipedia. BERT was trained with the masked language modeling (MLM) and next sentence prediction (NSP) objectives. It is efficient at predicting masked tokens and at natural language understanding (NLU) in general, but is not optimal for text generation (Turc et al, 2019).

Wav2Vec2: Wav2Vec2 is a speech model that accepts a float array corresponding to the raw waveform of the speech signal. The Wav2Vec2 model was trained using connectionist temporal classification (CTC), so the model output has to be decoded using the Wav2Vec2CTCTokenize (Zhou et al, 2020).

Some of ViTs architectures model, each model variant represents different configurations of the Vision Transformer:

ViT-B8: This model uses a base configuration with smaller patch sizes (8x8).

ViT-S16: This is a smaller version of the model with standard patch sizes (16x16).

ViT-L16: This model is a larger variant with standard patch sizes (16x16) (Dosovitsky et al, 2021).

Additionally, we have vision transformers that have significantly impacted the field of computer vision, along with various pre-trained models such as DeiT, BEiT and MAE.

DeiT (Data-efficient Image Transformers): DeiT is a model developed by Facebook AI. These models serve as distilled versions of vision transformers (Figure 1.23). Additionally, the creators of DeiT have

introduced more efficiently trained ViT models, which can be directly incorporated into ViT Model or ViT For Image Classification. There are four variants available across three different sizes: facebook-deit-tiny-patch (16, 224), facebook-deit-small-patch (16, 224), facebook-deit-base-patch (16, 224), and facebook-deit-base-patch (16, 384). It's important to use DeiT Image Processor to prepare images for the model's input (Touvron et al, 2021).

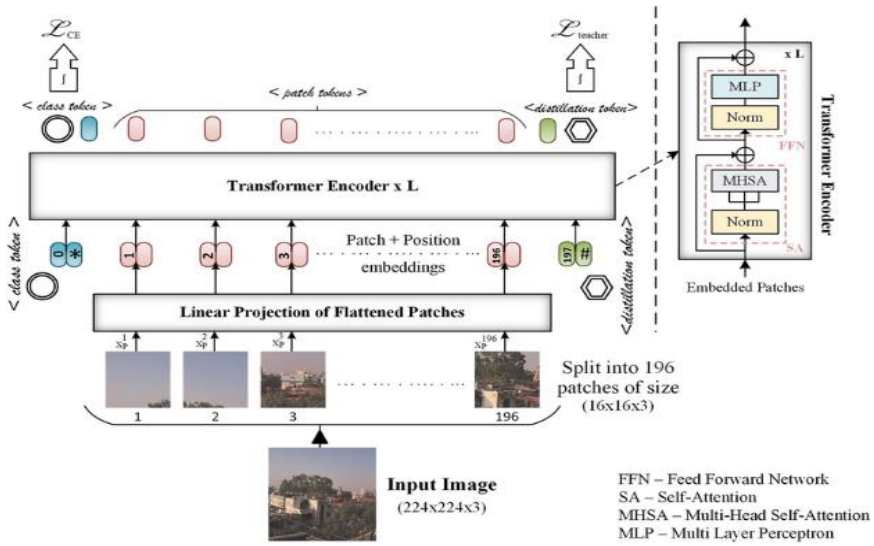

Figure 1.23: DeiT model architecture (Anju Mohan et al, 2024).

BEiT (BERT pre-training of Image Transformers): BEiT is a model developed by Microsoft Research. BEiT models excel beyond supervised pre-trained vision transformers by employing a self-supervised approach inspired by BERT (Figure 1.24), specifically through masked image modeling. This technique is built upon a VQ-VAE architecture (Touvron et al, 2021).

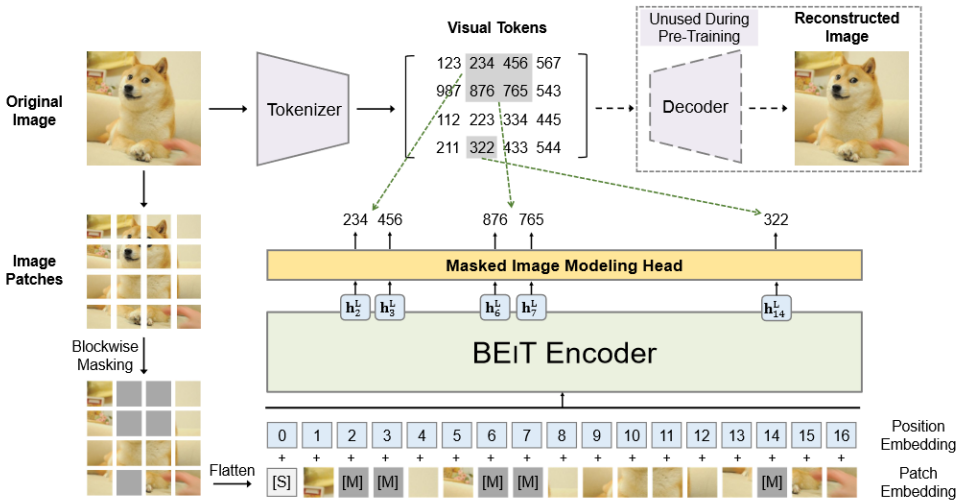

Figure 1.24: BEiT architecture (Hango Bao et al, 2022).

MAE (Masked Auto encoders): MAE is a model developed by Facebook AI. It involves pre-training Vision Transformers to reconstruct pixel values for a significant portion (75%) of masked patches, employing an asymmetric encoder-decoder architecture (Figure 1.25). Through this approach, the authors demonstrate that MAE outperforms supervised pre-training after fine-tuning (He et al, 2021).

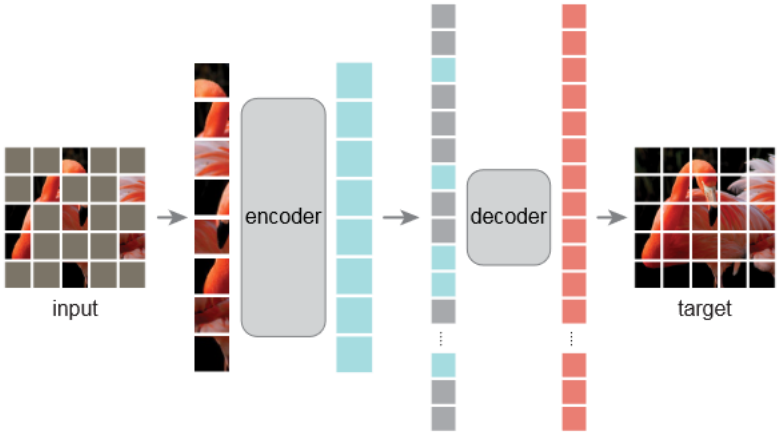

Figure 1.25 : MAE model architecture (Kaiming He et al, 2021).

1.5.2. FINE-TUNING

Fine-tuning, technique consisting of specializing a pre-trained AI model to accomplish a specific task. This generally involves training the model as a whole, or only certain layers of a neural network, for a small number of iterations on a specific data set corresponding to the target task (Figure 1.26). This practice is sometimes translated as “refining”, “fine adjustment”, “tweaking” or even “specialization” [Web-27].

Fine-tuning contains several strategies:

Customization for the target task: Fine-tuning involves adapting the pre-trained model to a specific target task by updating its learned representations. This is achieved by training the model on a new dataset related to the target task.

Freezing and unfreezing layers: Freezing involves initially locking the weights of most layers in a pre-trained model, excluding the last few layers, to maintain learned representations from prior tasks. In contrast, unfreezing occurs during training, where some or all frozen layers are opened up to update their weights. This fine-tunes the model's representations to align more closely with the specifics of the new task.

Learning rate scheduling: Fine-tuning often involves using a lower learning rate compared to the initial pre-training phase. This helps prevent catastrophic forgetting of the pre-trained knowledge while allowing the model to adapt to the new task.

Regularization techniques: Regularization techniques such as dropout or weight decay may be employed during fine-tuning to prevent overfitting, especially when the target task's dataset is small.

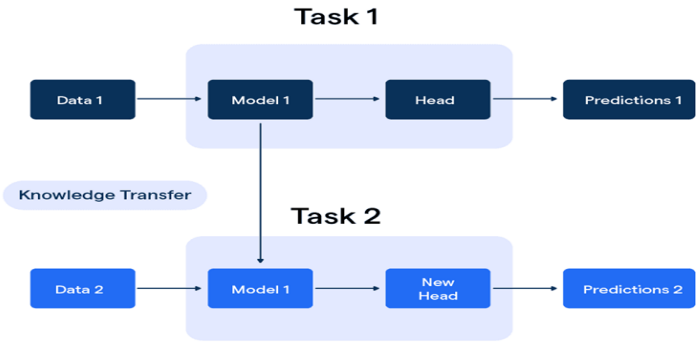

Figure 1.26 : Fine-tuning [Web-28].

1.5.3. BENEFITS OF TRANSFER LEARNING

Transfer learning is a powerful technique in deep learning that leverages pre-trained models to enhance performance on new tasks, reduce training time and data requirements, and promote generalization across domains. Here's a breakdown of its key aspects [Web-29]:

Transfer learning overview: Transfer learning leverages pre-trained models to enhance performance on new tasks, reduce training time and data requirements, and promote generalization across domains.

Efficient knowledge utilization: Transfer learning, especially when coupled with fine-tuning, efficiently uses pre-existing models and learned representations, reducing the need for extensive annotated datasets and computational resources.

Streamlined learning process: By leveraging pre-trained models, transfer learning minimizes the need to start from scratch, streamlining the learning process and accelerating convergence during training.

Adaptability and generalization: Fine-tuning in transfer learning allows models to adapt representations to target tasks, leading to improved generalization performance and enhanced predictive capabilities by discerning subtle patterns specific to the task.

Accelerated convergence: Initialization with pre-trained knowledge accelerates convergence during training, jumpstarting the learning process and expediting parameter optimization.

Versatile performance: Transfer learning, particularly through fine-tuning, achieves state-of-the-art performance in various tasks like image classification, object detection, natural language processing, and speech recognition, highlighting its pivotal role in advancing machine learning research and applications.

1.6. CONCLUSION

In this chapter, we launched on a complete examination of learning approaches, beginning with an explanation of fundamental ideas in machine learning, including various types and tasks. Following that, we explored the area of deep learning, unpacking its techniques and methodologies that have revolutionized numerous domains. We also presented the innovative paradigm of transformers, emphasizing their role and applications in enhancing learning capabilities, particularly in capturing complex relationships within sequential data, Finally, our discussion highlighted the notion of transfer learning, which supports the effectiveness of pre-trained models in multiple tasks.

In the following chapter, we will demonstrate, through a comprehensive literature review, the AI techniques and modalities used in person identification and authentication processes.

CHAPTER 2

PERSON IDENTIFICATION BASED ON FACIAL FEATURES

2.1 INTRODUCTION

Person identification is essential in many areas of technology, including security, financial integrity, accuracy in healthcare, and law enforcement effectiveness. As technology continues to advance, new methods of identifying individuals have appeared, each with unique benefits and uses. To accomplish accurate and reliable identification of people, these modalities make use of a variety of biometric attributes, including face features, voice patterns, iris scans, fingerprints, gait patterns, and behavioral factors.

The creation and evaluation of person identification systems rely significantly on the availability of complete biometric data in high-quality databases. In this chapter, dedicated particularly to person identification from facial features, we examine the most recent developments and approaches influencing this important subject of study as we explore the newest research and improvements in person identification area.

2.2 PERSON IDENTIFICATION MODALITIES

The crucial process of verifying the identification of a person or entity by evaluating their credentials or supporting documentation is known as authentication. Authentication serves as a crucial component in the vast array of identification technology modalities, confirming the identity of the person requesting access using a variety of biometric or behavioral characteristics. These include iris, fingerprint, and face recognition, in addition to a variety of additional techniques. Authentication is essential because it ensures that only authorized parties can access devices, systems, or sensitive data, strengthening security protocols and preventing any illegal access. The key modalities of identifying a person are: Face recognition, Voice recognition, Fingerprint scanning, Iris scanning, Gait recognition, Behavioral biometrics, and DNA matching.

2.2.1 SPEAKER RECOGNITION

The process of recognizing a person by their distinctive voice is known as speaker recognition (SR). Various factors such as vocal tract forms (Figure 2. 1), larynx sizes, and other variations in voice production organs lead people to sound different from one another. Every person also has a unique speaking style, enunciation pattern, word selection, and so forth. In addition to fingerprints and retinal scans, speech can also be used as a biometric because of all these factors.

Speaker recognition authentication uses a biometric technique to recognize or authenticate people by listening to their distinctive voice traits. There are multiple important steps in this process: voice capture is the process of recording a person's spoken words as audio data using microphones or other audio equipment; feature extraction is the process of removing pertinent characteristics for additional analysis, such as pitch, intensity, spectral features, frequency of voice, length of speech segments, and energy distribution across various frequency bands; Speech analysis: algorithms examine extracted features, such as phonemes, syllables, words, emotional states, and accents, to find patterns, linguistic traits, and emotional content; speaker verification, also known as voice matching, verifies the speaker's identity by

comparing the extracted voice features with a voiceprint that has been stored and matching a passphrase entered into the system. Lastly, authentication verifies the speaker's stated identity using the comparative data gathered via speaker verification. After establishing a match between the extracted voice characteristics and the recorded voiceprint, the system authenticates the speaker's identity, providing access to the requested service or resource [Web-30].

Figure 2. 1 : speaker recognition [Web-31].

2.2.2 FINGERPRINT SCANNING

Fingerprint scanning, sometimes referred to as fingerprint recognition or fingerprint authentication, is a biometric technique work by examining a finger pressed against a smooth surface. The finger’s ridges and valleys are scanned, and a series of distinct points, where ridges and valleys end or meet, are called minutiae. These minutiae are the points the fingerprint recognition system uses for comparison. In order to identify an individual.

There are several crucial steps in the fingerprint authentication process. Using technologies like optical, capacitive, or ultrasonic sensors, a fingerprint scanner first takes a picture of the fingerprint as shown in (Figure 2. 2). A fingerprint template is a distinct digital representation of the fingerprint that is created when certain fingerprint properties, like ridge orientations, bifurcations, and ends, are retrieved from the image. In order to identify or authenticate the person, this extracted template is compared with templates that have been saved in a database during template matching. This matching procedure makes use of sophisticated algorithms that take pressure, angle, and distortion fluctuations into account. The system determines the individual's identification or authenticity based on the comparison results. Many benefits come with fingerprint scanning, such as great accuracy, speed, usability, security, and non-intrusiveness, Because of its quick processing speed and accurate recognition abilities, it is a dependable approach for access control and authentication, making it appropriate for applications like device unlocking or protecting sensitive data (Devi et al, 2022).

Fingerprint scanning provides numerous advantages, including high accuracy, speed, ease of use, security, and non-intrusiveness. With its ability to precisely identify individuals, fingerprint scanning offers a reliable method for authentication and access control. Its rapid processing speed ensures efficiency, making it suitable for various applications such as unlocking devices or securing sensitive information.

Figure 2. 2 : Fingerprint scanning [Web-32].

2.2.3 IRIS SCANNING

Iris scanning, sometimes referred to as iris biometrics or iris recognition, is a type of biometric technology that uses a person's distinctive iris patterns to identify them. The colored area of the eye that encircles the pupil is called the iris (Figure 2. 3). Iris scanning is a technique used for identification and authentication that involves taking high-resolution pictures of the iris and examining its complex patterns.

There are various crucial steps involved in the iris scanning procedure. First, a specialized camera takes a close-up picture of the iris, frequently illuminating distinctive patterns with near-infrared light to prevent discomfort. Subsequently, crypts, furrows, and collarette arrangement are extracted from this image by software algorithms, which create a digital template of the iris. These retrieved characteristics are transformed into a mathematical template and used as a secure, one-of-a-kind identity that is kept in a database. The person's iris is scanned once again during identification or authentication, and advanced matching algorithms are used to compare the features of the iris with templates that have been stored to detect similarities. Ultimately, the system decides whether to accept the individual's identity or authentication based on these comparative results [Web-33].

Iris scanning presents numerous advantages, including high accuracy, stability, non-intrusiveness, resistance to forgery, and a wide range of applications. With its ability to accurately capture unique iris patterns, this technology ensures precise identification of individuals, enhancing security measures in various sectors such as border control, banking, and healthcare.

Figure 2. 3 : iris scanning [Web-34].

2.2.4 GAIT RECOGNITION

Gait recognition, sometimes referred to as gait analysis or gait biometrics, is a type of biometric technology that uses a person's distinctive gait patterns to identify them (Figure 2. 4). Each person walks differently due to a variety of characteristics including body composition, weight distribution, and limb

movements. These traits are recorded and examined by gait recognition systems in order to determine an individual's identification.

There are several processes involved in gait recognition, from gathering information to making decisions. In the beginning, metrics like stride length, step duration, and angle of movement are recorded by sensors such as floor sensors or video cameras while a person walks. Following an analysis of this data by software algorithms, distinct elements from the walking pattern are extracted, such as the trajectories of body parts, the time of footfalls, and the spatial relationships between body segments. These characteristics function as a distinct identifier that is kept in a database by being encoded into a mathematical template. The person's gait is recorded once more during identification or authentication, and sophisticated algorithms are used to compare the elements of the recorded gait with templates that have been stored to determine resemblance. Ultimately, the system decides whether to accept the individual's identity or authentication based on these comparative results (Wan et al, 2018).

Gait recognition offers numerous advantages, including behavioral biometrics utilization, affordability, continuous authentication, non-intrusiveness, and resistance to forgery. It operates as a biometric authentication method that analyzes an individual's unique walking pattern, offering a touchless and unobtrusive means of identification.

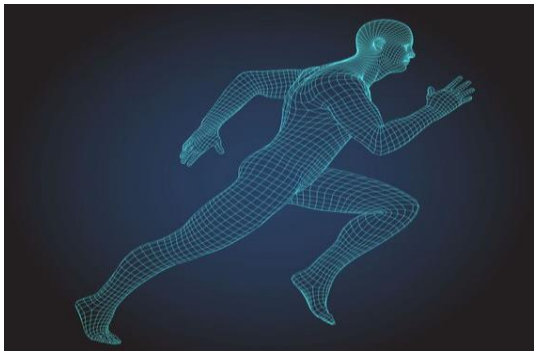

Figure 2. 4 : Gait recognition [Web-35].

2.2.5 DEOXYRIBONUCLEIC ACID MATCHING

By examining each person's distinct Deoxyribonucleic Acid (DNA) sequence (Figure 2. 5), a forensic technique called DNA matching also referred to as DNA profiling or DNA fingerprinting is used to identify them. Deoxyribonucleic acid, or DNA, is the genetic substance found in nearly all living things, including people. Except for identical twins, every person has distinct genetic markers in their DNA that are inherited from their biological parents and distinctive to them.

DNA is extracted from biological materials such as blood, saliva, hair, skin cells, or body fluids in the first of several important steps in the DNA matching process. After that, using specialized procedures, DNA extraction separates the genetic material from the gathered samples. DNA profiling is the process of analyzing the extracted DNA to find genetic markers using techniques such as Short Tandem Repeat (STR) analysis and Polymerase Chain Reaction (PCR). Individually distinct markers combine to form a DNA fingerprint. The DNA profile is then compared to databases that already exist in order to identify possible matches. Reports on the match likelihood are produced by forensic scientists after they have interpreted the data. If a high degree of confidence is needed, further DNA analysis techniques or retesting of the original samples may be used for confirmation testing.

DNA matching is widely utilized across multiple domains, including law enforcement, criminal investigations, paternity testing, missing person identification, and medical research. In criminal justice and law enforcement, it plays a pivotal role in identifying suspects, establishing connections to crime scenes, and resolving previously unsolved cases. Considering its versatility and significance, DNA matching emerges as a crucial technique, offering valuable insights into human identity, relationships, and health [Web-36].

Figure 2. 5 : DNA matching [Web-37].

2.2.6 BEHAVIORAL BIOMETRICS

Behavioral biometrics, the study and analysis of distinct human behavior patterns for identity, authentication, and security purposes, diverges from standard biometric modalities like fingerprint or iris recognition, which rely on physical traits. By scrutinizing digital, physical, and cognitive behaviors, behavioral biometrics distinguishes between cybercriminal actions and genuine client behaviors, thereby detecting fraud and identity theft. For instance, fraudsters often exhibit different digital navigation patterns, such as copying and pasting, compared to authentic users. Behavioral data thus provides valuable insights into fraudulent activities.

A wide array of human activities falls within the scope of behavioral biometrics, including keyboard and mouse movements, voice recognition, signature dynamics, gesture recognition, and touchscreen interaction. Typically, sensors or monitoring devices are employed to collect data on these behavioral patterns. Subsequently, unique features are extracted and encoded into a digital template representing an individual's behavioral biometric signature. These captured patterns are then compared to stored templates for identification or authentication, informing decisions based on the comparison outcomes (levin et al, 2022).

Behavioral biometrics offer numerous advantages. Continuous authentication, low intrusiveness, intuitive interaction, flexibility, and heightened security are just a few of these benefits. Unlike conventional biometric methods, which rely on static physical traits, behavioral biometrics analyze human behavior patterns such as speech modulation, mouse movements, and typing rhythm to validate identity.

2.2.7 FACE RECOGNITION

Facial recognition technology serves as a crucial tool for verifying and identifying individuals, offering real-time or image-based identification capabilities. It captures and analyzes key facial features such as eye spacing, nose shape, and mouth structure to establish identity (Figure 2. 6). This technology holds significant importance within the realm of biometric security due to its high level of security and reliability, making it widely utilized in law enforcement and security applications (Stouffer et al, 2023).

Figure 2. 6 : Face recognition [Web-38].

2.3 PERSON IDENTIFICATION USING FACE RECOGNITION

Face recognition technology has been incorporated into our daily life, whether it is used on mobile phones, laptops, Personal Computers (PC), or traffic surveillance. Facial recognition computer programming first started in 1964, which measured the size of the mouth and eyes. An additional 21 facial markers were added in 1977. In 1988, linear algebra was used to interpret, simplify and manipulate human markers on images. Massachusetts Institute of Technology (MIT) introduced the first successful example of facial recognition technology known as Eigenfaces in 1991.Eight years later the Defense Advanced Research Projects Agency (DARPA) developed a database composed of 2400 images for 850 people. In 2005, a competition known as Face Recognition Grand Challenge (FRGC) designed existing face recognition initiatives. In 2014, deepface was an internal algorithm used by Facebook to recognize faces. In 2018, a smart monitoring system used live facial recognition to identify 50,000 people in a crowd, to assist Chinese policeto arrest a suspect of economic crime. A more recent development in face recognition occurred in China in 2019, where individuals' faces needed to be checked by the operator to buy a new phone. The aforementioned indicates that facial recognition has been widely used for almost six decades (Dash et al, 2023).

2.3.1 FACE RECOGNITION PROCESS

Face recognition is a complex process that involves several stages of image processing and machine learning. Below are common steps within a facial recognition process:

Phase1: Model training

Model training involves the process of training a machine learning model to recognize and authenticate individuals based on their facial features.

Data collection: Collecting a diverse dataset of facial images is vital for effective model training. This dataset should represent individuals who will undergo authentication, covering variations in lighting,

pose, expression, and background. A diverse dataset helps the model generalize better to different scenarios (Kaduwela et al, 2024).

Data preprocessing: Preprocessing involves preparing the collected data for training by standardizing the format, enhancing image quality, and removing noise. Techniques such as resizing, normalization, and data augmentation are applied to improve the quality and consistency of the dataset [Web-39].

Feature extraction: Feature extraction is the process of extracting discriminative features from facial images. Techniques like Principal Component Analysis (PCA) or Convolutional Neural Networks (CNNs) are commonly used to extract meaningful representations of facial features from raw image data.

Model training: After extracting relevant features from facial images, a machine learning model is trained using these feature representations and corresponding labels indicating the identities of individuals. Traditional models such as Support Vector Machines (SVMs) or k-Nearest Neighbors (k-NN) can be trained on the extracted features to learn patterns and relationships within the data. However, deep learning models, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in learning hierarchical and discriminative feature representations directly from raw image data. Popular CNN architectures like VGGNet, ResNet, and FaceNet have been widely adopted for facial recognition tasks, demonstrating superior performance compared to traditional methods. Additionally, recent advancements in transformer-based models, such as Vision Transformers (ViTs), have shown promising results by effectively capturing long-range dependencies and contextual information in images. These deep learning models can be trained end-to-end on the facial image data, eliminating the need for manual feature engineering and potentially learning more robust and discriminative representations for accurate person identification.

Model evaluation: Evaluate model performance on unseen data using metrics like accuracy, precision, and recall.

Model deployment: After training, the trained model is deployed into the identification system where it can process incoming facial images in real-time. The deployed model analyzes the extracted features of the incoming images and makes authentication decisions based on the learned patterns.

Phase 2: Person identification

The identification process determines the identity of an individual based on the extracted features. It involves matching the extracted features of the input facial image against the learned representations of the trained model to identify the individual.

Face detection: The process of face detection is a crucial step in face recognition authentication, as it involves locating and identifying human faces within an image or video frame. These algorithms analyze visual data, identifying patterns, gradients, and textures that correspond to facial features (Kutnyk et al, 2024). Accurate face detection is essential for subsequent stages of the authentication process to ensure reliable identification and verification of individuals. This can be achieved using techniques such as Haar cascade classifier and MTCNN (Multi-Task Cascaded Convolutional Networks).

- **Haar cascade:** The Haar Cascade algorithm is employed to recognize objects within images or video sequences by analyzing sets of rectangular pixel patterns known as Haar-like features (Figure 2. 7). These features enable effective discrimination of objects from backgrounds. Through training on these distinctive features, the classifier can identify objects in new images or videos. Facial detection is a particular strength of this classifier, thanks to its ability to distinguish facial features such as eyes, nose, and mouth. Initially, a dataset containing positive images with faces and negative images without faces is used to train the Haar Cascade classifier.

Once trained, the classifier scans each window of the input images or videos to detect faces, refining results to minimize false positives based on criteria like size, position, and shape [Web40].

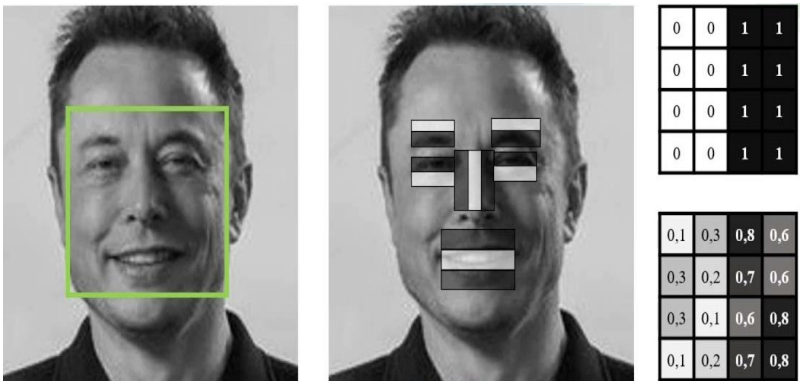

Figure 2. 7 : Haar Cascade classifier [Web-41].

- **Multi-Task Convolutional Networks (MTCNN):** The MTCNN algorithm is a deep learning-based face detection and alignment method that uses a cascading series of convolutional neural networks (CNNs) to detect and localize faces in digital images or videos (Figure 2. 8). The algorithm is capable of detecting faces of different scales and orientations, and is robust to variations in lighting conditions, facial expressions, and occlusions [Web-42].

Figure 2. 8 : MTCNN classifier [Web-43].

Image pre-processing: The system prepares the input images for feature extraction and comparison by performing image preparation activities after faces have been discovered. Among the image preparation techniques are normalization, alignment, cropping, and resizing.

Classification: Following image preparation, the system gathers facial feature information and calculates a distance metric or similarity score between the input face and the faces that are recorded in the database of the model. Based on the similarity score, the algorithm then uses a judgment threshold to categorize the input face as a match or non-match.

Testing and evaluation: The system is tested and evaluated to determine how well it performs when the categorization process is complete. A range of input faces, including those of known and unknown

peoples. Evaluation metrics are calculated to quantify the system's performance objectively, including accuracy, precision, recall, and F1 score (Kaduwela et al, 2024).

2.3.2 DOMAINS OF APPLICATION

The application of person identification techniques encompasses a wide range of sectors, including but not limited to:

Security: Facial recognition technology has become crucial in the security sector, enhancing safety and efficiency across various applications. For access control, it replaces traditional methods like keys or passwords by verifying individuals' identities, ensuring secure and seamless entry into buildings and restricted areas. It is also used in surveillance and monitoring, enabling real-time identification and tracking of individuals in public safety contexts such as airports and large venues. In digital security, facial recognition provides robust authentication for devices like smartphones and laptops, enhancing security against forgery. At border control points, it streamlines immigration procedures by verifying identities and detecting fraudulent documents. Furthermore, integrating facial recognition with other biometric measures, like voice recognition, creates more secure multifactor authentication systems. This technology integrates seamlessly with existing security systems, offering comprehensive solutions. These advancements highlight the critical role of facial recognition in modern security frameworks [Web-44].

Law enforcement: In Law Enforcement, face recognition technology serves multiple critical functions. It aids in Suspect Identification by comparing facial features against databases of known offenders, helping investigators quickly narrow down potential suspects. This technology also plays a role in Missing Persons Cases by analyzing images or videos to match missing individuals' faces with sightings from various sources. In Criminal Investigations, technology modalities analyze audio or video footage to identify suspects or witnesses, while voice recognition matches voices with known individuals or provides clues about suspects. For Forensic Analysis, face recognition helps identify individuals in crime scene images or videos, linking suspects to evidence. Face recognition is also used for Wanted Persons Lists, alerting law enforcement when encountering matching individuals. In Courtroom Evidence, facial recognition authenticates images or videos, and voice recognition verifies audio recordings or statements. Moreover, Interagency Collaboration benefits from these technologies, facilitating data exchange and collaboration between law enforcement agencies for seamless information sharing and investigations (Doyle et al, 2019).

Biometrics: Facial recognition technology represents a pivotal advancement in the domain of biometrics, offering a multifaceted approach to identity verification. By harnessing unique facial features, this technology significantly enhances security protocols for accessing systems and buildings. Integrated seamlessly into multi-factor authentication frameworks, facial recognition adds an additional layer of robustness alongside traditional methods, fortifying overall security infrastructure. End-users reap the benefits of enhanced convenience, effortlessly accessing devices and systems without the need for cumbersome passwords. Moreover, the integration of facial recognition technology reinforces fraud prevention measures during transactions, while simultaneously enabling personalized user experiences tailored to individual authentication profiles. The implementation of continuous authentication mechanisms further ensures ongoing security, providing continuous monitoring and validation of user identities. The accessibility landscape is also positively impacted, as facial recognition technology accommodates users with disabilities, offering an inclusive authentication solution. Stringent data protection measures are paramount in safeguarding biometric data, ensuring the privacy and security of individual’s sensitive information within the biometric authentication framework (Smith et al, 2021).

Financial services: Facial recognition technology is increasingly integral to the financial services sector, offering a robust solution for identity verification and fraud prevention. By analyzing unique facial features, this technology ensures secure access to accounts, authorizes transactions, and enhances customer authentication processes. Its implementation during customer onboarding significantly reduces the risk of identity theft and streamlines remote account setup, eliminating the need for branch visits. Moreover, facial recognition enables personalized services and targeted marketing efforts based on authenticated user profiles, fostering deeper customer engagement. In customer support interactions, it ensures secure authentication and compliance with regulatory standards, while also facilitating convenient and secure remote banking experiences. Facial recognition technology elevates security, efficiency, and customer satisfaction within the financial services landscape (Foro et al, 2023).

Education: Facial recognition technology has become a cornerstone in modern educational settings, offering diverse functionalities that streamline processes and enhance security. By automating attendance tracking, this technology alleviates the burden of manual input for educators, saving valuable instructional time while ensuring accurate records. Its application extends to controlling access to restricted areas within campuses, bolstering overall security through biometric authentication measures. Additionally, in the context of remote exam proctoring, facial recognition serves as a robust tool for maintaining exam integrity, deterring academic dishonesty, and preserving the credibility of assessment processes. Moreover, personalized learning experiences are facilitated as facial recognition algorithms adapt educational content to match individual student preferences and performance levels, fostering more effective and tailored learning journeys. Administrative tasks such as registration and library checkouts are streamlined, enhancing operational efficiency across educational institutions. Furthermore, through interactive platforms and virtual classrooms, facial recognition technology promotes student engagement by providing immersive learning experiences and real-time feedback for educators to gauge comprehension levels (Andrejevic et al, 2019). And by implementing parental authentication using facial recognition, schools ensure authorized access during interactions, safeguarding student privacy and maintaining a secure learning environment.

Marketing and customer service: In Marketing and Customer Service, the integration of facial recognition technology emerges as a transformative force, reshaping customer interactions and experiences in profound ways. At its core, facial recognition technology facilitates customer identification, enabling companies to personalize marketing messages and recommendations with unprecedented precision based on individual characteristics. This personalized approach extends to targeted advertising campaigns, where facial recognition plays a pivotal role in tailoring advertisements to specific audience segments, thereby enhancing conversion rates and return on investment. Moreover, interactive marketing experiences enriched by facial recognition, such as personalized content delivery and voice-controlled interactions, foster deeper engagement with customers, consequently elevating brand perception and affinity. In the domain of customer service, facial recognition technology holds immense potential in analyzing customer sentiment and preferences through visual cues, thereby driving iterative improvements in products and services. The integration of facial recognition into customer service workflows facilitates personalized experiences, efficient queue management, and heightened security measures, all of which contribute to enhanced customer satisfaction and long-term loyalty. The wealth of data insights generated by facial recognition technology empowers companies to continuously optimize marketing campaigns and customer service processes, ensuring better outcomes and sustained competitive advantage in today's dynamic business landscape [Web-45].

Entertainment and gaming: In Entertainment and Gaming, facial recognition technology emerges as a transformative force, reshaping user experiences and immersion in unprecedented ways. Primarily, its integration revolutionizes user authentication processes across gaming platforms and online services, ushering in an era of enhanced convenience and security by eliminating traditional passwords. Facial

recognition technology facilitates avatar customization, enabling players to immerse themselves in virtual worlds that authentically reflect their own features, thereby enhancing personalization and immersion. Emotion recognition represents another compelling application, wherein facial expressions are analyzed to dynamically influence gameplay, imbuing interactions with a deeper level of engagement and responsiveness. And, the integration of voice commands offers hands-free control, further enhancing immersion and intuitiveness within gaming environments. The convergence of face and voice recognition technologies extends to interactive storytelling, where narratives are personalized based on player choices and expressions, fostering dynamic and deeply engaging experiences. Multiplayer interactions are similarly enriched through real-time communication and social features, promoting collaboration and immersion among players. Additionally, facial recognition technology serves as a potent tool in combating cheating behaviors, ensuring fair play and integrity within online gaming environments. Beyond gaming, facial recognition technology captivates audiences through immersive entertainment applications such as augmented reality, where personalized interactions and games are crafted to engage users in novel and captivating experiences. The integration of facial recognition technology in Entertainment and Gaming represents a paradigm shift, offering unparalleled levels of immersion, personalization, and engagement for users across diverse platforms and experiences (Kondrashov et al, 2023).

2.3.3 CHALLENGES AND POSSIBLE SOLUTIONS

While many businesses successfully leverage facial recognition technology (FRT), they likely encountered various hurdles during its development and implementation. The top challenges faced by facial recognition developers and implementers fall into three main categories: Accuracy, security, ethics.

Achieving strong accuracy is one of the major challenges in developing facial recognition technology (FRT). Variations in lighting conditions, facial misalignment, differing poses, and occlusions such as masks present significant obstacles. These challenges have been exacerbated in situations like global pandemics, where masks are worn constantly, making FRT difficult to implement reliably. To overcome these challenges, researchers have found that deploying deep learning techniques with large and diverse datasets significantly improves accuracy. Deep learning models enhance facial recognition by extracting unique facial components from images and using these features to identify individuals from extensive databases. Additionally, 3D imaging has proven effective in achieving more accurate results, although it necessitates the use of expensive sensors and cameras. Expanding datasets to include a wide variety of images with different lighting, poses, and occlusions can also enhance the robustness of FRT. While it is currently impossible to achieve 100% accuracy in facial recognition, these methods collectively contribute to substantial improvements in performance.

FRT has been in existence for some time, but only recently has it come under serious scrutiny due to emerging ethical concerns. These concerns stem from inefficiencies and biases inherent in facial recognition models, which have been particularly problematic in the context of law enforcement. Instances of wrongful imprisonment attributed to inaccurate FRT have heightened controversy and led to significant backlash. Consequently, companies such as International Business Machines (IBM), Amazon, and Microsoft have either ceased or restricted the use of their FRT by law enforcement. IBM discontinued its FRT services for mass surveillance and racial profiling following the George Floyd case, while Amazon imposed a one-year moratorium on police use of its technology, and Microsoft prohibited police use of its FRT until appropriate regulations are established. To address the ethical challenges associated with FRT, a recent Stanford report recommends several measures: maintaining up-to-date and transparent datasets, enabling users to test third-party systems with their own data, ensuring thorough documentation of any software changes that may affect performance, and adhering

to domain-specific regulations. And, enhancing the training processes of AI-powered FRT can mitigate ethical issues and improve system reliability.

Security is a critical challenge in the deployment of FRT, as the use of biometric data such as facial images poses significant risks for identity theft and other malicious activities. An example of these vulnerabilities was demonstrated at the annual Black Hat hacker convention in Las Vegas, where researchers successfully hacked Apple's Face ID within 120 seconds. To mitigate these security threats, several measures can be employed. Ensuring the robustness of the machine learning and deep learning algorithms used in FRT systems is paramount. Conducting white-box or black-box AI security assessments can further enhance the system's defenses. Leveraging cloud storage for biometric data offers enhanced security due to encryption and redundant storage solutions that protect against hardware failures. Cloud service providers also continuously collaborate with cyber security experts to fortify their security protocols. While these methods are not foolproof, they significantly bolster the security of FRT systems and protect sensitive biometric data from exploitation (Javaid et al, 2024).

2.4 STATE OF THE ART FOR PERSON IDENTIFICATION

In the field of identifying individuals, state-of-the-art technology in face recognition has emerged as a pioneering solution. This is the aim of this section.

2.4.1 DATASETS

A data set, sometimes spelled dataset, is a collection of related data that's usually organized in a standardized format. Data sets are used for analytics, business intelligence, artificial intelligence (AI) model training and a variety of other use cases (for example facial recognition). Data sets can vary significantly in both size and type of data. Data sets are available in a variety of formats, such as JavaScript Object Notation (JSON) and Extensible Markup Language (XML). Such formats provide a standardized structure for sharing data across multiple platforms and applications. The data itself is usually written in plain text, so it can be easily filtered, updated and in other ways transformed to meet specific requirements (Sheldon et al, 2023).

So, we introducing some datasets for facial recognition of famous faces that's facilitates the development of facial recognition techniques. These datasets include wide-ranging data on well-known faces, aiding in training facial recognition models more effectively and enhancing their performance in identifying.

2.4.1.1 MICROSOFT CELEB

Microsoft Celeb (MS-Celeb-1M, or MS1M) [Web-46], is a dataset of 10 million face images harvested from the Internet for the purpose of developing face recognition technologies. According to Microsoft Research, who created and published the dataset in 2016, MS Celeb is the largest publicly available face recognition dataset in the world, containing over 10 million images of nearly 100,000 individuals. Microsoft's goal in building this dataset was to distribute an initial training dataset of 100,000individuals' biometric data to accelerate research into recognizing a larger target list of one million people "using all the possibly collected face images of this individual on the web as training data".

These images cover a wide variety of poses, expressions, ages, and lighting conditions, making it suitable for training and evaluating deep learning models for face recognition tasks. The celebrities included in the dataset come from various domains, including entertainment, sports, politics, and more. MS-Celeb-1M has been widely used by researchers and practitioners in the field of computer vision and face recognition for benchmarking and developing state-of-the-art face recognition algorithms [Web-47].

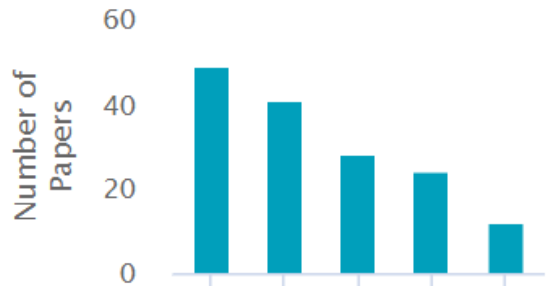

Figure 2. 9 : MS-Celeb-1M paper publications [Web-48].

2.4.1.2 MEGAFACE

MegaFace [Web-49], is a large-scale public face recognition training dataset that serves as one of the most important benchmarks for commercial face recognition vendors. It includes 4,753,320 faces of 672,057 identities from 3,311,471 photos downloaded from 48,383 Flickr users' photo albums. All photos included a Creative Commons licenses, but most were not licensed for commercial use. MegaFace face recognition dataset exploited good intentions of Flickr users and the Creative Commons license system to advance facial recognition technologies around the world by companies including Ali baba, Amazon, Google, Cyber Link, Sense Time, and Vision Semantics to name only a few. According to the press release from the University of Washington, "more than 300 research groups were working with MegaFace" as of 2016, including multiple law enforcement agencies [Web-50].

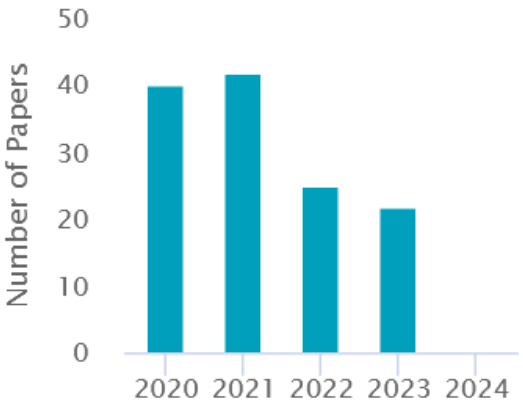

Figure 2. 10 : MegaFacepaper publications [Web-51].

2.4.1.3 CELEBFACES ATTRIBUTES

CelebFaces Attributes Dataset (CelebA) [Web-52], is a vast collection of over 200,000 celebrity images, each meticulously annotated with 40 attributes, encompassing aspects like facial expressions and features. With diverse poses, backgrounds, and a wide range of identities represented, including landmark locations and binary attribute annotations, CelebA serves as a comprehensive resource for various computer vision tasks such as face recognition, attribute recognition, face detection, facial part localization, and even face editing and synthesis, making it a valuable asset for research and development in this field [Web-53].

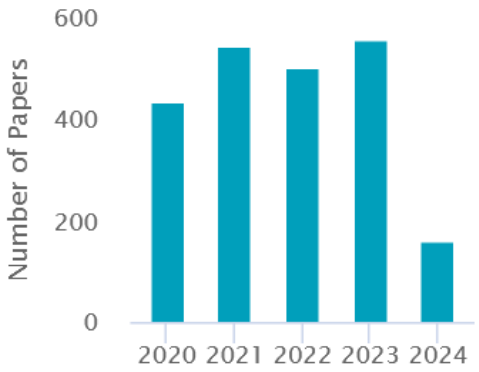

Figure 2. 11 : CelebFaces Attributes [Web-54].

2.4.1.4 PINS-FACE-RECOGNITION

Pins-Face-Recognition data [Web-55], were collected for facial recognition analysis from Pinterest and were carefully cropped for specific purposes. They include 105 celebrity images, encompassing a total of 17,534 individual faces, each representing a crucial piece in the creation and optimization of facial recognition models [Web-55].

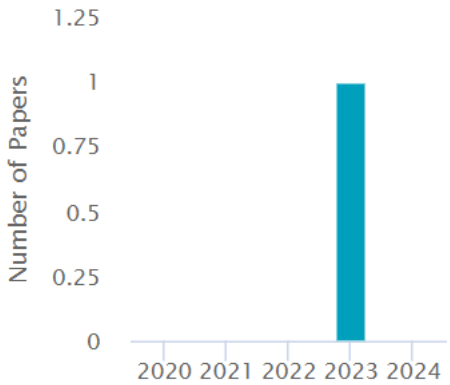

Figure 2. 12 : PINS-Face-Recognition paper publications [Web-56].

2.4.1.5 DIGIFACE-1M

DigiFace-1M [Web-57], is a synthetic dataset for face recognition, obtained by rendering digital faces using a computer graphics pipeline. It contains 1.22M images of 110K unique identities. The dataset consists of two parts. The first part contains 720K images with 10K identities. For each identity, 4 different sets of accessories are sampled and 18 images are rendered for each set. The second part contains 500K images with 100K identities. For each identity, only one set of accessories is sampled and only 5 images are rendered. Following the format of the existing datasets, we provide the aligned crop around the face, resized into 112×112 resolution (Bae et al, 2022).

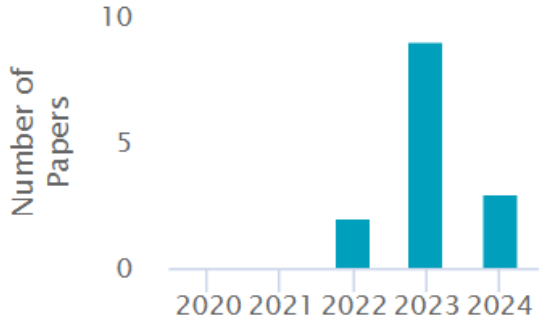

Figure 2. 13 : DigiFace-1M paper publications [Web-58].

2.4.1.6 VGGFACE2

The VGGFace2 dataset [Web-59], is made of around 3.31 million images divided into 9131 classes, each representing a different person identity. The dataset is divided into two splits, one for the training and one for test. The latter contains around 170000 images divided into 500 identities while all the other images belong to the remaining 8631 classes available for training. While constructing the datasets, the authors focused their efforts on reaching a very low label noise and a high pose and age diversity thus, making the VGGFace2 dataset a suitable choice to train state-of-the-art deep learning models on face-related tasks. The images of the training set have an average resolution of 137x180 pixels, with less than 1% at a resolution below 32 pixels (considering the shortest side) (Cao et al, 2017).

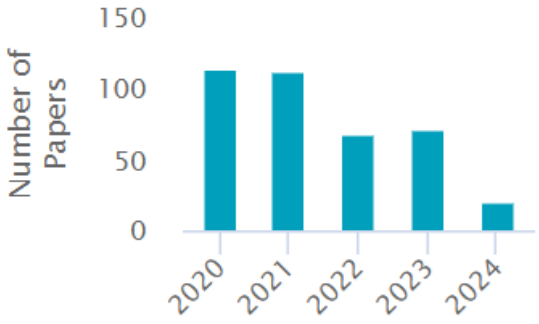

Figure 2. 14 : VGGFace2 paper publications [Web-60].

2.4.1.7 UMDFACES

UMDFaces [Web-61], is a face dataset divided into two parts, Still Images made of around 367,888 face annotations for 8,277 subjects divided into 3 batches. The annotations contain human curated bounding boxes for faces and estimated pose (yaw, pitch, and roll), locations of twenty-one key points, and gender information generated by a pre-trained neural network. And video Frames made of Over 3.7 million annotated video frames from over 22,000 videos of 3100 subjects The annotations contain the estimated pose (yaw, pitch, and roll), locations of twenty-one key-points, and gender information generated by a pre-trained neural network (Bansal et al, 2016).

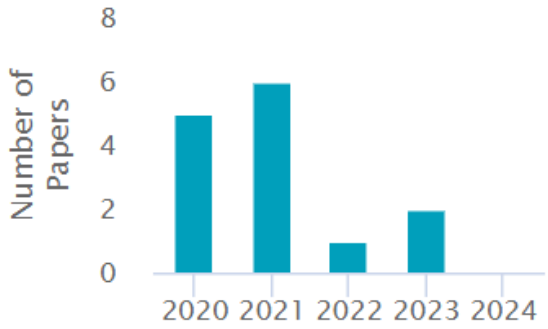

Figure 2. 15 : UMDFaces paper publications [Web-62].

2.4.1.8 IARPA JANUS BENCHMARK A

The IARPA Janus Benchmark A**(**IJB-A**)**database [Web-63]**,** database is developed with the aim to augment more challenges to the face recognition task by collecting facial images with a wide variations in pose, illumination, expression, resolution and occlusion. IJB-A is constructed by collecting 5,712 images and 2,085 videos from 500 identities, with an average of 11.4 images and 4.2 videos per identity (Klare et al, 2015).

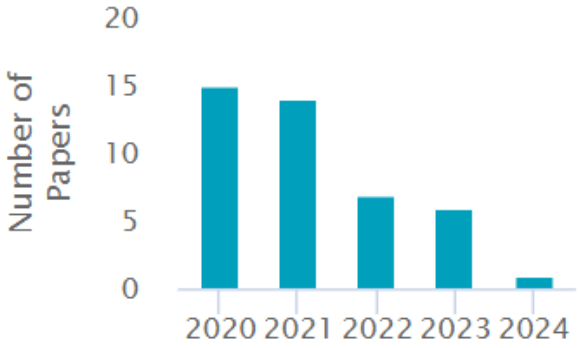

Figure 2. 16 : IJB-A paper publications [Web-64].

2.4.1.9 WEB FACE 260M

Web Face 260M [Web-65]**,** is a million-scale face benchmark, which is constructed for the research community towards closing the data gap behind the industry, It consists of Noisy 4M identities and 260M faces and High-quality training data with 42M images of 2M identities by using automatic cleaning a test set with rich attributes and a time-constrained evaluation protocol (Zhu et al, 2022).

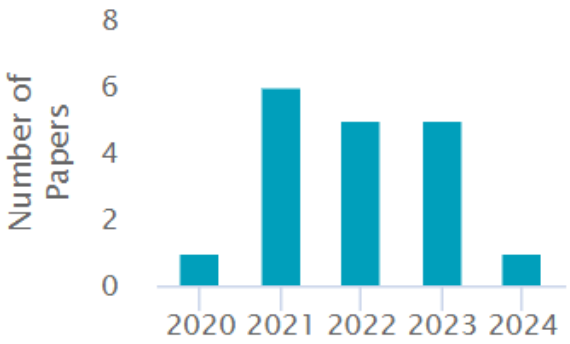

Figure 2. 17 : Web Face 260M paper publications [Web-66].

2.4.1.10 LABELED FACES IN THE WILD

Labeled Faces in the Wild (LFW) [Web-67], is an image dataset containing face photographs, collected especially for studying the problem of unconstrained face recognition. It includes over 13,000 images of faces collected from across the web. Each face in this data set was labeled with the person's name in the image.1680 of the photographed persons distinctly appear in two or more photos in the data set. The faces in these images were detected by the Viola-Jones face detector (Paul Viola and Michael Jones, 2001).LFW includes four different sets of images, including the original and three types of aligned images that can be used to test algorithms under different conditions. For alignment, the dataset uses funneled images (ICCV 2007), LFW-a, and deep funneled images (NIPS 2012). Deep funneled and LFW-a images produce superior results for most face verification algorithms over the funneled images and the original images (Gray et al, 2008).

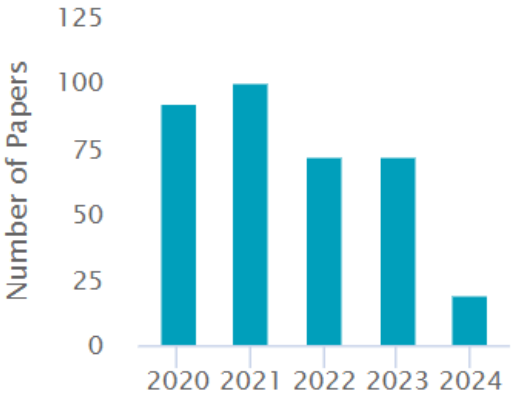

Figure 2. 18 : LFW paper publications [Web-68].

2.4.2 RECENT WORKS

2.4.2.1 THE WORK OF CHAMBINO ET AL. 2021

In this work, a novel architecture for facial recognition that uses multiple deep convolutional neural networks and multispectral images is proposed. A domain-specific transfer-learning methodology applied to a deep neural network pre-trained in RGB images is shown to generalize well to the multispectral domain. They also proposed a skin detector module for forgery detection. Several experiments were planned to assess the performance of our methods. First, they evaluated the performance of the forgery detection module using face masks and coverings of different materials. A second study was carried out with the objective of tuning the parameters of our domain-specific transfer-learning methodology, in particular which layers of the pre-trained network should be retrained to obtain good adaptation to multispectral images. A third study was conducted to evaluate the performance of support vector machines (SVM) and k-nearest neighbor classifiers using the embeddings obtained from the trained neural network. Finally, they compared the proposed method with other state-of-the-art approaches. The experimental results show performance improvements in the Tufts and CASIA NIR-VIS 2.0multispectral databases, with a rank-1 score of 99.7% and 99.8%, respectively (Chambino et al, 2021).

2.4.2.2 THE WORK OF YIMINGGE ET AL. 2022

YimingGe et al, they propose a Convolutional Visual Self-Attention Network (CVSAN), which uses self-attention to augment the convolution operator. Specifically, this is achieved by connecting a convolutional feature map, which enforces local features, to a self-attention feature map that is capable of modeling long-range dependencies. Since there is currently no publicly available large-scale masked face data, they generate a Masked VGGFace2 dataset based on the face detection algorithm to train the CVSAN model. Experiments show that the CVSAN algorithm significantly improves the performance of MFR compared to other algorithms. Accuracy is 98% (YimingGe et al, 2022).

2.4.2.3 THE WORK OF LIU ET AL. 2023

Xinhua Liu et al, relied on in their research paper that the overstaffing production in underground coal mining is not convenient for daily management, and incomplete information of coal miners hinders the rescue process of firefighters during mine accidents. And to address this safety sustainability issue, they proposed a novel face recognition method based on an improved multiscale neural network in this paper. A new depth wise separable (DS)-inception block is designed and a joint supervised loss function based on center loss theory is developed to construct a new multiscale model. The miniers can be recognized in the harsh underground environment during the life rescue. Experimental results show that the accuracy, recall and F1-score indexes of the proposed method for the miner face recognition in the underground mining environment are 97.26%, 94.17% and 95.42%, respectively. Transfer model with joint supervised loss can effectively improve the recognition accuracy by about 0.5–1.5%. In addition, the average recognition accuracy of the proposed face recognition method achieves to 91.34% and the miss detection rate is less than 5% in the dugout tunnel of coal mine (Liu et al, 2023).

2.4.2.4 THE WORK OF KUMAR ET AL. 2023

Kumar et al, they have implemented two different approaches for facial detection. The first is a CNN-based approach that extracts key points from an image and classifies it using a KNN algorithm. The next approach uses a Siamese network to classify the input image. The initial part focuses primarily on data collection and training. The following part clearly explains the implementation of both approaches. The

performance of these approaches was also evaluated and illustrated optimally. They achieved 99% prediction accuracy in KNN (Kumar et al, 2023).

2.4.2.5 THE WORK OF HANGARAGI ET AL. 2023

To control people's entry into restricted areas or grant access to ATMs or computers. In this paper, the researchers proposed model that can detect and recognize the face using Face mesh. Due to Face mesh, the model operates in a variety of conditions such as varying illumination and background. The model can also handle non-frontal images of males and females of all ages and races. The Labeled wild face (LWF) dataset images and images captured in real-time are used to train the deep neural network of the model. During testing, if the face landmarks of the test image match with the face landmarks of any of the training images the model gives the name of the person else model outputs as "unknown". 94.23% accuracy are achieved for face recognition by the proposed model (Hangaragi et al, 2023).

2.4.2.6 THE WORK OF IKROMOVICH ET AL. 2023

In their study, they evaluated facial recognition techniques using the Labeled Faces in the Wild dataset. Comparing transfer learning with fine-tuning pre-trained models to training from scratch, they found that the former significantly improves accuracy, with an achieved accuracy of 98%. Even with limited labeled data, transfer learning proves effective, requiring less data compared to training from scratch. Their results underline the efficiency of transfer learning in deep CNNs for facial recognition, suggesting its potential for developing more accurate and resource-efficient systems for real-world applications (Ikromovic et al, 2023).

2.4.2.7 THE WORK OF RAHMAN ET AL. 2023

Rahman et al, present a system capable of automatic recognition of face mask position could alert and ensure that an individual is wearing a mask properly before entering a crowded public area and putting themselves and others at risk. They first develop and publicly release a dataset of face mask images, which are collected from 391 individuals of different age groups and gender. Then, they study six different architectures of pre-trained deep learning models, and finally propose a model developed by fine tuning the pre-trained state of the art MobileNet model. And evaluate the performance (accuracy, F1-score, and Cohen's Kappa) of this model on the proposed dataset and MaskedFace-Net, a publicly available synthetic dataset created by image editing. Its performance is also compared to other existing methods. The proposed MobileNet is found as the best model providing an accuracy, F1-score, and Cohen's Kappa of 99.23%, 99.22%, and 99.19%, respectively for face mask position recognition. It outperforms the accuracy of the best existing model by about 2%. An automatic face mask position recognition system has been developed, which can recognize if an individual is wearing a mask correctly or incorrectly. The proposed model performs very well with no drop in recognition accuracy from real images captured by a camera (Rahman et al, 2023).

2.4.2.8 THE WORK OF CARDAIOLI ET AL. 2023

Matteo Cardaioli et al, they proposed BLUFADER, a novel continuous authentication system that takes advantage of blurred face detection and recognition to fast, secure, and transparent de-authenticate users, preserving their privacy. They obfuscated a webcam with a physical blur layer and use deep learning algorithms to perform face detection and recognition continuously. To evaluate BLUFADER's practicality, they collected two datasets formed by 30 recruited subjects (users) and thousands of physically blurred celebrity photos. The de-authentication system was trained and evaluated using the former, while the latter was used to appraise the privacy and increase variance at training time. To guarantee the privacy-preserving effectiveness of the selected physical blurring filter, they show that

state-of-the-art deblurring models are not able to revert our physical blur. Further, they demonstrate that our approach outperforms state-of-the-art methods in detecting blurred faces, achieving up to 95% accuracy. Moreover, BLUFADER effectively de-authenticates users up to 100% accuracy in under 3 seconds, while satisfying security, privacy, and usability requirements. Last, their continuous authentication face recognition module based on Siamese Neural Network preventively protect users from adversarial attacks, enhancing the overall system security (Cardaioli et al, 2023).

2.4.2.9 THE WORK OF RODRIGUEZ ET AL. 2024

The researchers worked on propose a novel method for images-to video face recognition in realistic scenarios, they proposed an enhanced images-to-video recognition approach, pairing facial images with attributes like pose and quality. Utilizing datasets such as ENFSI 2015, SCFace, XQLFW, Choke Point, and ForenFace, they assessed evidence strength using calibration methods for likelihood ratio estimation. Three models ArcFace, FaceNet, and QMagFaceundergo validation, with the Cost Log-Likelihood Ratio (CLLR) as a key metric. Results indicate that prioritizing high-quality frames and aligning attributes with reference images optimizes recognition, yielding similar CLLR values to the top 25% best frames approach. A combined embedding weighted by frame quality emerges as the second-best method. Upon preprocessing facial images with the super resolution Code Former, it unexpectedly increased CLLR, undermining evidence reliability, advising against its use in such forensic applications (Zeno Geradts, 2024).

2.4.2.10 THE WORK OF TOMAŠEVIĆ ET AL. 2024

Darian Tomašević et al, present a novel identity-conditioned generative framework capable of producing large-scale recognition datasets of visible and near-infrared privacy-preserving face images. The framework (ArcBiFaceGAN) relies on a novel identity-conditioned dual-branch style-based generative adversarial network to enable the synthesis of aligned high-quality samples of identities determined by features of a pretrained recognition model. In addition, the framework incorporates a novel filter to prevent samples of privacy-breaching identities from reaching the generated datasets and improve both identity separability and intra-identity diversity. Extensive experiments on six publicly available datasets (Tufts Face Database) reveal that our framework achieves competitive synthesis capabilities while preserving the privacy of real-world subjects. The synthesized datasets also facilitate training more powerful recognition models than datasets generated by competing methods or even small-scale real-world datasets. Employing both visible and near-infrared data for training also results in higher recognition accuracy on real-world visible spectrum benchmarks. Therefore, training with multispectral data could potentially improve existing recognition systems that utilize only the visible spectrum, without the need for additional sensors, precision is 0.97 (Tomasevic a et al, 2024).

2.5 CONCLUSION

Through this chapter, we have highlighted the remarkable advancements in person identification made possible by cutting-edge technologies such as voice, face, and DNA recognition. These techniques have significantly increased security in a variety of industries and have benefited from large datasets like MegaFace and VGGFace2. More recently, face recognition has concentrated on surmounting obstacles to produce more reliable and accurate systems. These advancements in facial recognition systems have been facilitated by the introduction of recent AI technologies, most notably deep learning techniques and the transfer learning approach.

This is further demonstrated in the following chapter that will describe how to use transfer learning to create a person identification system based on facial recognition, underscoring the ongoing progress in utilizing machine learning techniques to create reliable and effective person identification systems.

CHAPTER 3

FACIAL RECOGNITION BASED ON TRANSFER LEARNING APPROACH

3.1. INTRODUCTION

In this final chapter, we aim to explore our experimental study to define the optimal pre-trained model for facial recognition systems. In the first part, we provide an overview and the architecture of our transfer learning approach, using different pre-trained model architectures, including CNNs such as EfficientNet, DenseNet, ResNet-50, VGG16, and MobileNet, as well as ViTs such as ViT-B8, ViT-S16, and ViT-L16. Next, we discuss the experiments we conducted to optimize the pre-trained models and their evaluations. Finally, we compare the best model with recent works on LFW datasets.

3.2. SYSTEM ARCHITECTURE

In this section, we present a comprehensive overview of our experimental study utilizing a transfer learning approach for facial recognition. The approach is illustrated through a diagram that outlines four major stages, as depicted in (Figure 3. 1). We initiate with the initial stage, which entails detailing the composition and characteristics of the dataset curated for our study. The second stage involves preprocessing the dataset, where we standardize the data to ensure its suitability for subsequent analysis. The third stage focuses on classification, where we explore the efficacy of various pre-trained CNN architectures such as EfficientNet, DenseNet, ResNet-50, VGG16, MobileNet , as well as Vision Transformer (ViT) pre-trained models including ViT-B8, ViT-S16, and ViT-L16, see these architectures in (Chapter 1). We fine-tune each model with our prepared dataset to determine the best-performing model for our facial recognition system. Furthermore, before the final stage of threshold determination, we incorporate a face detection or detection model using the Haar Cascade classifier to detect faces within an image. This ensures that our system accurately identifies and processes facial features before proceeding to the threshold process.

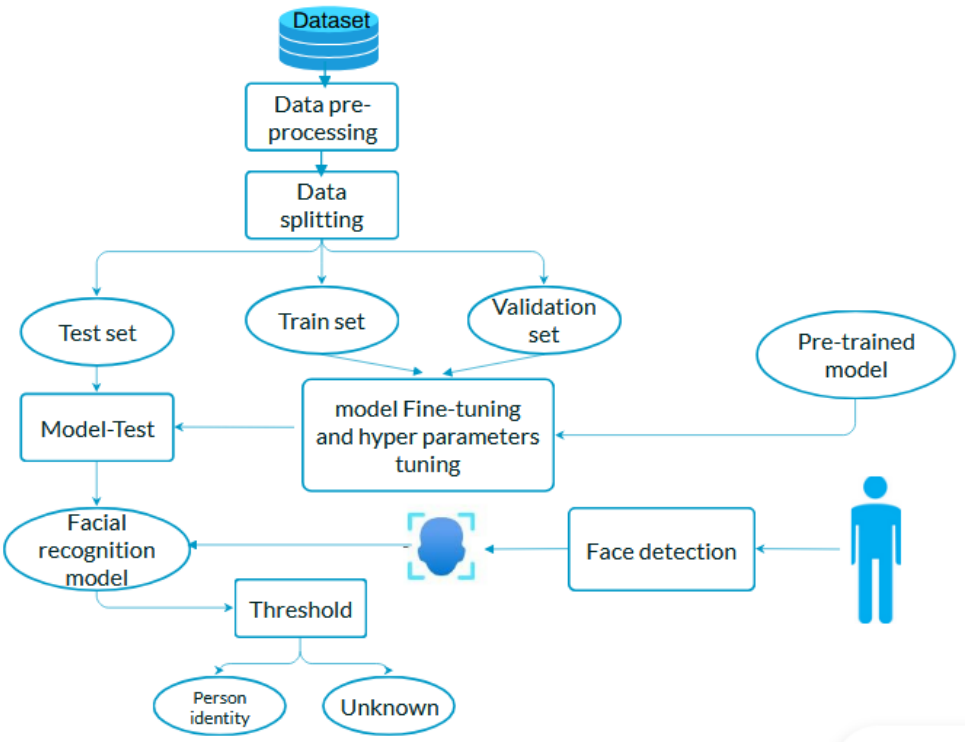

Figure 3. 1: General schema of the proposed approach

3.2.1. DATASET

The initial stage is dataset collection, which can be approached in two primary ways. The first method involves gathering data from various sources, such as controlled photo sessions, real-world scenarios, or user-submitted images. This approach allows for customization but requires significant effort in data collection, preprocessing, and annotation. The second method, which we have adopted in this work, utilizes standard datasets readily available in the field of facial recognition, such as LFW (Labeled Faces in the Wild). These datasets offer pre-collected, often pre-processed, and labeled facial images, saving time and resources. Regardless of the chosen method, it's important to ensure the dataset's diversity in terms of age, gender, ethnicity, lighting conditions, and poses to enhance the model's generalization capabilities. The selected dataset should be split into training, validation, and test sets, and any ethical considerations or privacy concerns must be addressed, particularly when dealing with personal data. The quality and representativeness of this dataset directly impact the facial recognition system's performance and real-world applicability.

3.2.2. DATA PRE-PROCESSING

The second stage involves preprocessing the dataset to ensure its suitability for subsequent analysis. This process involves resizing and normalization to ensure that the data is appropriately formatted and ready for training and evaluation. Resizing is an important first step in image preprocessing, as images come in all shapes and sizes. Normalizing the pixel values of the images is essential for enhancing the convergence and performance of the learning model. It helps stabilize the learning process and improves the overall efficiency and effectiveness of the model.

3.2.3. DATA SPLITTING

After data preprocessing, to ensure the generalizability of our proposed approach, it is crucial to split dataset into three segments: the training set, validation set, and test set. This will allow us to realistically measure our model performance by ensuring that the dataset used to train the model and the dataset to evaluate it are distinct, we will explain our division of the datasets as follows:

- **Training set:** The training set is a portion of the dataset reserved to fit the model, and this later, sees and learn from the data in the training set to directly improve its parameters. To maximize our model performance the training set must be large enough to yield meaningful results.

- **Validation set:** The validation set is the set used to evaluate and tuning the hyper-parameters of the pre-trained models during training. Helping to assess the model's performance and make adjustments. By evaluating a trained model on the validation we gain insight into its ability to generalize to unseen data. This assessment helps identify potential issues such as ovefitting, which can have a significant impact on the model performance in real-world scenarios. And is also essential for hyper-parameters tuning. Hyper-parameters are settings that control the behavior of the model such as learning rate and batch size.

- **Testing set:** The test set used to evaluate the final performance of a trained model. The test dataset serves as an impartial gang of the model's generalization to unseen data, providing a benchmark of its performance in real world scenario by maintaining the test set separate throughout development, we ensure a reliable assessment of the model's capabilities. It also allows an unbiased evaluation of the model ability to handle new data. Throughout this evaluation we determine the model's proficiency in recognizing relevant patterns and making accurate predictions beyond the training and validation phases.

3.2.4. MODEL FINE-TUNING

In this step, various pre-trained models are fine-tuned on specific datasets dedicated to the person identification task. The fine-tuning process involves adjusting the pre-trained models' parameters to better suit the unique characteristics of the new datasets. By utilizing the learned features from the original training, these models can achieve higher accuracy and performance in identifying individuals from facial images.

In this step, various pre-trained models are fine-tuned on specific datasets dedicated to the person identification task. The fine-tuning process involves adjusting the pre-trained models' parameters to better suit the unique characteristics of the new datasets. This process includes:

- **Selecting appropriate pre-trained models:** We explore different architectures such as CNNs, including EfficientNet, DenseNet, ResNet-50, VGG16, and MobileNet, known for their robust feature extraction capabilities in image recognition tasks. Additionally, we assess Vision Transformer (ViT) models, including ViT-B8, ViT-S16, and ViT-L16, which utilize self-attention mechanisms to capture complex patterns in image data. Our aim is to determine the most effective approach for this task.
- **Modifying the model architecture:** The final layers of the pre-trained model are often replaced with new layers specific to the person identification task, such as a new classification layer with the number of outputs matching the number of individuals in the dataset.

- **Freezing early layers:** The initial layers of the pre-trained model, which capture low-level features, are frozen to preserve the general feature extraction capabilities learned from large-scale datasets.
- **Training on the new dataset:** The model is then trained on the person identification dataset, with a focus on updating the weights of the unfrozen layers.
- **Hyperparameter tuning:** For each model, we tune the hyperparameters using our preprocessed dataset to optimize their performance. This process involves adjusting parameters such as learning rates, batch sizes, and epochs to achieve the best possible results.

3.2.5. MODEL TEST

After fine-tuning, we evaluate the models based on performance metrics such as accuracy, precision, recall, and F1-score, allowing us to select the best-performing model for our facial recognition system. This evaluation process includes making predictions on the test dataset, calculating metrics by comparing the predicted labels to the actual labels.

3.2.6. PERSON IDENTIFICATION

Person identification is achieved through a two-stage process: Face detection and identity determination using the fine-tuned model.

3.2.1.1 FACE DETECTION

This step involves the integration of a face detection model, specifically employing the Haar Cascade classifier, to detect faces within the input images. By incorporating this face detection mechanism, we ensure that our system accurately identifies and extracts facial features from the images before proceeding to further analysis. This is essential for enhancing the reliability and robustness of our facial recognition system, as accurate detection of facial regions is fundamental for subsequent processing steps.

3.2.1.2 IDENTITY DETERMINATION

Once the face is detected and extracted, the fine-tuned model computes probability scores for each class (person). Before assigning the identity based on the highest probability, the system employs a similarity threshold determined during the training phase. This threshold serves as a decision boundary for accepting or rejecting an identification based on facial features.

The threshold is established using a similarity score (or distance metric) between the input face and the stored facial data. If the computed similarity score surpasses this threshold, the system accepts the identification and confirms the person's identity. Conversely, if the score falls below the threshold, the system rejects the identification, indicating that the input face does not sufficiently match any stored data.

3.3. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we will discuss the datasets we are using and how we process them through preprocessing. Next, we will conduct hyper-parameter tuning, where we will experiment with pre-trained models using batch size, number of epochs, dropout, and learning rate. Finally, we will select the best model and compare it with recent work on LFW datasets.

3.3.5. DATASETS USED

Selecting appropriate dataset is crucial for any study. In our face recognition transfer learning approach, we opted to use two types of datasets from Kaggle namely Labelled Faces in the Wild (LFW) and Pins Face Recognition (PINS-FR), to obtain an effective outcome for the system. LFW contains 13000 images of faces with labels. PINS-FR is a collection of celebrity images sourced from pintrest (105 celebrity) and 17534 faces. The use of two different datasets, rather than only one, is motivated by the fact that this allows us to generalize our findings concerning the most suitable model for person identification task.

Shown in (Figure 3. 2) are samples from the LFW dataset, while (Figure 3. 3) displays samples from the PINS-FR dataset.

Figure 3. 2 : Sample LFW dataset.

Figure 3.3 : Sample PINS-FR dataset.

3.3.6. DATA PREPARATION

When we analyzed the distribution of images across different classes to identify anomalies, and most importantly, to prepare the datasets (LFW and PINS-FR) for the crucial step of preprocessing, we found that the datasets contain some classes with a higher number of images than others. This situation is known as class imbalance. To organize and select balanced classes, we set a threshold to include only those individuals who had a sufficient number of images. For both the LFW and PINS-FR datasets, rigorous filtering was applied to ensure adequate samples for training and testing. In the case of the LFW dataset, individuals with fewer than 50 face images per class were excluded, guaranteeing sufficient data for analysis. Similarly, for the PINS-FR dataset, classes with fewer than 170 face images were omitted based on distribution analysis, ensuring robust sample sizes for effective modeling and evaluation.

Upon completion of data preparation, the ensuring stage of pre-processing will be the focal point of our discussion we resized the datasets images to 224x224 pixels, we applied the same resizing to both the LFW and PINS-FR datasets, to ensure compatibility with our learning model and to simplify the training process. Also we normalized the images by scaling the pixel values to a range between 0 and 1, this was achieved by dividing the pixel values by 255, to ensuring that all features have a similar scale, thus preventing certain features from dominating others during training.

After data preprocessing, to ensure the generalizability of our proposed approach, it is crucial to split both datasets (LFW, and PINS-FR) into three segments: the training set, validation set, and test set.

We divided our datasets as follows: For LFW, we allocated 70% for training, 15% for validation, and 15% for testing, as shown in (Figure 3. 4). Specifically, we have 1,092 images for training, 234 images for validation, and 234 images for testing.

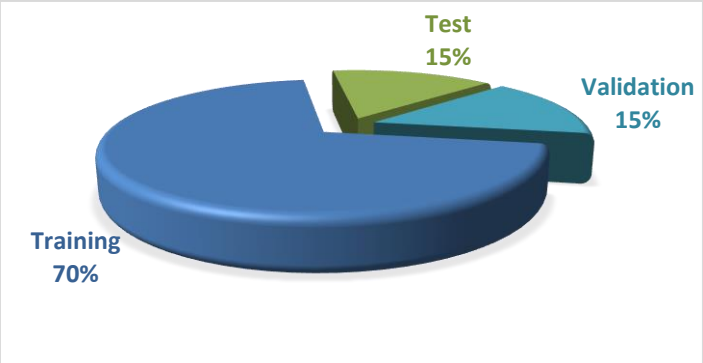

Figure 3. 3 : Splitting LFW data into training, validation and test sets.

For PINS-FR, as shown in (Figure 3. 5), we allocated 70% for training, 15% for validation, and 15% for testing. 6984 images for training, and 1497 for both training and testing.

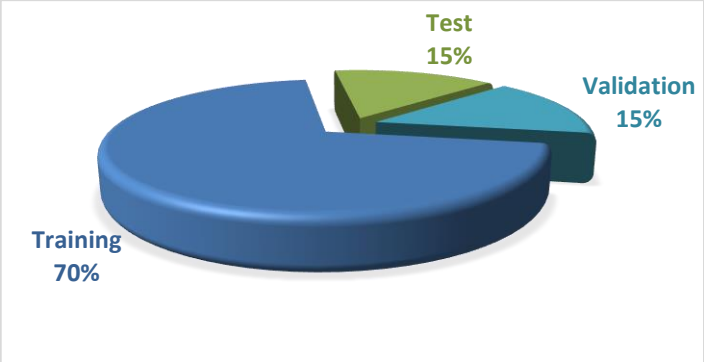

Figure 3. 4 : Splitting PINS-FR data into training, validation and test sets.

3.3.7. HYPER-PARAMETERS TUNING

In this section, our objective is to elucidate the experimental outcomes stemming from a spectrum of models pre-trained utilizing both CNN and ViT architectures. Our aim is to discern the most appropriate model, optimizing its parameters for superior performance. These experiments are meticulously crafted to conduct a comprehensive assessment of each architecture's efficacy and to discern the pivotal components and hyper-parameters that underpin optimal results. The study investigates the impact of several key hyper-parameters, including batch size, dropout rate, learning rate, and the number of epochs.

The analysis of these hyper-parameters will be conducted employing the EfficientNet model, while employing identical methodologies across other models.

3.3.7.1. BATCH SIZE

A batch refers to a subset of the training data that is processed together in one forward pass and one backward pass (one iteration) of the neural network. When training a neural network, it is common to divide the entire training dataset into smaller batches to improve computational efficiency [Web-69].

Processing data in batches helps in optimizing the training process by allowing the model to update its parameters more frequently compared to processing the entire dataset at once. We varied the batch size in different experiments on both LFW and PINS-FR, the results in terms of accuracy and loss are shown in (Figure 3. 5 – Figure 3. 8).

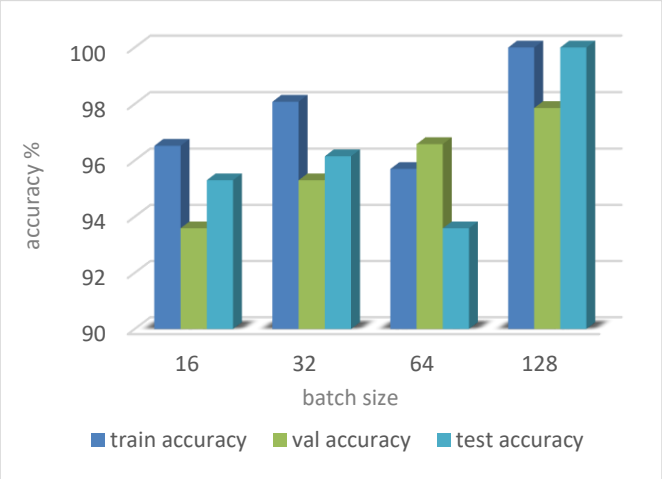

Figure 3. 5 : Accuracy evaluation of EfficientNet model with different batch sizes on LFW dataset.

In the histogram presented in Figure 3. 5, experiments were conducted using batch sizes of 16, 32, 64, and 128 on the LFW dataset. The analysis focused on accuracy results for training, testing, and validation datasets. The histogram illustrates that a batch size of 128 yielded the best performance, achieving a maximum accuracy of 100% in test, 100% in train, and 97.86% in validation. Consequently, we conclude that a batch size of 128 is optimal for EfficientNet model.

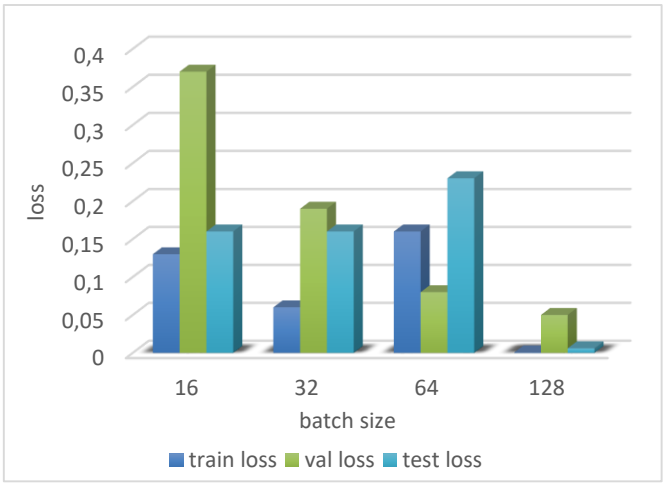

Figure 3. 6 : Loss evaluation of EfficientNet model with different batch sizes on LFW dataset.

From the histogram of Figure 3. 6, which evaluates the loss of EfficientNet, it is evident that the larger batch size of 128 helps in achieving better generalization in this scenario, as indicated by the lower validation and test losses compared to smaller batch sizes.

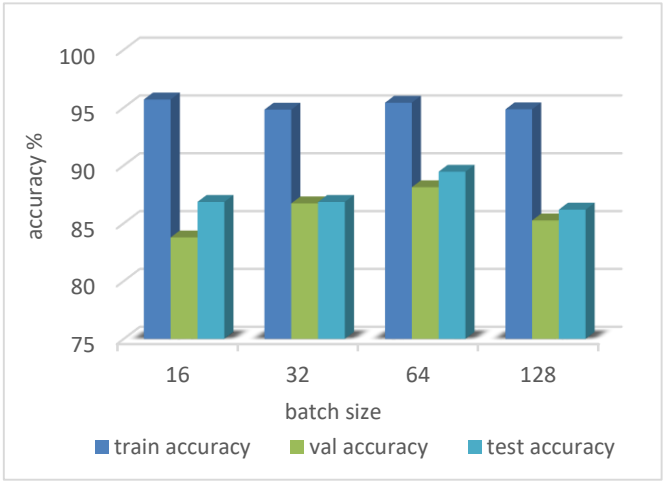

Figure 3. 7 : Accuracy evaluation of EfficientNet model with different batch sizes on PINS-FR dataset.

We conducted a similar experiment on the PINS dataset. From histogram Figure 3. 7, it can be observed that the optimal batch size corresponds to 16, as evidenced by its superior performance in terms of achieving the highest accuracy it reaches 86.84% in test, 95.7% in train, and 83.77% in validation.

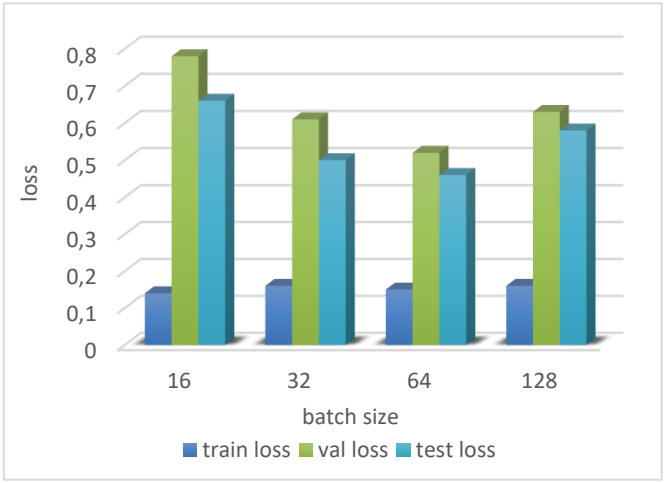

Figure 3. 8 : Loss evaluation of EfficientNet model with different batch sizes on PINS-FR dataset.

From the histogram in Figure 3. 8, the best batch size appears to be 64, as it results in the lowest validation (0.52), test (0.46), and train (0.15) losses. This batch size leads to the best overall model performance in this context.

3.3.7.2. DROPOUT RATE

Dropout is a technique where randomly selected neurons are ignored during training. They are 'dropped out' randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass, and any weight updates are not applied to the neuron on the backward pass (Brownless, 2022).

If neurons are randomly dropped out of the network during training, other neurons will have to step in and handle the representation required to make predictions for the missing neurons. This is believed to result in multiple independent internal representations being learned by the network, this, in turn, results in a network capable of better generalization and less likely to overfit the training data.

In our experiment, we aim to determine the best dropout rate for regularization in order to alleviate overfitting and enhance the generalization capabilities of our fine-tuned model. The objective is to identify a dropout rate that ensures the model performs effectively on both the training set and new, unseen data (test set). The outcomes in terms of accuracy and loss of this experiment on both the LFW and PINS-FR datasets, conducted with various dropout rates, are presented in (Figure 3. 9 - Figure 3. 12).

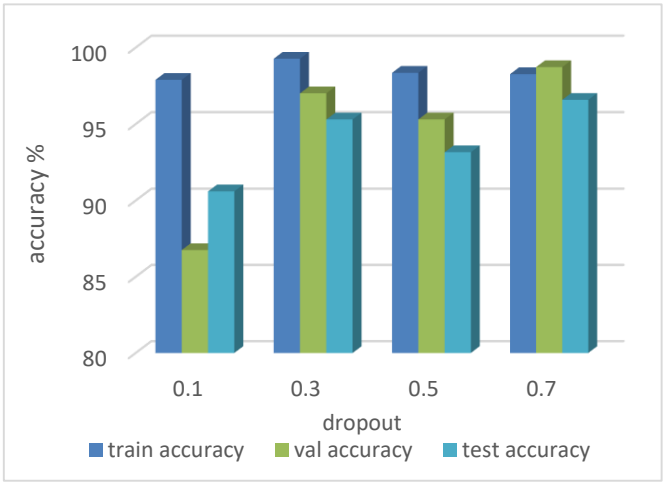

Figure 3. 9 : Accuracy evaluation of EfficientNet model with different dropout rates on LFW dataset.

As indicated by the results in the histogram of Figure 3. 9, the dropout value of 0.7 is the best, where the test accuracy reaches 96.58% (unseen data), 98.26% in train, and 98.72% in validation, compared to the other values in the experiment that reduce the efficiency of our model.

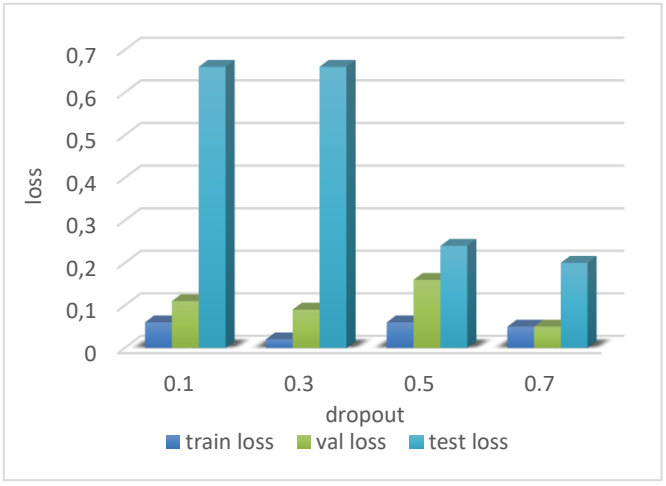

Figure 3. 10 : Loss evaluation of EfficientNet model with different dropout rates on LFW dataset.

As indicated in the histogram illustrated in Figure 3. 10, a dropout rate of 0.7 provides the best trade-off between training loss 0.57 and generalization performance, with the lowest validation 0.51, and test losses 2.02. This dropout rate effectively alleviates overfitting and enhances the model's ability to perform well on unseen data.

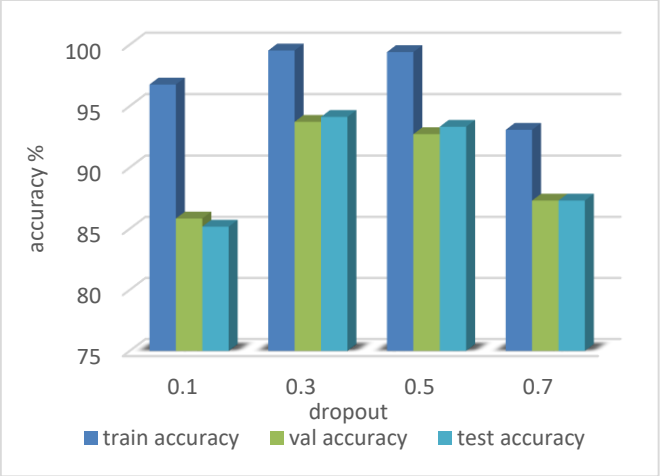

Figure 3. 11 : Accuracy evaluation of EfficientNet model with different dropout rates on PINS-FR dataset.

On PINS-FR dataset, a dropout rate of 0.3, as illustrated in the histogram shown in Figure 3. 11, achieved the best test accuracy, reaching 94.12 %, this accuracy decreased when experimenting with other dropout values.

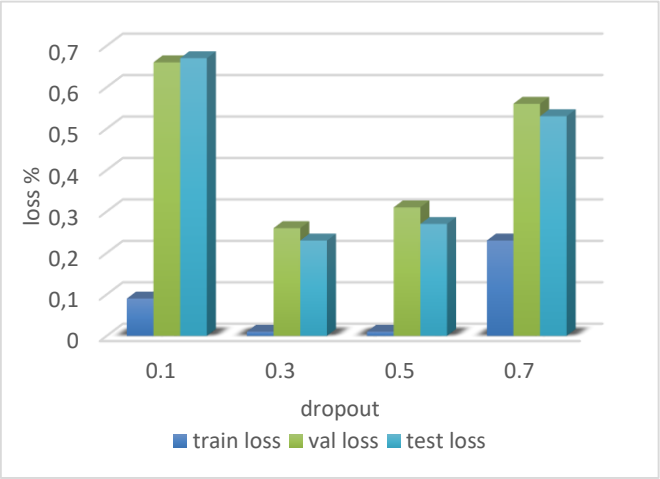

Figure 3. 12 : Loss evaluation of EfficientNet model with Different dropout on PINS-FR dataset.

In the loss evaluation, as illustrated in Figure 3. 12, a dropout rate of 0.3, achieved the lowest loss in test (0.23), train (0.01), and validation (0.26). So, this value of dropout enhances the model's ability to perform well on unseen data.

3.3.7.3. LEARNING RATE

The learning rate is one of the most important hyper-parameter for fine-tuning the model, the learning rate controls the step size during the model's weights updates. The choice of learning rate is pivotal, a high learning rate enables the model to learn rapidly by taking larger strides, but risks overshooting the minimum loss. On the other hand, a low learning rate ensures careful, albeit slower, progress, potentially causing the model to become stuck in local minima.

To determine a good balance between underfitting and overfitting, we experimented with different learning rates, the results are presented in (Figure 3. 13 – Figure 3. 16).

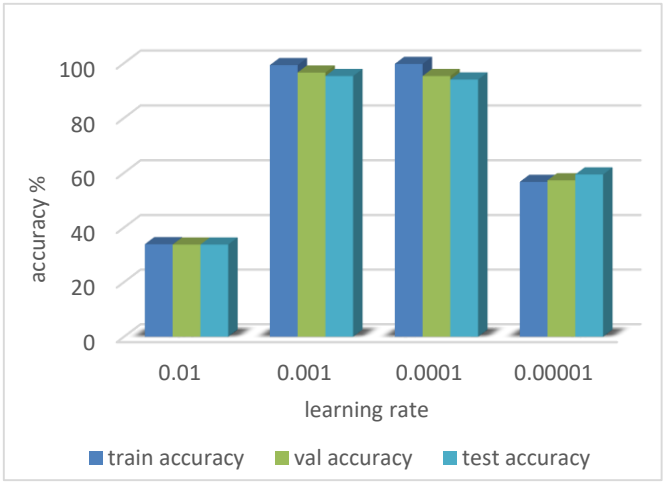

Figure 3. 13 : Accuracy of EfficientNet model with different learning rate on LFW dataset.

On LFW dataset, Figure 3. 13 indicates that the best learning rate is 0.001, achieving a highest accuracy, 95.3% on the test set, means that performs better on unseen data, and 99.27% in train set, and 96.58%. In contrast, other learning rate values reduce the efficiency of our model.

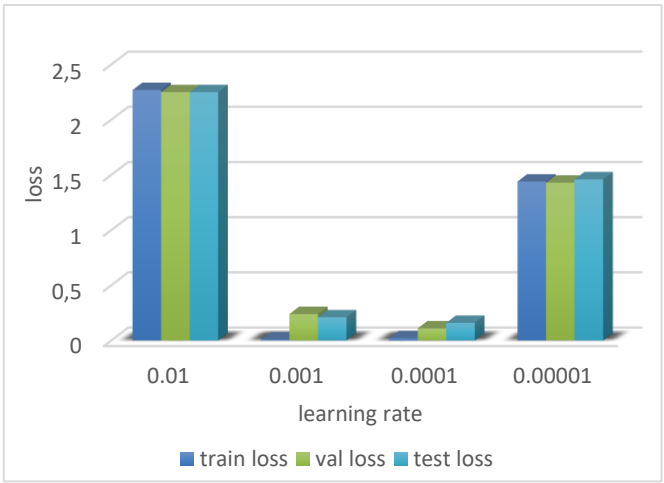

Figure 3. 14 : Loss evaluation of EfficientNet model with different learning rate on LFW dataset.

Figure 3. 14 shows that the best learning rate is 0.0001, because it results in low loss, in test loss (0.16), validation loss (0.11), and train loss (0.02), thus indicating the best generalization to new data. Learning rates of 0.01 and 0.00001 show poor performance due to overfittting and underfitting.

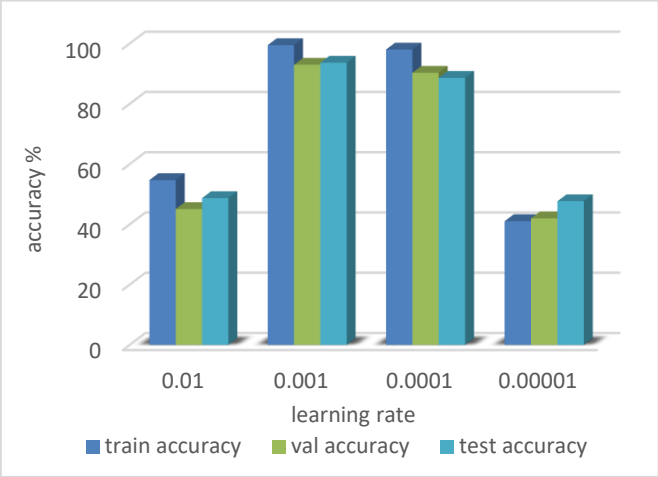

Figure 3. 15 : Accuracy evaluation of EfficientNet model with Different learning rate on PINS-FR dataset.

Our experiment on the PINS-FR dataset, as shown in Figure 3. 15, indicates that the best value for the learning rate is 0.001. This learning rate achieves the highest training accuracy 99.48%, indicating a strong fit to the training data. Additionally, the validation and test accuracies are high (92.97%, 93.65%, respectively) and slightly better than those achieved with the 0.0001 learning rate and other values. This suggests that the model generalizes well with a learning rate of 0.001.

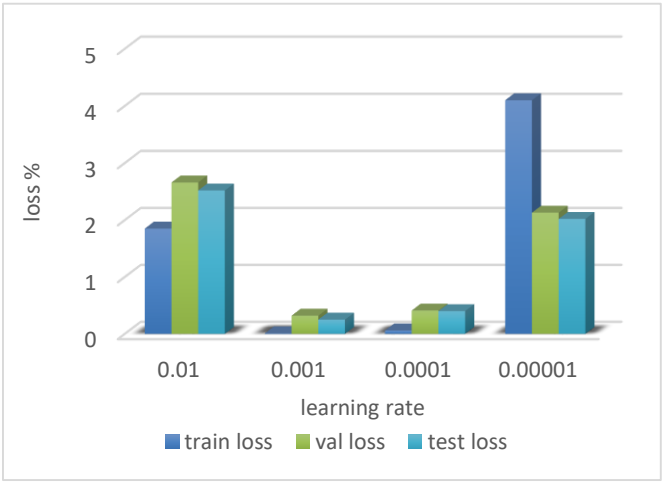

Figure 3. 16 : Loss evaluation of EfficientNet model with Different learning rate on PINS-FR dataset.

As depicted in Figure 3. 16, the learning rate of 0.001 achieves the lowest losses across training (0.01), validation (0.32), and test set (0.25). These results strongly indicate that the model performs better with this learning rate.

3.3.7.4. EPOCHS NUMBER

An epoch is a complete pass through the entire training dataset, during which the model processes all the training examples once. In other words, one epoch consists of multiple iterations, where each

iteration processes one batch of data. After completing one epoch, the model has seen and learned from all the training examples in the dataset. Training typically involves running multiple epochs to improve the model's accuracy and reduce the loss function.

To avoid overfitting, we applied early stopping. Early stopping in epochs is a powerful technique, because it allows the training process to automatically determine the appropriate number of epochs needed for training, rather than relying on a fixed number that may lead to overfitting. To study the behavior of our model during the training phase based on the number of epochs, several experiments were conducted. The results are illustrated by the curves in (Figure 3. 17 - Figure 3. 20)

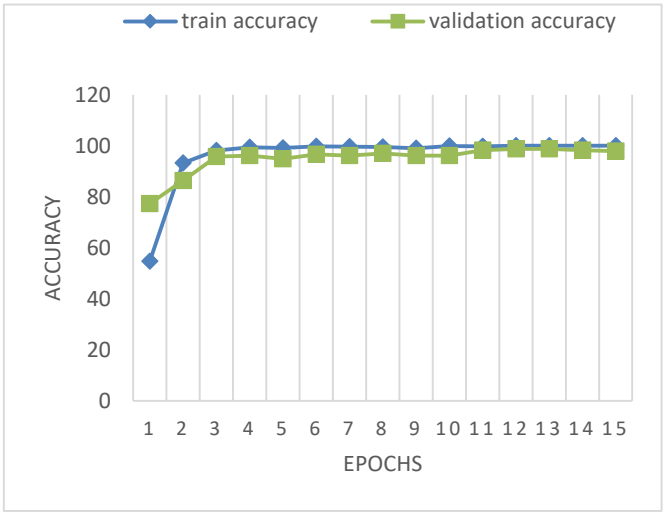

Figure 3. 17 : Accuracy of EfficientNet model with varying epochs number on LFW dataset.

In the curve illustrated in Figure 3. 17, for (epochs 1-3), the model undergoes significant improvement in both training and validation accuracies, indicating effective learning. For (Epochs 4-10), both training and validation accuracies reach high levels and stabilize, suggesting that the model has learned the data well and generalizes effectively. Concerning the range (epochs 11-15), although the training accuracy approaches 100%, the validation accuracy plateaus or slightly decreases.

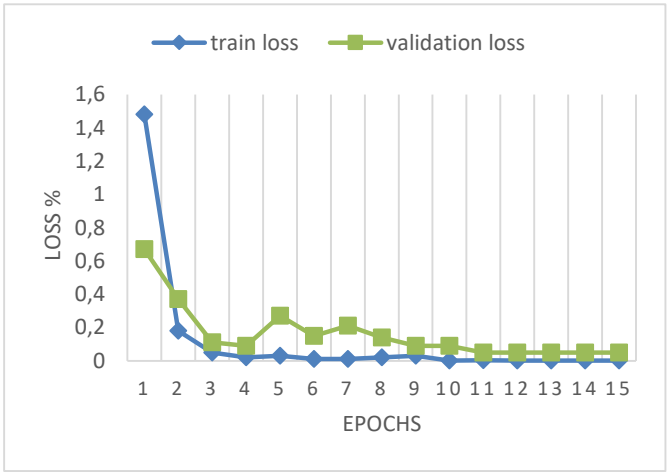

Figure 3. 18 : Loss evaluation of EfficientNet model with varying epoch's number on LFW dataset.

In the loss evaluation, reduction in training loss across epochs indicates that the model is fitting the training data well, improving its ability to capture patterns and minimize errors. As the validation loss decreases initially but starts to plateau after Epoch 10, it suggests that the model's generalization performance stabilizes. Despite further training, the validation loss doesn't decrease significantly, indicating that the model does not significantly overfit and is performing well on unseen data.

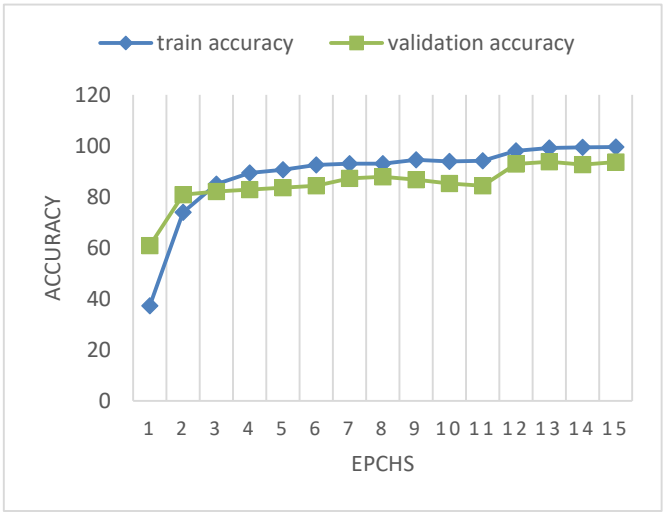

Figure 3. 19 : Accuracy of EfficientNet model with varying epoch's number on PINS-FR dataset.

In PINS-FR dataset, as shown in Figure 3. 19, from epoch 1 to 3, both training and validation accuracies show significant improvement, indicating effective learning in the early stages. From epoch 4 to 10, training accuracy continues to increase gradually, while validation accuracy fluctuates around a relatively stable value. This suggests that the model is learning well from the training data, but its performance on unseen data is not consistently improving. From epoch 11 to epoch 15, training accuracy continues to increase, approaching near-perfect accuracy (93.58%). Validation accuracy remains relatively stable, indicating that the model's generalization capability has peaked.

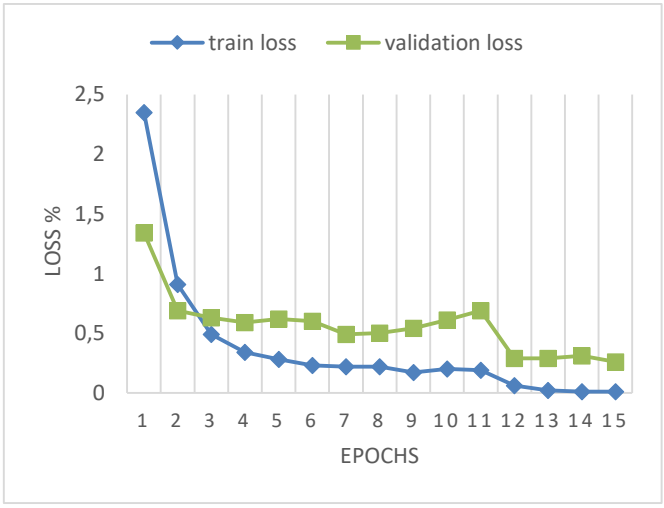

Figure 3. 20 : Loss evaluation of EfficientNet model with varying epoch's number on PINS-FR dataset.

In loss evaluation on PINS-FR datasets, as shown in the curve of Figure 3. 20, in the initial phase of training (epochs 1-3), both training and validation losses see substantial decreases, signaling swift learning and better adaptation to the data. As training progresses (epochs 4-10), the pace of improvement slows, with fluctuations in validation loss hinting at inconsistent generalization performance despite ongoing training efforts. In the final optimization phase (epochs 11-15), while training loss continues to decrease, validation loss remains relatively stable, indicating the model has reached its maximum generalization capacity.

Through experiments conducted on the LFW and PINS-FR datasets, we performed extensive tuning of various hyper-parameters, including the number of epochs, batch size, dropout rate, and learning rate. These experiments encompassed an exploration of diverse pre-trained CNN architectures, as well as ViT pre-trained models. Our objective was to empirically identify the most effective combination of hyper-parameters and pre-trained models to obtain the best-performing facial recognition system. The best hyper-parameter configurations are presented in Table 3.1, and Table 3. 2.

Table 3. 1 : Pre-trained models with best hyper-parameters on LFW dataset.

Models	Batch size	Dropout	Learning rate	Epochs
Efficient-Net	128	0.2	0.001	15
MobileNet	64	0.2	0.001	12
Resnet-50	64	0.2	0.0001	8
VGG16	64	0.3	0.0001	15
Dense-net	64	0.1	0.0001	12
Vit-B8	64	0.2	0.001	10
Vit-S16	64	0.2	0.001	12
Vit-L16	64	0.1	0.001	15

Table 3. 2, represents the most effective combination of hyper-parameters and pre-trained models to obtain the best-performing facial recognition system on PINS-FR dataset.

Table 3. 2 : Pre-trained models with best hyper-parameters on PINS-FR dataset.

Models	Batch size	Dropout	Learning rate	Epochs
Efficient-Net	64	0.3	0.001	15
MobileNet	32	0.4	0.001	12
Resnet-50	64	0.1	0.001	15
VGG16	64	0.1	0.0001	14
Dense-net	64	0.1	0.0001	13
Vit-B8	32	0.3	0.001	15
Vit-S16	128	0.4	0.001	14
Vit-L16	64	0.1	0.001	15

3.4. COMPARISON OF RESULTS

Through this experimental study, we aim to identify the most suitable model architecture that yields optimal performance for our facial recognition system. Below, we will conduct a comparative analysis of different models trained using the best respective combinations of hyper-parameters. Subsequently, we will compare the performance of the selected model with existing literature for the LFW database in order to evaluate our approach relatively to the current state of the art. Regarding the PINS-FR dataset, there is currently no prior research available as it is a novel database.

Table 3. 3 : Performance comparison of pre-trained models on LFW dataset.

Models	**Accuracy**			**Loss**		
-	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
EfficientNet	100 %	97.86 %	100 %	0.0006	0.05	0.006
MobileNet	99.08 %	93.16 %	92.27 %	0.64	1.04	1.02
Resnet-50	100 %	96.58 %	96.15 %	0.34	0.45	0.46
VGG16	92.49 %	87.61 %	91.02 %	0.22	0.38	0.27
Densenet	100 %	97.86 %	97.86 %	0.004	0.07	0.05
Vit-B8	97.34 %	91.88 %	95.29 %	0.08	0.23	0.18
Vit-S16	95.51 %	85.47 %	89.31 %	0.12	0.46	0.36
Vit-L16	100 %	96.15 %	94.87 %	0.003	0.16	0.16

Table 3. 3, presents a comparative analysis of various CNN models, EfficientNet, MobileNet, ResNet-50, VGG16, Densenet, and ViT models (ViT-B8, ViT-S16, ViT-L16), on LFW dataset, across training, validation, and test accuracy and loss metrics. EfficientNet outperforms all other models, achieving perfect training accuracy (100%), and the highest validation (97.86%) and test (100%) accuracy. It also demonstrates the lowest validation and test losses (0.05% and 0.006% respectively), indicating superior generalization. These results highlight EfficientNet's efficacy in both learning capacity and generalization. Making it the most robust and reliable model among those compared.

Table 3. 4 : Performance comparison of pre-trained models on PINS-FR dataset.

Models	**Accuracy**			**Loss**		
-	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
Efficient-Net	99.53 %	93.72 %	94.12 %	0.01	0.26	0.23
MobileNet	96.86 %	79.09 %	80.16 %	0.68	1.52	1.46
Resnet-50	88.03 %	72.08 %	73.61 %	1.17	1.80	1.75
VGG16	99.26 %	64.13 %	66.19 %	0.05	1.40	1.35
Dense-net	93.13 %	75.68 %	77.42 %	2.23	0.95	0.88
Vit-B8	74.21 %	59.19 %	60.65 %	0.80	1.49	1.43
Vit-S16	58.29 %	48.90 %	50.90 %	1.36	1.76	1.71
Vit-L16	95.96 %	68.87 %	69.53 %	0.14	1.39	1.25

Table 3. 4 presents a comparative analysis of various models, including EfficientNet, MobileNet, ResNet-50, VGG16, DenseNet, and Vision Transformers (ViT-B8, ViT-S16, ViT-L16), on the PINS-FR dataset, across training, validation, and test accuracy and loss metrics. EfficientNet outperforms all other models, achieving an outstanding training accuracy of 99.53%, and the highest validation (93.72%) and test (94.12%) accuracies. It also demonstrates the lowest validation and test losses (0.26 and 0.23 respectively), indicating superior generalization and robustness. Making it the most robust and reliable model among those compared.

Table 3.5, and Table 3.6, provides the classification performance metrics for various models on the LFW dataset and PINS-FR dataset. The metrics reported are macro average and weighted average for precision, recall, and F1-score. The models evaluated include EfficientNet, MobileNet, ResNet-50, VGG16, DenseNet, ViT-B8, ViT-S16, and ViT-L16, each tested on a support of 234 instances.

Table 3.5 : Classification report on LFW dataset.

Models	**Macro avrage**			**Weighted avrage**			**Test accuracy**
-	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**-**
Efficient-Net	1.00	0.99	0.99	1.00	1.00	1.00	100
MobileNet	0.93	0.91	0.91	0.94	0.93	0.93	92.27
Resnet-50	0.98	0.94	0.96	0.96	0.96	0.96	96.15
VGG16	0.90	0.88	0.88	0.93	0.91	0.91	91.02
Dense-net	0.97	0.95	0.96	0.98	0.98	0.98	97.86
Vit-B8	0.95	0.95	0.95	0.96	0.95	0.95	95.29
Vit-S16	0.91	0.87	0.88	0.90	0.89	0.89	89.31
Vit-L16	0.96	0.94	0.95	0.96	0.95	0.95	94.87

In Table 3.5, EfficientNet achieves the highest performance metrics, indicating its superior suitability for the LFW dataset. DenseNet, ResNet-50, and the ViT models (particularly ViT-B8 and ViT-L16) also show strong performance. MobileNet and VGG16, while effective, exhibit lower scores compared to the top-performing models. Vision Transformers, especially ViT-S16, display varied performance, with larger models performing better. EfficientNet's exceptional scores highlight its robustness and effectiveness in handling the LFW dataset.

Table 3.6 : Classification report on PINS-FR dataset.

Models	Macro avrage			Weighted avrage			Test accuracy
-	Precision	Recall	F1-score	Precision	Recall	F1-score	-
Efficient-Net	0.94	0.94	0.94	0.94	0.94	0.94	94.12
MobileNet	0.84	0.80	0.80	0.85	0.80	0.80	80.16
Resnet-50	0.78	0.74	0.74	0.78	0.71	0.74	73.61
VGG16	0.70	0.66	0.66	0.70	0.66	0.66	66.19
Dense-net	0.83	0.77	0.78	0.84	0.77	0.78	77.42
Vit-B8	0.63	0.61	0.61	0.64	0.61	0.60	60.65
Vit-S16	0.52	0.51	0.51	0.53	0.51	0.51	50.90
Vit-L16	0.71	0.70	0.70	0.71	0.70	0.69	69.53

In Table 3.6, EfficientNet demonstrates the highest performance on the PINS-FR dataset, suggesting its robust feature extraction and classification capabilities. MobileNet also performs well, though slightly lower than EfficientNet. Traditional CNNs like ResNet-50 and DenseNet show moderate performance, with ResNet-50 slightly underperforming DenseNet in recall. VGG16 exhibits the lowest performance among traditional CNNs. Vision Transformers, particularly ViT-S16 and ViT-B8, show lower performance compared to CNNs, indicating they might not be as well-suited for this dataset. EfficientNet's consistent high metrics underline its effectiveness for this classification task.

3.4.5. CONFUSION MATRIX

Figure 3. 21, depicting a confusion matrix for the LFW dataset, offers valuable insights into our facial recognition model's performance. By examining diagonal dominance, off-diagonal patterns, and class imbalances, we gain a comprehensive understanding of the model's accuracy and areas for improvement. This analysis provides actionable insights to refine the model and enhance its performance across different classes. Serves as a valuable tool for evaluating the performance of our facial recognition model. The model performs exceptionally well, achieving near-perfect precision across all classes, the slight deviations in precision for class 2, do not significantly impact the overall performance, this indicate the model's robustness and reliability for the dataset.

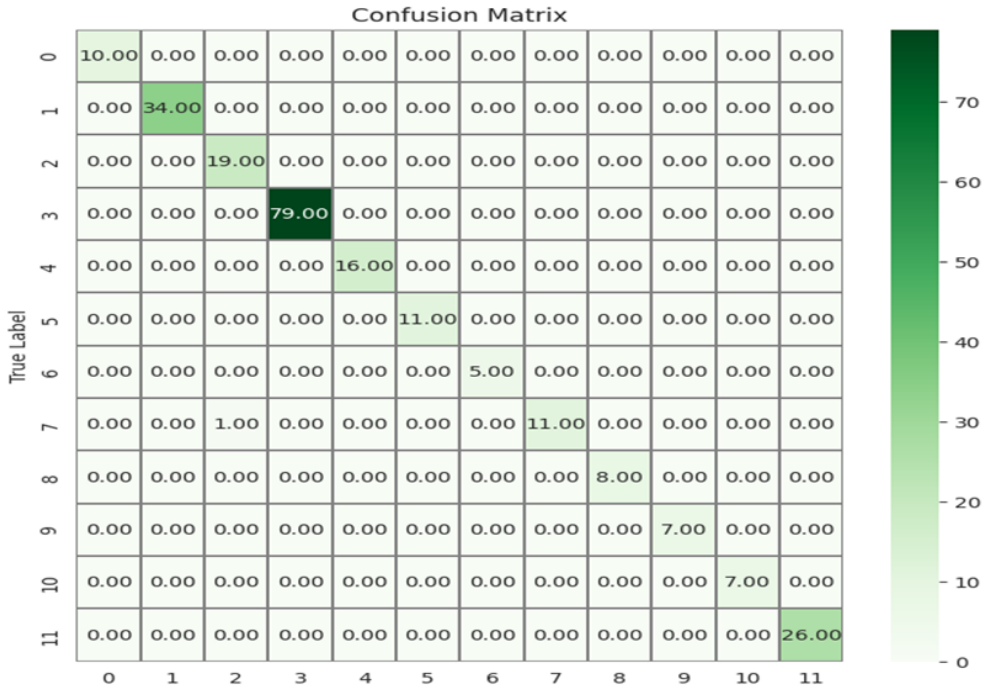

Figure 3. 21 : Confusion matrix of LFW dataset.

The Figure 3. 22, illustrating a confusion matrix for the PINS-FR dataset, there are sporadic instances of misclassifications observed, particularly between classes 0 and 1, and classes 1 and 2, these misclassifications suggest that the model may not perform equally well across all classes. However, it is evident that the model exhibits strong overall performance with an accuracy of 94.12%.

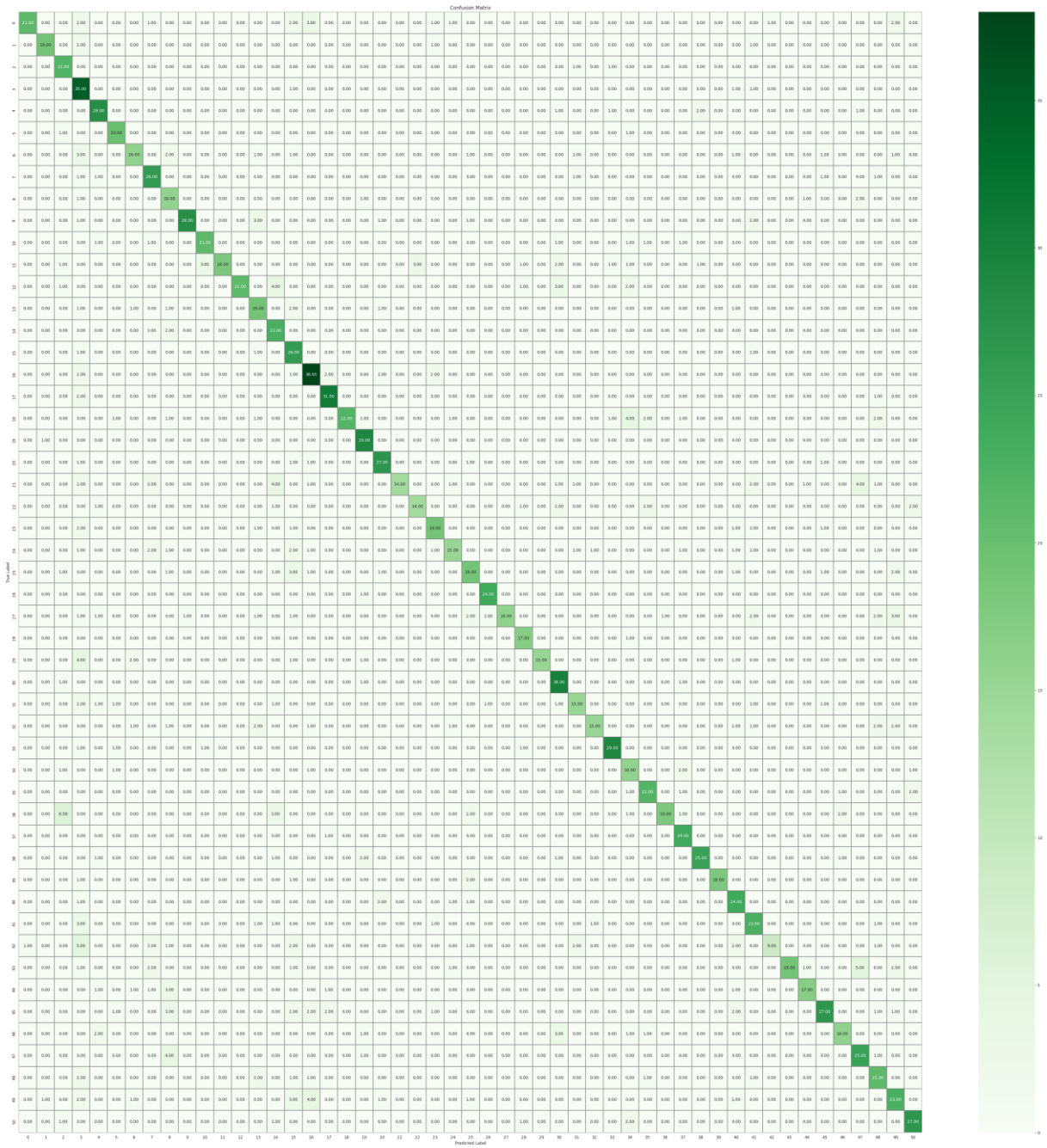

Figure 3. 22 : Confusion matrix of PINS-FR dataset.

3.4.6. COMPARISON WITH RECENT WORK IN LFW DATASET

The second part comprises comparing our model with a set of recent works that utilized the same dataset LFW. For the PINS-FR dataset, recent works associated with it are very rare because it was recently established. The comparison results are shown in (Table 3. 5).

Table 3. 5 : Comparative performance of EfficientNet model with recent work on LFW datasets.

System	Model	Accuracy
Samrat et al, 2021	Mobilnetv2	92.17%
Rajpal et al, 2022	VGG-16	97%
Wang, 2024	MobileNet-SSD	94.8%
Ul Haq et al, 2024	CapsNet	97.3 %
Our model	**EfficientNet**	**100 %**

From Table 3. 5, the comparison between MobileNetV2, VGG-16, MobileNet-SSD, and CapsNet, underscores a trade-off between efficiency and accuracy. While Samrat et al utilized Mobilenetv2, achieving 92.17% accuracy, highlighting its efficiency for resource-constrained applications. Rajpal et al used VGG-16, reaching a higher accuracy of 97%, reflecting the effectiveness of deeper networks at the cost of increased computational requirements. Wang combined MobileNet with SSD, resulting in 94.8% accuracy, indicating an improvement over Mobilenetv2 alone. Ul Haq et al demonstrated that CapsNet achieved 97.3% accuracy, underscoring its robustness in capturing spatial hierarchies and pose information. Our model, based on EfficientNet, achieved a perfect accuracy of 100%, showcasing its superior balance of depth, width, and resolution through compound scaling. These results suggest that while models like VGG-16 and CapsNet offer high accuracy, EfficientNet's remarkable performance may provide the best solution in balancing accuracy and computational efficiency. Our approach demonstrates that transfer learning can significantly enhance model performance, offering the best solution in terms of both accuracy and computational efficiency.

3.5. CONCLUSION

In this chapter, we have first presented our transfer learning approach for person identification from facial features. Then, we conducted an experimental study on both the LFW and PINS-FR datasets to identify the optimal pre-trained model for facial recognition tasks. Through a series of experiments, we systematically explored different architectures of pre-trained CNN and ViT models, tuning various hyper-parameters to optimize model performance. Our investigation concluded that EfficientNet model emerged as the most effective, achieving an outstanding accuracy of 100% on LFW dataset, and 94.12% on PINS-FR dataset. These results underscore the efficacy of transfer learning approaches in person identification task.

GENERAL CONCLUSION

In the area of person identification, facial recognition technology stands out as a powerful tool, offering a non-intrusive and efficient way of verifying individuals based on their unique facial features. Exploiting advancements in artificial intelligence and computer vision, facial recognition systems have shown remarkable accuracy and reliability, holding great promise for various applications spanning security, law enforcement, access control, and personalized services.

Our work explores harnessing the capabilities of facial recognition technology and transfer learning approach for person identification. To evaluate and enhance the performance of our facial recognition model, we conducted experiments utilizing two datasets, namely LFW and PINS. Through meticulous experimentation and rigorous analysis, we explored the efficacy of transfer learning technique in fine-tuning pre-trained models to excel in recognizing individuals based on their facial characteristics.

Our experimental study yielded promising results, showcasing the efficiency of our system with an impressive accuracy rate of 100% and 94.12% on LFW and PINS-FR datasets respectively. This achievement was made possible through careful adjustment of hyper-parameters such as batch size, dropout rate, and learning rate. Moreover, we conducted extensive comparative analyses with various CNN pre-trained models, including MobileNet, DenseNet, VGG16, and ResNet-50, as well as ViT pre-trained variants like ViT-B8, ViT-S16, and ViT-L16. And we have identified the EfficientNet architecture as the optimal solution for our facial recognition task.

The outcomes of our experiments underscore the effectiveness of leveraging transfer learning approach, particularly with the EfficientNet architecture, in enhancing person identification tasks based on facial features and recognition. By surpassing the performance of both conventional CNN architectures and advanced ViT models, our study not only achieves unparalleled accuracy but also sheds light on the broader potential of transfer learning in computer vision tasks.

Our research identifies several perspectives:

- Optimizing hyper-parameters: Developing techniques to select optimal hyper-parameters is vital for achieving high accuracy in our models. Utilizing methods like Bayesian Optimization Hyper-band, grid search and meta-heuristic methods can significantly enhance this process, ensuring our models perform at their best.
- Multi-modalities: Combining facial features with other modalities such as speech holds immense potential. This integration could enhance person identification systems by leveraging the unique strengths of different biometric methods, resulting in more reliable identification processes.
- Scaling up for national security: Beyond individual identification systems, there's an opportunity to expand our research towards establishing a comprehensive national security infrastructure. By integrating a cross-model approach into a larger multi-agent system, we could

revolutionize criminal tracking and security enforcement efforts, ultimately bolstering community safety on a broader scale.

These perspectives represent exciting avenues for future exploration and development in the field of person identification and national security

BIBLIOGRAPHY

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Andrejevic, M. (2019). "Facial recognition technology in schools: critical questions and concerns". *In : 'learning media and technology*, P: 118-120.

Mohan, A. S., & Abraham, L. (2024). An ensemble deep learning approach for air quality estimation in Delhi, India. *Earth Science Informatics*, 1-26.Ankan Bansal, A. N. (2016). "UMDFaces: An Annotated Face Dataset for Training Deep Networks". *In : ReshearchGate*.

Ato Adris, X. (2023). Machine learning per evitar obstacles amb turtlebot 3 burguer.biger-levin, A. (2022). "what is behavioral biometric". *In : biometric recognition and behavioral detection*, P: 67-70.

Kumar, C. R., Saranya, N., Priyadharshini, M., & Gilchrist, D. (2023). Face recognition using CNN and siamese network. *Measurement: Sensors*, *27*, 100800.

CHANGSHENG WAN, L. W. (2018). "A Survey on Gait Recognition". *In : ACM Computing Surveys*, P: 4-8.

Tomašević, D., Boutros, F., Damer, N., Peer, P., & Štruc, V. (2024). Generating bimodal privacy-preserving data for face recognition. *Engineering Applications of Artificial Intelligence*, *133*, 108495.

Gary, H. R.-M. (2008). "Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments". *In : Dans Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, P: 3.

Bae, G., de La Gorce, M., Baltrušaitis, T., Hewitt, C., Chen, D., Valentin, J., ... & Shen, J. (2023). Digiface-1m: 1 million digital face images for face recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3526-3535).

Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., & Guo, B. (2023, June). Peco: Perceptual codebook for bert pre-training of vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, No. 1, pp. 552-560).

Hugo, T., Cord, M., Matthijs, D., Francisco, M., Alexandre, S., & Herve, J. (2021, July). Training data-efficient image transformers & distillation through attention. In *ICML*.

Ikromovich, H. O., & Mamatkulovich, B. B. (2023). Facial recognition using transfer learning in the deep CNN. *Open Access Repository*, *4*(3), 502-507.

javaid, S. (2024). "Top 3 facial recognition challenges and solution in 2024". *In :AIMultiple.*

kaduwela, N. (2024). "Building a facial recognition solution". *In :LinkedIn.*

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16000-16009).

Kamil, M. Y. (2021). A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images. *International Journal of Electrical & Computer Engineering (2088-8708)*, *11*(1).

Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., & Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1931-1939).

kondrashov, S. (2023). "Unveiling the future:the impact of artificial intelligence on immercive gaming experiences". *In :LinkedIn.*

kutnyk, S. (2024). guide to face detection and recognition software development. *In :Medium.*

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*, 1-74.

Chambino, L. L., Silva, J. S., & Bernardino, A. (2020). Multispectral facial recognition: a review. *IEEE Access*, *8*, 207871-207883.

Manning, C. (2020). Artificial intelligence definitions. *Obtenido de Stanford University: https://hai. stanford. edu/sites/default/files/2020-09/AI-Definitions-HAI. pdf*.

Smith, M., & Miller, S. (2022). The ethical application of biometric facial recognition technology. *Ai & Society*, *37*(1), 167-175.

Cardaioli, M., Conti, M., Orazi, G., Tricomi, P. P., & Tsudik, G. (2023). BLUFADER: Blurred face detection & recognition for privacy-friendly continuous authentication. *Pervasive and Mobile Computing*, *92*, 101801.

Rahman, M. H., Jannat, M. K. A., Islam, M. S., Grossi, G., Bursic, S., & Aktaruzzaman, M. (2023). Real-time face mask position recognition system based on MobileNet model. *Smart health*, *28*, 100382.

Taye, M. M. (2023). Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers*, *12*(5), 91.

Patrick Doyle, J. A. (2019). "Law enforcement". *In : facial recognition use case catalog*, P: 6-10.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018, May). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)* (pp. 67-74). IEEE.

Devi, R. M., Keerthika, P., Suresh, P., Sarangi, P. P., Sangeetha, M., Sagana, C., & Devendran, K. (2022). Retina biometrics for personal authentication. In *Machine Learning for Biometrics* (pp. 87-104). Academic Press.

Gomes, R., Kamrowski, C., Langlois, J., Rozario, P., Dircks, I., Grottodden, K., ... & Haley, M. (2022). A comprehensive review of machine learning used to combat COVID-19. *Diagnostics*, *12*(8), 1853.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, *2*(3), 160.Sheldon, R. (2023). "What is a data set?". *In :TechTarget.*

Hangaragi, S., Singh, T., & Neelima, N. (2023). Face detection and Recognition using Face Mesh and deep neural network. *Procedia Computer Science*, *218*, 741-749.

Johnson, K. (2024). The Use of Clearview AI to Support Warrants Violates the Fourth Amendment. *Fordham Intellectual Property, Media and Entertainment Law Journal*, *34*(4), 991.

Dash, S. S. (2023). Face Recognition and Face Detection Benefits and Challenges. *no. July*, 177-191.

Srinivasa, K. G., Siddesh, G. M., & Sekhar, S. M. (Eds.). (2021). *Artificial Intelligence for information management: A healthcare perspective*. Springer.

TRISTAN FORO. (2023). "How Banks and Financial Institutions Can Use Facial Recognition to Protect People, Property, & Assets". *In :BriefCam.*

Han, X., Zhong, Y., Cao, L., & Zhang, L. (2017). Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing*, *9*(8), 848.

singh, d. (2023). “Dloa (part-20)- MobilNet CNN and implementation”. *In :Medium.*

Sridhar, p. k. (2023). “Exploring DenseNet: A concise Overview”. *In : mindfulmodeler.*

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.

Turc, I., Chang, M. W., Lee, K., & Toutanova, K. (2019). Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, *33*, 12449-12460.

Brownlee, J. (2022). *Machine learning mastery*. Machine Learning Mastery.

Rajpal, A., Sehra, K., Bagri, R., & Sikka, P. (2023). XAI-FR: explainable AI-based face recognition using deep neural networks. *Wireless Personal Communications*, *129*(1), 663-680.

Wang, S. (2024). A Face Recognition Method based on Lightweight Neural Network and Multi Hash Recognition Degree Weighting. *IAENG International Journal of Applied Mathematics*, *54*(3).

Ul Haq, M., Sethi, M. A. J., Ben Aoun, N., Alluhaidan, A. S., & Ahmad, S. (2024). CapsNet-FR: Capsule Networks for Improved Recognition of Facial Features. *Computers, Materials & Continua*, *79*(2).

Hisaria, S., Sharma, P., Gupta, R., & Sumalatha, K. (2024). An Analysis of Multi-Criteria Performance In *Deep Learning-Based Medical Image Classification: A comprehensive review.*

WEBOGRAPHY

[Web-1] Deep learning Vs Machine Learning. URL: https://k21academy.com/datascience-blog/deep-learning/dl-vs-ml/

[Web-2] What is logistic regression. URL: https://www.quora.com/What-is-logistic-regression

[Web-3] What is svm and how does it work. URL: https://techntales.medium.com/what-is-svm-and-how-does-it-work

[Web-4] Decision tree. URL: https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/

[Web-5] Random forest**.** URL: https://medium.com/@curryrowan/the-complete-guide-to-random-forests-part-2-934eabf35534

[Web-6]Understanding-polynomial-regression.URL:https://taherafirdose.medium.com/understanding-polynomial-regression-603eb25501d

[Web-7] URL: http://www.sthda.com/english/wiki/wiki.php?id_contents=7952

[Web-8] Reinforcement learning. URL: https://www.scribbr.com/ai-tools/reinforcement-learning/

[Web-9]Difference between supervised and unsupervised-learning. URL:https://www.javatpoint.com/difference-between-supervised-and-unsupervised-learning

[Web-10]Advantages and disadvantages of deep learning.URL: https://vngcloud.vn/en/blog/advantages-and-disadvantages-of-deep-learning

[Web-11] Hidden layer neural network. URL: https://www.coursera.org/articles/hidden-layer-neural-network

[Web-12] Forward propagation in neural networks.URL: https://towardsdatascience.com/forward-propagation-in-neural-networks-simplified-math-and-code-version

[Web-13] Neural-networks-activation-functions. URL: https://www.v7labs.com/blog/neural-networks-activation-functions

[Web14]IllustrationsofReLULeakyReLUPReLU.URL:https://www.researchgate.net/figure/Illustrations-of-ReLU-LeakyReLU-PReLU-and-the-proposed-adaptive-ReLU-AdaReLU-ReLU_fig1_342536629

[Web-15] Loss function in machine learning. URL: https://www.datacamp.com/tutorial/loss-function-in-machine-learning

[Web-16]When-to-use-recurrent-neural-networks.URL:https://iq.opengenus.org/when-to-use-recurrent-neural-networks-rnn/

[Web-17] Recurrent neural networks and lstm. URL: https://builtin.com/data-science/recurrent-neural-networks-and-lstm

[Web-18] The structure of the LSTM unit. URL: https://www.researchgate.net/figure/The-structure-of-the-LSTM-unit_fig1_347840605

[Web19]RecurrentNeuralNetworkwithaGatedRecurrentUnit.URL:https://www.researchgate.net/figure/Recurrent-Neural-Network-with-a-Gated-Recurrent-Unit

[Web-20] Recurrent-modern gru. URL: https://d2l.ai/chapter_recurrent-modern/gru.html

[Web-21] URL:https://insightsimaging.springeropen.com/articles

[Web-22] Transformer. URL: https://www.ibm.com/topics/transformer-model

[Web23]Whataretransformersconceptandapplicationsexplained.URL:https://www.marktechpost.com/2023/01/24/what-are-transformers-concept-and-applications-explained

[Web-24] Transfer learning. URL: https://www.techtarget.com/searchcio/definition/transfer-learning

[Web-25]CNNarchitecture. URL:https://medium.com/@daniyalmasoodai/pre-train-cnn-architectures-designs-performance-analysis-and-comparison-802228a5ce92

[Web-26] Vgg-16-cnn-model. URL: https://www.geeksforgeeks.org/vgg-16-cnn-model/

[Web-27] Training-vs-fine-tuning. URL: https://encord.com/blog/training-vs-fine-tuning

[Web-28] Transfer-learning. URL: https://botpenguin.com/glossary/transfer-learning

[Web-29] URL:https://h2o.ai/wiki/transfer-learning

[Web-30] Speaker recognition. URL: https://www.sciencedirect.com/topics/engineering/speaker-recognition

[Web-31] URL:https://www.pinterest.com/pin/977914506583358946/

[Web-32] URL:https://www.pinterest.com/pin/1106196727212726683/

[Web-33] Iris recognition. URL: https://www.innovatrics.com/iris-recognition-technology

[Web-34] URL:https://www.pinterest.com/pin/977914506583657553/

[Web-35] URL:https://www.pinterest.com/pin/198721402294863021/

[Web-36] The-dawn-of-dna-profiling. URL: https://www.yourgenome.org/theme/the-dawn-of-dna-profiling-the-eureka-moment-that-revolutionised-crime-solving

[Web-37] URL: https://www.pinterest.com/pin/606789749824341249/

[Web-38] URL: https://www.pinterest.com/pin/497366352612278905/

[Web-39]Facial-recognition-how-it-works. URL:https://datascientest.com/en/facial-recognition-how-it-works-2

[Web-40]Faces-detection-using-haar-cascade.URL:https://medium.com/@baselanaya/faces-detection-using-haar-cascade-3e175aef84f5haar

[Web-41] URL:https://www.youtube.com/watch?v=3rHkR3l8zyE

[Web-42] Multi-task-cascaded-convolutional-neural-network. URL: https://medium.com/the-modern-scientist/multi-task-cascaded-convolutional-neural-network-mtnn-a31d88f501c8

[Web-43] URL:https://towardsdatascience.com/robust-face-detection-with-mtcnn-400fa81adc2e

[Web-44] URL:https://www.cyberlink.com/faceme/insights/articles/228/how-to-use-facial-recognition

[Web-45] Facial recognition. URL: https://fitsmallbusiness.com/facial-recognition-in-retail

[Web-46] DigiFace1M. URL: https://github.com/microsoft/DigiFace1M

[Web-47] Msceleb. URL: https://exposing.ai/msceleb/

[Web-48] Ms-celeb-1m. URL: https://paperswithcode.com/dataset/ms-celeb-1m

[Web-49] URL:https://pan.baidu.com/s/1M7KF8IrcWCmzRprtahszcA

[Web-50] Megaface. URL: https://exposing.ai/megaface

[Web-51] Megaface dataset. URL: https://paperswithcode.com/dataset/megaface

[Web-52]Celeba-dataset.URL:https://www.kaggle.com/datasets/jessicali9530/celeba-dataset/download

[Web-53] CelebA. URL: https://mmlab.ie.cuhk.edu.hk/projects/CelebA

[Web-54] Celeba dataset. URL: https://paperswithcode.com/dataset/celeba

[Web55]Pinsfacerecognition.URL:https://www.kaggle.com/datasets/hereisburak/pinsfacerecognition/download

[Web-56] Pins-face-recognition dataset. URL: https://paperswithcode.com/dataset/pins-face-recognition

[Web-57] DigiFace1M. URL: https://github.com/microsoft/DigiFace1M

[Web-58] DigiFace-1M dataset. URL: https://paperswithcode.com/datasets?q=DigiFace-1M

[Web59]vggface2.URL:https://www.kaggle.com/datasets/hearfool/vggface2/download?datasetVersionNumber=1

[Web-60] URL:https://paperswithcode.com/dataset/vggface2

[Web-61] Umdfaces. URL: https://umdfaces.io/

[Web-62] Umdfaces dataset. URL: https://paperswithcode.com/dataset/umdfaces

[Web-63] The-IARPA-Janus-Benchmark. URL: https://www.researchgate.net/figure/The-IARPA-Janus-Benchmark-A-IJB-A-dataset-face-verification-11-test-protocol/download

[Web-64] Ijb-a dataset. URL: https://paperswithcode.com/dataset/ijb-a

[Web-65] URL:https://www.face-benchmark.org/download.html

[Web-66] Webface. URL: https://paperswithcode.com/dataset/webface260m

[Web-67] URL:https://www.kaggle.com/datasets/jessicali9530/lfw-dataset/download

[Web-68] URL:https://paperswithcode.com/dataset/lfw

[Web69]Whatisthedifferencebetweenbatchandanepochindeeplearning.URL:https://www.quora.com/What-is-the-difference-between-a-batch-and-an-epoch-in-deep-learning

[Web-70] Python. URL: https://www.python.org/doc/essays/blurb/

[Web-71] What-is-kaggle. URL: https://www.datacamp.com/blog/what-is-kaggle

[Web-72] What is numpy. URL: https://numpy.org/doc/stable/user/whatisnumpy.html

[Web-73] What-is-pandas-in-python. URL: https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/

[Web-74] Python introduction matplotlib. URL: https://www.geeksforgeeks.org/python-introduction-matplotlib/

[Web-75] Introduction-to-seaborn-python. URL: https://www.geeksforgeeks.org/introduction-to-seaborn-python/

[Web76]Learningmodelbuildingscikitlearnpythonmachinelearninglibrary.URL:https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/

[Web77]Tenserflow.URL:https://www.techtarget.com/searchdatamanagement/definition/TensorFlow

[Web78] What-is-keras. URL: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras

ANNEX

IMPLEMENTATION TOOLS

1. INTRODUCTION

In this annex, we started by introducing the implementation tools employed in our study. Subsequently, we delineate the various steps involved in developing our person identification system, which is founded on facial features and employs transfer learning techniques.

2. HARDWARE TOOLS

In our system for person identification, we employed a range of hardware tools to support the utilization of facial features and transfer learning techniques. These tools encompassed high-performance computing devices, such as GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units), which enabled efficient processing and analysis of large volumes of image data. Additionally, we utilized specialized hardware accelerators designed specifically for deep learning tasks, optimizing the performance of our system during training and inference phases. Our setup included a Python 3 environment running on a Google Compute Engine backend equipped with a GPU, with available system resources of 4.26 GB RAM out of 12.67 GB total, and 28.74 GB disk space out of 78.19 GB total. These hardware resources played a crucial role in enhancing the speed, accuracy, and scalability of our person identification system.

3. SOFTWARE TOOLS

3.1. PYTHON PROGRAMMING LANGUAGE

Python is an interpreted, object-oriented, high-level programming language with dynamic sementics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [Web-70].

3.2. COLAB

Google Colaboratory, or Colab, is a cloud service, offered by Google (free or paid), which is a hosted version of Jupyter Notebook that enables you to write and execute Python code through your browser, that requires no setup to use and provides free access to computing resources and easy to share, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

3.3. KAGGLE

Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientists to solve data science challenges. The aim of this online platform (founded in 2010 by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017) is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides [Web-71].

4. USED LIBRARIES

4.1 NUMPY

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the *n-array* object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences [Web-72].

4.1. PANDAS

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions [web2.activestate]. Pandas simplifies numerous time-consuming and repetitive tasks related to data manipulation. It streamlines processes such as data cleansing, filling missing values, normalization, merging datasets, and performing joins. Additionally, it facilitates data visualization, statistical analysis, and inspection, empowering users to gain valuable insights from their datasets. Furthermore, Pandas provides efficient methods for loading and saving data, enhancing workflow efficiency and productivity [Web-73].

4.3 MATPLOTLIB

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand. It was originally developed by John D. Hunter in 2003 and is now maintained by a large community of developers [web4.geeksforgeek]. Matplotlib offers a versatile plotting library capable of generating a diverse range of visualizations, from basic line and scatter plots to complex histograms and pie charts. Its extensive customization options empower users to tailor every aspect of their plots to suit their needs, including colors, markers, labels, and annotations. Integrated seamlessly with NumPy, Matplotlib enables straightforward plotting of data arrays. It excels in producing publication-quality plots with precise control over aesthetics. Moreover, its extensibility is evident through its vast ecosystem of add-on toolkits and extensions, including Seaborn and Pandas plotting functions [Web-74].

4.4. SEABORN

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset. Seaborn offers a variety of specialized plot types to explore relationships and distributions within datasets. Relational plots focus on visualizing the relationship between two variables, while categorical plots are tailored for categorical variables, providing insight into their visualization. Distribution plots allow for examination of both univariate and bivariate distributions. Regression plots provide visual guidance to highlight patterns in the dataset during exploratory data analyses [Web-75].

4.5. SKLEARN

Scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface. It is an open-source machine-learning library that provides a plethora of tools for various machine-learning tasks such as Classification, Regression, Clustering, and many more [web4]. The Scikit-learn library offers several advantages for machine learning tasks, With a wide range of tuning parameters available, Scikit-learn strikes a balance between flexibility and ease of use by providing sensible defaults, streamlining the model development process. The library is renowned for its exceptional documentation, making it accessible for users at all levels of expertise [Web-76].

4.6. TENSERFLOW

TensorFlow is an open source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers. The TensorFlow software handles data sets that are arrayed as computational nodes in graph form. The edges that connect the nodes in a graph can represent multidimensional vectors or matrices, creating what are known as tensors. Because TensorFlow programs use a data flow architecture that works with generalized intermediate results of the computations, they are especially open to very large-scale parallel processing applications, with neural networks being a common example [web5]. TensorFlow's framework comprising high-level and low-level APIs. Google advises using high-level APIs for streamlined development but acknowledges the value of low-level TensorFlow Core APIs for experimentation and debugging. Mastery of these low-level APIs offers insight into the inner workings of machine learning technology, according to Google [Web-77].

The versatility of TensorFlow applications, can operate on CPUs, GPUs, and Google's custom TPUs. Initially developed in 2016, TPUs were integrated with TensorFlow to enhance performance in various Google services like RankBrain and Street View mapping.

4.7 KERAS

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports

multiple backend neural network computation. Keras is relatively easy to learn and work with because it provides a python frontend with a high level of abstraction while having the option of multiple back-ends for computation purposes. This makes Keras slower than other deep learning frameworks, but extremely beginner-friendly [web6]. Keras has capability to seamlessly switch between different backend frameworks, including TensorFlow, Theano, PlaidML, MXNet, and CNTK. TensorFlow has adopted Keras as its official high-level API, embedded within TensorFlow for rapid deep learning operations with built-in neural network modules. However, developers also have the option to utilize TensorFlow's Core API for customized computations, offering flexibility and control to implement ideas efficiently. Keras streamlines implementation with consistent, straightforward APIs, reducing errors and speeding up prototyping. Leveraging TensorFlow, Keras offers fast performance and seamless customization, supporting various deployment options. Despite its high-level abstraction, Keras maintains speed and flexibility, making it ideal for rapid development. Its extensive documentation and strong community support make it a preferred choice for both research and commercial applications, utilized by companies like Netflix, Uber, and others. Additionally, Keras boasts features such as compatibility with CPU and GPU, support for diverse neural network models, and modularity for expressive and flexible research [Web-78].

5. IMPLEMENTATION STEPS

Here are the steps to implement our facial recognition system.

5.1. IMPORTING LIBRARIES

To implement our facial recognition system, we first need to import several essential libraries. These include libraries for numerical computations, such as NumPy, and for handling image data. Figure 1 shows a snippet of code for importing the necessary libraries.

```
from matplotlib import pyplot as plt
from keras.models import Model, Sequential
from sklearn.model_selection import train_test_split
import tensorflow as tf
import pandas as pd
import numpy as np
import gradio as gr
import seaborn as sns
```

Figure 1 : Importing libraries.

5.2. THE DATA SPLITTING

We divided LFW database into three subsets: one for training, one for validation, and one for testing training, validation, testing. The division of these data was carried out using the code presented below.

```
X1_train, X1_temp, Y1_train, Y1_temp = train_test_split(X, encoded_Y, test_size=0.3, random_state=30)
X1_dev, X1_test, Y1_dev, Y1_test = train_test_split(X1_temp, Y1_temp, test_size=0.5, random_state=30)
```

Figure 2 : The data splitting.

5.3. DATA PREPROCESSING

We preprocessed the data to ensure it is suitable for training our facial recognition model. The code presented below illustrates these preprocessing steps, ensuring that all images are uniformly prepared for our model, enhancing the model's performance and robustness.

```
X1_train = np.array([cv2.resize(img, (224, 224)) for img in X1_train])
X1_test = np.array([cv2.resize(img, (224, 224)) for img in X1_test])
X1_dev = np.array([cv2.resize(img, (224, 224)) for img in X1_dev])

X1_train = X1_train.astype('float32') / 255.0
X1_dev = X1_dev.astype('float32') / 255.0
X1_test = X1_test.astype('float32') / 255.0
```

Figure 3 : Data preprocessing instructions.

5.4. LOADING EFFICIENTNET MODEL

To create a pre-trained EfficientNet model ready for use in facial recognition tasks, we can employ a similar approach as described in the instructions below.

```
eff_net = tf.keras.Sequential([
    hub.KerasLayer(('/kaggle/input/efficientnet-v2/tensorflow2/imagenet1k-b0-classification/2'),
                   trainable=True, input_shape=(224, 224, 3)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(12, activation='softmax')
])
```

Figure 4 : Loading EfficientNet model instructions.

5.6. CALLBACKS

We used callbacks-list, Manages a sequence of callback functions, such as Early Stopping and Reduce Learning Rate, during the training of machine learning models to enhance performance and prevent overfitting.

```
callbacks_list = [
    EarlyStopping(monitor='val_accuracy', patience=4, verbose=1),
    ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=4, min_lr=1e-5, verbose=1)
]
```

Figure 5 : Callbacks instructions.

5.7. TRAINING FUNCTION

This line of code executes the training of our model.

```
history = eff_net.fit(X1_train, Y1_train, validation_data=(X1_dev, Y1_dev),
                      epochs=15, batch_size=batch_size, callbacks=callbacks_list)
```

Figure 6 : Training function instruction.

5.8. TESTING FUNCTION

In the instruction bellow, the "test loss" defines the error between predictions and the actual test dataset, whereas "test accuracy" measures the proportion of correctly classified instances out of the total instances in the test dataset.

```
test_loss, test_accuracy = eff_net.evaluate(X1_test, Y1_test)
```

Figure 7 : Testing function instruction.

5.9. INTERFACE

Our interface for facial recognition is a stand-alone application with a minimalistic design. It features a single button for uploading an image file. Once an image is provided, the interface displays the detected faces with bounding boxes and labels indicating the recognized individuals. Additionally, it could show the probability scores associated with each detection, as shown in the figure below.

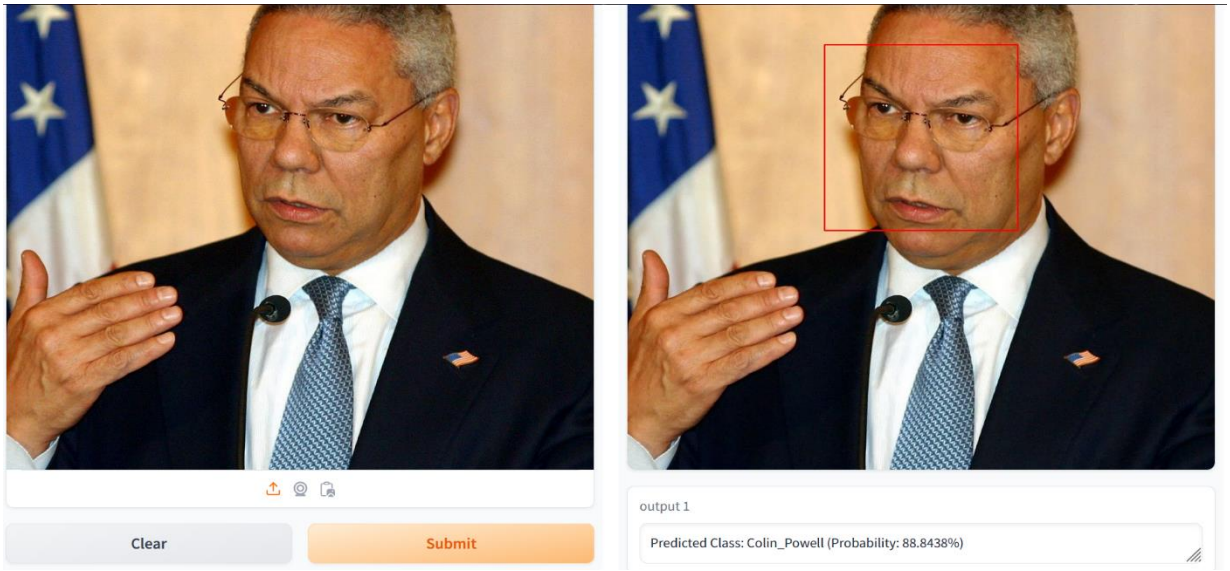

Figure 8 : Person identification based on facial features interface.

6. CONCLUSION

In this annex, we initiated by introducing the implementation tools utilized in our study. Following this, we outlined the sequential steps involved in crafting our person identification system, and we clarified the code snippets from our Python implementation, elucidating key segments relevant to the development and functioning of our person identification system, which is centered on facial features and utilizes a transfer learning approach.