



HAL
open science

Biometrics for face skin analysis using machine learning based approaches

Rola El Saleh

► **To cite this version:**

Rola El Saleh. Biometrics for face skin analysis using machine learning based approaches. Machine Learning [stat.ML]. Université Paris-Est Créteil Val-de-Marne - Paris 12, 2021. English. NNT : 2021PA120033 . tel-04072085

HAL Id: tel-04072085

<https://theses.hal.science/tel-04072085v1>

Submitted on 17 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ PARIS-EST CRÉTEIL

**MATHÉMATIQUES ET SCIENCES ET TECHNOLOGIES DE
L'INFORMATION ET DE LA COMMUNICATION**

Thèse de doctorat

Signal, Image, Automatique

Rola EL SALEH

**BIOMÉTRIE POUR L'ANALYSE DE LA PEAU DU VISAGE:
ANALYSE PAR APPROCHES "MACHINE LEARNING**

**BIOMETRICS FOR FACE SKIN ANALYSIS USING MACHINE
LEARNING BASED APPROACHES**

Thèse dirigée par Prof. Amine NAIT-ALI

Soutenue le 16 décembre 2021

Dr. HDR Estelle CHERRIER	ENSICAEN	Rapporteur
Prof. Mouloud ADEL	Aix-Marseille Université	Rapporteur
Prof. Christine FERNANDEZ-MALOIGNE	Université de Poitiers	Examinatrice
Dr. Régis FOURNIER	UPEC	Examineur
Prof. Amine NAIT-ALI	UPEC	Directeur

Biometrics for face skin analysis using machine learning based approaches

Abstract

The emergence of artificial intelligence (AI), access to large databases, and the availability of supercomputers have undoubtedly revolutionized the medical treatment in various fields. In fact, taking into consideration the development of “Machine-learning” (ML) algorithms and in particular “Deep-learning” (DL), has greatly benefited the biomedical field and added to its efficiency. In the context of dermatology, much research has been carried out to automatically analyze images of the skin in order to predict diseases and to follow their evolution over time. This thesis continues this present research and proposes a computer-aided diagnostic system based on “Deep-learning” (DL) approaches that analyzes facial images and identifies potential facial diseases using only facial phenotypes, without region of interest extraction. This medical, facial biometrics is based on the use of pre-trained convolutional neural networks, VGG-16, EfficientNet b0 and Inception v3, which are fine-tuned to create new models adapted to classify facial skin images into eight distinct pathologies namely: acne, actinic keratosis, angioedema, blepharitis, eczema, melasma, rosacea and vitiligo. Thus, a transfer learning method has been used. Specifically, the original architectures of the three models have been changed by adding new layers to the top. In this study, the proposed algorithms are trained and validated on a database that we have created for this purpose. The models are tested and evaluated considering different acquisition conditions (facial pose, brightness, image resolution, etc...). Very promising results have thus been obtained and will be discussed in detail in this dissertation.

Keywords: Skin disease classification, Face analysis, Deep learning, Convolutional neural network, VGG-16, EfficientNet b0, Inception V3, Medical biometrics, Dermatology.

Résumé

L'émergence de l'intelligence artificielle (IA), l'accès aux grandes bases de données, et la disponibilité de super-calculateurs ont incontestablement révolutionné les différents domaines. En particulier, en considérant le développement d'algorithmes d'Apprentissage Automatique (AA) et notamment de l'Apprentissage Profond (AP), le domaine biomédical en a largement bénéficié. Dans le contexte de la dermatologie, de nombreuses recherches ont été menées pour analyser automatiquement les images de la peau afin de prédire les maladies et de suivre leur évolution au cours du temps. Cette thèse propose un système de diagnostic assisté par ordinateur basé sur des approches d'AP qui analyse les images du visage et identifie les maladies potentielles du visage utilisant uniquement des phénotypes faciaux, sans extraction de région d'intérêt. Cette biométrie médicale, faciale est fondée sur l'utilisation de réseaux de neurones convolutifs pré-entraînés, VGG-16, EfficientNet b0 et Inception v3, qui sont affinés pour créer de nouveaux modèles adaptés pour classer les images de la peau du visage en huit maladies distinctes : acné, kératose actinique, angioedème, blépharite, eczéma, mélasma, rosacée et vitiligo. Ainsi, une méthode d'apprentissage par transfert est utilisée. En effet, les architectures originales des trois modèles sont modifiées en ajoutant de nouvelles couches au sommet. Les algorithmes proposés sont entraînés et validés sur une base de données que nous avons créé à cet effet. Les modèles sont testés et évalués en considérant des conditions d'acquisitions différentes (pose du visage, luminosité, résolution d'image, etc). Des résultats très prometteurs sont ainsi obtenus.

Keywords: Classification des maladies de la peau, Analyse du visage, Apprentissage profond, Réseau de neurones convolutifs, VGG-16, EfficientNet b0, Inception V3, Biométrie médicale, Dermatologie.

Résumé substantiel

L'intelligence artificielle (IA), l'apprentissage automatique et l'apprentissage profond sont des termes interconnectés et très répandus. L'intelligence artificielle est la capacité des machines à effectuer des tâches comme les êtres humains. L'apprentissage automatique est un sous-domaine de l'intelligence artificielle qui automatise l'apprentissage des ordinateurs à partir de données et améliore leurs performances grâce à l'expérience et en considérant la moindre intervention humaine. Par ailleurs, l'apprentissage profond est un sous-domaine de l'apprentissage automatique, principalement basé sur des réseaux de neurones artificiels qui apprennent à partir d'énorme quantité de données et imitant ainsi le fonctionnement du cerveau humain en termes de traitement d'informations. L'intérêt est évidemment de prendre des décisions ou reconnaître des modèles. Les applications de l'apprentissage profond ont révolutionné la technologie utilisée dans les différents domaines tels que : les affaires, l'éducation, la fabrication, la banque, l'armée, la santé et bien d'autres. En particulier, d'énormes investissements ont été entrepris dans le domaine médical afin d'apporter des solutions adaptées à des techniques de soin. Ainsi, l'intégration de l'apprentissage profond dans les algorithmes d'aide au diagnostic est devenue une pratique courante. Les différentes modalités sont ainsi considérées pour le traitement de bio-signaux (e.g. EEG, ECG, EMG, etc), ainsi que pour le traitement d'images médicales (e.g. échographiques, RMN, etc). Pour ces dernières, la caractérisation, la segmentation et la classification automatique ont permis d'améliorer considérablement les performances des logiciels. En particulier, la classification des images est l'une des tâches les plus importantes de la vision par ordinateur. Elle consiste à analyser des images et à les classer dans une catégorie parmi un ensemble prédéfinies. Contrairement aux techniques traditionnelles de classification d'images, les algorithmes d'apprentissage profond combinent les tâches d'extraction de caractéristiques et de classification. Dans ce contexte, de nombreux algorithmes ont été proposés. En particulier, les réseaux neuronaux convolutifs (CNN) sont des algo-

rithmes d'apprentissage profond les plus populaires utilisés pour la classification des images. Il s'agit de réseaux inspirés de la manière dont le cerveau humain reconnaît l'information. Les CNN éliminent l'extraction manuelle de caractéristiques de bas et moyen niveaux. Dans la littérature, plusieurs modèles CNN ont été proposés permettant d'atteindre une grande précision de classification et dans certains cas, peuvent dépasser même la reconnaissance humaine.

Dans le contexte de notre sujet de recherche, l'apprentissage profond a fait l'objet d'exploration dans des applications liées à la dermatologie, en allant du diagnostic au traitement. Ainsi, de nombreuses études ont été menées pour développer des approches automatisées d'analyse d'images de la peau humaine afin de prédire et de classer les pathologies, d'évaluer et de suivre leur évolution dans le temps.

Dans ce contexte, l'objectif de cette recherche est de développer un système de diagnostic assisté, fondé sur l'IA afin d'identifier les lésions cutanées du visage en utilisant différentes approches d'apprentissage profond, et en particulier les modèles CNN. Huit pathologies cutanées du visage sont traitées dans cette étude. Il s'agit, en effet, de pathologies les plus courantes affectant les personnes à tout âge et tout genre. En particulier, nous nous sommes intéressés aux pathologies suivantes : l'acné, la kératose actinique, l'angioedème, la blépharite, l'eczéma, le mélasme, la rosacée et le vitiligo. Le protocole considéré utilise des techniques de pré-traitement issues de la biométrie faciale. L'analyse est effectuée indépendamment des conditions d'acquisition telles que la résolution de l'image, l'éclairage, l'occultation ainsi que la pose du visage. Nous avons donc construit un nouveau modèle basé sur l'apprentissage par transfert en utilisant le VGG-16, adapté à l'identification des maladies de la peau du visage, que nous avons appelé "Facial Skin Diseases Network" (FSDNet). Notre système peut également prédire une classe de "Peau saine" et une classe de "visage-absent", notamment si l'image présentée ne représente pas un visage. L'efficacité de FSDNet est évaluée en comparant ses performances avec celles obtenues par d'autres architectures CNN telles que le VGG-16 pré-entraîné,

et l'apprentissage transfert utilisant Inception V3, et l'apprentissage par transfert utilisant EfficientNet b0. L'apprentissage par transfert est appliqué avec la même architecture proposée par FSDNet afin de rendre la comparaison significative. Les principales contributions de notre étude se résument comme suit : (1) identification des pathologies sans extraction locale de la région d'intérêt relative à la texture de la peau (modèle entraîné indépendamment de la résolution de l'image, de la pose et de l'illumination, etc). (2) Un plus grand nombre de pathologies est identifié par rapport aux techniques les plus récentes, publiées jusqu'ici.

En raison de l'absence d'un ensemble de données publiques standard, nous avons créé une base de données contenant 20 000 images étiquetées contenant : des visages présentant huit maladies cutanées, des visages à la peau normale, et enfin des images présentant des objets et des scènes variantes (e.g. images de nourriture, de tasses, d'arbres, etc.) collectées de différentes sources et utilisées pour entraîner et valider nos réseaux.

Ce rapport est divisé en quatre chapitres. Le premier chapitre se concentre sur l'état de l'art des méthodes d'identification des maladies cutanées du visage. Nous décrirons également les différentes maladies qui seront classées par l'approche que l'on a développée. Nous introduisons ainsi le fondement de notre approche d'identification basée sur l'apprentissage profond ayant pour objectif de classer plusieurs pathologies cutanées du visage (i.e. acné, kératose actinique, angioedème, blépharite, eczéma, mélasma, rosacée et vitiligo).

Dans le deuxième chapitre, nous passons en revue, le principe de l'apprentissage profond. Ainsi, nous présentons le concept de " hiérarchie des caractéristiques " qui fait de l'apprentissage profond un outil performant et adapté aux applications de vision par ordinateur. Nous présentons ensuite les différentes architectures des réseaux de neurones profonds, en mettant en évidence les réseaux de neurones convolutifs (CNN), en explicitant les blocs et couches constituant le réseau, ses différentes architectures, ainsi que son mode d'apprentissage. Enfin, nous procédons à l'utilisation

des CNN pour la classification d'images.

Dans le troisième chapitre, nous abordons l'architecture des différents réseaux VGG, Inception v3, et EfficientNet b0, que nous avons utilisés comme base pour construire nos modèles adaptés à la classification des pathologies cutanées du visage. Le VGG est considéré comme une architecture générale et comme une référence pour de nombreuses tâches et ensembles de données en dehors d'ImageNet, la base de données sur laquelle le réseau est entraîné initialement. Inception v3 est la troisième édition du CNN Inception de Google. Alors que la plupart des CNN populaires travaillent sur l'augmentation de la profondeur du réseau, les modèles Inception utilisent d'autres techniques pour améliorer les performances du réseau en termes de vitesse et de précision. De nombreuses versions ont été créées, chaque version étant une amélioration de la précédente. Les modèles EfficientNet, récemment développés par Google, présentent un nouveau concept de mise à l'échelle des réseaux pour améliorer les performances, appelées "Compound Scaling". Les modèles CNN précédents suivent la méthode classique de mise à l'échelle arbitraire d'une seule dimension et d'ajout de couches supplémentaires. Les modèles EfficientNet mettent uniformément à l'échelle toutes les dimensions (i.e. profondeur, largeur et résolution) en utilisant un coefficient composé. Ils améliorent la précision et l'efficacité du modèle. Nous expliquerons également comment nous pouvons appliquer ces réseaux pré-entraînés sur ImageNet pour classer d'autres images de catégories différentes de celles d'ImageNet.

Dans le quatrième chapitre, nous allons élaborer une méthode complète de bout en bout qui permet d'identifier les pathologies cutanées du visage dans les images RVB en utilisant l'apprentissage profond et les techniques de vision par ordinateur. Afin d'accomplir cette tâche, nous utiliserons l'apprentissage par transfert avec les trois modèles pré-entraînés VGG-16, Inception v3 et EfficientNet b0 pour créer nos modèles adaptés à ce contexte. Dans la suite du chapitre, nous détaillerons l'approche que nous proposons tout en mettant en évidence les résultats que nous avons obtenus

pour chaque modèle en termes de précision, d'exactitude et d'autres métriques. Nous démontrons ainsi que l'apprentissage par transfert peut améliorer considérablement la précision de l'identification des lésions cutanées du visage.

Enfin, une conclusion générale et des perspectives seront présentées à la fin de ce rapport.

Acknowledgement

I would like to express my deep and sincere gratitude to my research supervisor, Professor Amine NAIT-ALI, for giving me the opportunity to do research and providing invaluable guidance throughout this research, for his patience, motivation, and immense knowledge. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

I would also like to give special thanks to Dr Sambit BAKSHI for his contribution in this thesis.

My sincere thanks go to all members of the jury for accepting to participate in the defense of this dissertation.

My appreciations also go to my family for their love, prayers, caring, sacrifices and support all through my studies. I am very thankful to my husband, my daughter, and my son for their love, continuing support, tremendous understanding, and encouragement.

List of publications

International journal

Rola EL SALEH, Samer CHANTAF and Amine NAIT-ALI. Identification of facial skin diseases from face phenotypes using FSDNet in uncontrolled environment. Machine Vision and Applications, 2021 (accepted for publication).

International conference

1. Rola EL SALEH, Sambit BAKHSHI and Amine NAIT-ALI. Deep convolutional neural network for face skin diseases identification. 2019 Fifth International Conference on Advances in Biomedical Engineering (ICABME), 2019, pp. 1-4, doi: 10.1109/ICABME47164.2019.8940336.

2. Hazem ZEIN, Samer CHANTAF, Rola EL SALEH and Amine NAIT-ALI. Generative Adversarial Networks Based Approach for Artificial Face Dataset Generation in Acne Disease Cases. 2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART).

Book chapter

Rola EL SALEH, Hazem ZEIN, Samer CHANTAF, Amine NAIT-ALI. Artificial Intelligence in Dermatology: A Case Study for Facial Skin Diseases. In: Advances in Artificial Intelligence, Computation, and Data Science. Springer International Publishing, 2021, pp. 163-178.

Contents

List of figures	13
List of tables	18
General introduction	20
1 Facial skin diseases: generalities	24
1.1 Introduction	24
1.2 State of the art facial skin diseases identification methods	25
1.3 Facial skin diseases	33
1.3.1 Acne	33
1.3.2 Actinic keratosis	35
1.3.3 Angioedema	36
1.3.4 Blepharitis	36
1.3.5 Eczema	37
1.3.6 Melasma	38
1.3.7 Rosacea	39
1.3.8 Vitiligo	40
1.4 Conclusion	41
2 Deep learning based skin diseases classification	42
2.1 Introduction	42
2.2 Artificial intelligence	44

2.3	Machine learning	44
2.4	Deep learning	45
2.4.1	Deep learning versus machine learning	45
2.4.2	Comparison between machine learning and deep learning . . .	47
2.4.3	How deep learning works?	47
2.5	Neural networks basics	48
2.5.1	How artificial neural networks work?	49
2.5.2	Activation function	50
2.6	Deep neural networks	53
2.6.1	Architecture of deep neural networks	55
2.7	Convolutional neural networks	56
2.7.1	CNN blocs	57
2.7.2	CNN training	66
2.8	CNN and Image Classification	70
2.8.1	What is image classification?	70
2.8.2	Image classification based on Deep learning	70
2.8.3	Convolutional Neural Networks for image classification	71
2.9	Conclusion	72
3	Convolutional neural networks architectures	74
3.1	Introduction	74
3.2	Visual Geometry Group (VGG)	75
3.2.1	VGG-16	77
3.3	Inception-v3	79
3.3.1	Factorized convolutions	80
3.3.2	Auxiliary Classifier	81
3.3.3	Efficient grid size reduction	82
3.4	EfficientNet	84
3.4.1	EfficientNet architecture	85

3.4.2	Inverted Residual Block	87
3.4.3	Squeeze and Excitation Block	88
3.5	Transfer learning with CNN	89
3.5.1	What is a pre-trained model?	89
3.5.2	Transfer learning	90
3.5.3	Transfer learning scenarios	92
3.6	Conclusion	94
4	Facial Skin Disease Identification AI Based Approach	96
4.1	Introduction	96
4.2	Facial skin diseases identification method	96
4.3	Facial skin disease dataset	97
4.4	Facial skin diseases identification using VGG-16	100
4.4.1	Classification using pre-trained VGG-16	101
4.4.2	Classification using transfer learning with VGG-16	109
4.4.3	Test	118
4.4.4	Performance evaluation of FSDNet	121
4.5	Facial skin disease identification using EfficientNet b0	124
4.6	Facial skin disease identification using transfer learning with Inception V3	134
4.7	Comparative study	143
4.8	Conclusion	144
	General conclusion and perspectives	146
	Bibliography	150

List of Figures

1.1	Different types of acne	34
1.2	Examples of actinic keratosis on the face	35
1.3	Angioedema in face	36
1.4	Blepharitis in face	37
1.5	Eczema images	38
1.6	Examples of Melasma	39
1.7	Rosacea images	40
1.8	Vitiligo Images	40
2.1	AI vs Machine learning vs Deep learning	43
2.2	Machine learning and Deep learning technique	46
2.3	ML vs DL algorithms performance	46
2.4	Comparison between ML and DL	47
2.5	Hierarchical feature learning	48

2.6	Artificial neural network architecture	49
2.7	Structure of ANN	50
2.8	Step function.	51
2.9	Sigmoid function.	52
2.10	Hyperbolic tangent	52
2.11	Rectified Linear Unit (ReLU)	53
2.12	Deep neural network architecture	54
2.13	Feature hierarchy learned by a deep neural network on faces	55
2.14	Example of a filter applied to an input to produce a feature map	57
2.15	General block diagram of a CNN	58
2.16	Basic CNN architecture	58
2.17	CNN convolution operation	59
2.18	Example of stride	60
2.19	Example of zero padding	61
2.20	Example of max pooling	62
2.21	Example of fully connected layer	63
2.22	Example of dropout	64
2.23	Training process of CNN	66
2.24	Gradient descent algorithm	68

3.1	VGG architecture	76
3.2	VGG16 architecture	78
3.3	Factorized convolutions	80
3.4	Asymmetric Convolutions	81
3.5	Auxiliary classifier in inception-v3	82
3.6	Efficient grid size reduction	83
3.7	Architecture of Inception-v3	83
3.8	Model scaling	85
3.9	The Swish activation function	87
3.10	Residual and inverted residual block	87
3.11	SE block structure	88
3.12	Benefits of transfer learning for deep learning with CNN	91
3.13	Transfer learning scenarios	92
3.14	Transfer learning methods	93
4.1	Bloc diagram of facial skin diseases identification	97
4.2	Sample images from our database	99
4.3	Images from our dataset	99
4.4	Pre-trained VGG-16 architecture	102
4.5	Facial skin diseases architecture	103

4.6	Plots of training and validation	106
4.7	Evaluation metrics	107
4.8	Confusion matrices	108
4.9	FSDNet architecture	111
4.10	Training and validation accuracies and loss for 30 epochs	112
4.11	Training and validation loss in different batch sizes	113
4.12	Plots of training and validation curves	115
4.13	Classification report	116
4.14	Confusion matrices of FSDNet	117
4.15	Study of the effect of brightness on the performance in class Eczema .	122
4.16	Study of the effect of brightness on the performance in class Acne . .	122
4.17	Study of the effect of brightness on the performance in class Normal .	122
4.18	Face pose effect in class Vitiligo	123
4.19	Face pose effect in class Angioedema	123
4.20	Face pose effect in class Melasma	124
4.21	Fine-tuned EfficientNet b0 architecture	125
4.22	Curves of training and validation	126
4.23	Training and validation curves of fine-tuned EfficientNet b0	127
4.24	Classification report of fine-tuned EfficientNet b0	127

4.25 Confusion matrices of fine-tuned EfficientNet b0 128

4.26 A study of the effect of brightness in 3 classes 132

4.27 Face pose effect on the performance in 3 classes 133

4.28 Fine-tuned Inception V3 134

4.29 Curves of training and validation 135

4.30 Training and validation curves of fine-tuned Inception V3 136

4.31 Classification report of fine-tuned Inception V3 136

4.32 Confusion matrix of fine-tuned Inception V3 137

4.33 Evaluation of the effect of brightness (Inception V3) in 3 classes . . . 141

4.34 Face pose effect on the performance of fine-tuned Inception V3 142

List of Tables

3.1	VGG configurations	77
3.2	EfficientNet B0 baseline network architecture	86
4.1	Training and Validation times	103
4.2	Training accuracy and loss	104
4.3	Validation accuracy and loss	104
4.4	Training and validation times	113
4.5	Training and validation accuracies and loss	113
4.6	Number of correct and incorrect predicted labels	114
4.7	Training and validation accuracy loss of fine-tuned EfficientNet	126
4.8	Summary of results	143
4.9	Summary of test images classification	143

General introduction

Artificial intelligence, machine learning, and deep learning are the buzzword of this era. These popular and trendy terms are interconnected. Artificial intelligence is the ability of machines to perform tasks like humans. Machine learning is a subfield of artificial intelligence that automates the learning of computers from data and improves their performance through experience while considering the slightest human intervention. On the other hand, deep learning is a subfield of machine learning, mainly based on artificial neural networks which learn from huge amounts of data and thus mimics the functioning of the human brain in terms of processing information. The interest is obviously to make decisions or recognize models. Deep learning applications have revolutionized the technology used in different fields such as: business, education, manufacturing, banking, military, healthcare and many more.

In particular, huge investments have been made in the medical field in order to provide solutions adapted to care techniques. Thus, the integration of deep learning into diagnostic aid algorithms has become a common practice. The different modalities are thus considered for the processing of bio-signals (e.g. EEG, ECG, EMG, etc.), as well as for the processing of medical images (e.g. ultrasound, NMR, etc.). For the latter, characterization, segmentation and automatic classification have significantly improved software performance. In particular, the classification of images is one of the most important tasks of computer vision. It consists of analyzing images and classifying them in a category among a predefined set. Unlike

traditional images classification techniques, deep learning algorithms combine the tasks of feature extraction and classification. In this context, many algorithms have been proposed. In particular, convolutional neural networks (CNNs) are the most popular deep learning algorithms used for image classification. These are networks inspired by the way the human brain recognizes information. CNNs eliminate the manual extraction of low and mid-level features. In the literature, several CNN models have been proposed to achieve high classification accuracy and in some cases can even exceed human recognition.

In the context of our research topic, deep learning has been explored in applications related to dermatology, from diagnosis to treatment. Thus, numerous studies have been carried out to develop automated approaches for analyzing images of human skin in order to predict and classify pathologies, and to evaluate and monitor their evolution over time.

Within this context, the purpose of this research is to develop an AI based aided diagnostic system to identify facial skin diseases using different deep learning approaches, and especially CNN models. Eight skin pathologies of the face are handled in this study. These are, in fact, the most common pathologies affecting people of all ages and genders. In particular, we were interested in the following pathologies: acne, actinic keratosis, angioedema, blepharitis, eczema, melasma, rosacea and vitiligo. The protocol considered uses pre-treatment techniques derived from facial biometrics. Analysis is performed regardless of acquisition conditions such as image resolution, lighting, shadowing, and face pose. So we built a new model based on transfer learning using VGG-16, suitable for identifying facial skin diseases, which we called "Facial Skin Diseases Network" (FSDNet). Our system can also predict a class of "normal skin" and a class of "no-face", especially if the image presented does not represent a face. The efficiency of FSDNet is evaluated by comparing its performance with that obtained by other CNN architectures such as the pre-trained VGG-16, and transfer learning using Inception V3, and transfer learning using Ef-

efficientNet b0. Transfer learning is applied with the same architecture offered by FSDNet in order to make the comparison meaningful. The main contributions of our study are summarized as follows: (1) identification of pathologies without extraction of the region of interest relating to the texture of the skin (model trained independently of image resolution, pose and illumination, etc.). (2) A greater number of pathologies are identified compared to the most recent techniques published so far.

Due to the absence of any standard public dataset for the same, we established a database that contains 20000 labelled images belonging to the eight facial skin diseases and faces with normal skin in addition to no face images (images of food, cups, trees,etc..) collected from different sources, which is used to train and validate our networks.

This report is divided into four chapters. The first chapter focuses on the state of the art in methods of identifying skin diseases of the face. We will also describe the different diseases that will be classified by the approach that we have developed. We thus introduce the basis of our identification approach based on deep learning with the objective of classifying several skin pathologies of the face (i.e. acne, actinic keratosis, angioedema, blepharitis, eczema, melasma, rosacea and vitiligo).

In the second chapter, we review the principle of deep learning. Thus, we present the concept of "hierarchy of characteristics" which makes deep learning a powerful tool suitable for computer vision applications. We then present the different architectures of deep neural networks, by highlighting convolutional neural networks (CNN), by explaining the blocks and layers constituting the network, its different architectures, as well as its learning mode. Finally, we proceed to the use of CNNs for image classification.

In the third chapter, we discuss the architecture of the different VGG, Inception v3, and EfficientNet b0 networks, which we used as a basis to build our models adapted to the classification of facial skin pathologies. VGG is considered a general

architecture and a benchmark for many tasks and datasets outside of ImageNet, the database on which the network is initially trained. Inception v3 is the third edition of CNN Inception from Google. While most popular CNNs are working on increasing network depth, Inception models use other techniques to improve network performance in terms of speed and accuracy. Many versions have been created, each version being an improvement on the previous one. The EfficientNet models, recently developed by Google, present a new concept of scaling networks to improve performance, called "Compound Scaling". Previous CNN models follow the classic method of arbitrarily scaling a single dimension and adding additional layers. EfficientNet models uniformly scale all dimensions (i.e. depth, width and resolution) using a compound coefficient. They improve the accuracy and efficiency of the model. We will also explain how we can apply these pre-trained networks on ImageNet to classify other images into categories other than ImageNet.

In the fourth chapter, we will develop a complete end-to-end method that identifies facial skin pathologies in RGB images using deep learning and computer vision techniques. In order to accomplish this task, we will use transfer learning with the three pre-trained models VGG-16, Inception v3 and EfficientNet b0 to create our models suitable for this context. In the rest of the chapter, we will detail the approach we propose while highlighting the results we obtained for each model in terms of precision, accuracy and other metrics. We thus demonstrate that transfer learning can significantly improve the accuracy of identifying facial skin lesions.

Finally, a general conclusion and perspectives will be presented at the end of this report.

Chapter 1

Facial skin diseases: generalities

1.1 Introduction

Skin is the largest organ in the human body, covering the body's surface from head to toes. It protects the body against diseases, UV radiation, chemicals and microbes. It controls our body temperature, prevents loss of water, and transmits sensory stimuli such as heat, cold, and touch. The skin is composed of three layers: the epidermis, the dermis, and the hypodermis [1]. The epidermis is the outer, waterproof layer forming the body's first line defense against bacteria, germs, UV radiation and environmental factors. It provides the skin tone and is composed mainly from various types of cells. The dermis is the middle layer containing connective tissues which produce the proteins collagen and elastin that determine the structure, shape, and elasticity of the skin, hair follicles, sweat and oil glands, and blood vessels. The hypodermis is the deepest layer made up of fats, connective tissues, blood vessels and nerves. Actually, the thickness of the epidermis differs depending on what area of the body it is located. Facial skin is thinner than the skin on other parts of the body. Blood vessels, hair follicles, sweat and oil glands are concentrated in the face to protect and moisturize it and also to heal wounds and

scars [2]. Like skin on the rest of the body, facial skin is affected by inflammatory, bacterial and viral diseases and also cancer. Facial skin diseases affect persons at all ages and from different geographies. They differ in symptoms, severity, and pain, and vary from rashes to severe infections happening due to, allergens, auto-immune system disorder, medications, infections or genetics. They can have physical and psychological impacts on the affected persons. Skin disorders are usually diagnosed by visual inspection. However, such naked-eye observation may result in a wrong diagnosis especially in diseases with similar symptoms, and thus the treatment is prolonged, and therefore the cost of treatment increases. To reduce the problem of missed diagnoses, computer-aided diagnosis systems were proposed to help dermatologists overcome the limits of human knowledge. Besides prevention of missed diagnoses, automated identification of diseases also ensures touchless diagnosis that protects physicians in case of contagious diseases, saves time because they can be used anytime and anywhere, solves the problem of lack of dermatologists in rural regions and even more reduces the cost where people can't bear the high costs of consultations.

In the remainder of this chapter, we will focus on state of the art facial skin diseases. Image processing techniques previously proposed for facial skin diseases detection and classification will be reviewed, as well as our proposed approach. Then, we will describe different diseases that will be classified according to our method. Finally, the chapter will end in a conclusion.

1.2 State of the art facial skin diseases identification methods

In the past few years, facial skin diseases have been worthwhile as their procurement and complexities have increased. With the breakthrough in medical technology, the concept of computer-aided diagnostic systems for facial skin diseases have

evolved. These systems simplify the detection and improves the accuracy of prediction and classification of facial skin diseases and minimizes the missed diagnoses. They are based on artificial intelligence using machine learning. These systems interpret, process and predict or classify. Many machine learning based methods have been suggested related to facial skin diseases classification. They use different techniques and algorithms. These approaches are based on image processing techniques, machine learning and deep learning for feature extraction and classification.

C.Y Chang et al. have proposed an automated method to detect facial spots and acne using support-vector- machine based classifier. Their method consists of detecting the face region from the input images using a skin color detection method, and the facial view (front view or profile) to extract the ROI that includes the eyes, the eyebrows, the mouth, and the nose. Once the ROI is extracted, they apply an image segment method to detect potential defects. The classification of the defects includes three stages. First, they compute 14 texture features from co-occurrence matrix: Contrast, Homogeneity, Mean, Variance, Energy, Entropy, Angular Second Moment, Correlation, Sum Average, Sum Entropy, Difference Average, Difference Variance and Difference Entropy. Then, they keep the most significant features that will be used in the classification. Finally, a support-vector-machine (SVM) classifier is used to classify the defects. It consists of two SVMs. The first SVM classifies the images as defects or normal skin, while the second one classifies the defects into acnes or spots. The system is evaluated using 147 images in 3 views: front, left profile, and right profile. The proposed approach detects the defects and recognizes acnes and spots with an average accuracy of 98.95% and 95.2% respectively [3].

Chantharaphaichit et al. have suggested a blob detection based method to detect acne. It consists of detecting regions that have a circular shape called blobs, corresponding to acne. Then, they extract the features that will be used to detect acne. They also use Bayesian classifier following features extraction to minimize the misclassification. This method can detect acne with an accuracy of 70.6%, but is

affected by many conditions such as the form of acne and lighting [4].

M. Amini et al. have developed a mobile application to detect acne and classify it into papules and pustules using smartphone images. The application requires that the images should be acquired from the front. The entire face should be included in the frame and the eyes should be open. The images are first calibrated, and then the features of the face images are identified to detect the human face in the image using facial recognition algorithm. The face images are then normalized to guarantee that the acne is identified on the same region, even with distance and face position changes toward the camera, and the ROI is extracted to detect acne. Actually, acne causes skin redness with comparison to the skin around it. This change of color is used to detect acne. Once identified, acne lesions are classified into papules or pustules. To assess their approach, 10 real human images, and 60 digital face images where lesions are placed virtually are used. The system could detect acne lesions with an accuracy of 92%, and predicts the type of lesion with an accuracy of 98% [5].

An automatic skin disease prediction method based on image processing and machine learning has been proposed by **L. Bajaj et al.** The proposed approach comprises two phases. In the first phase, color face skin images undergo a series of image processing techniques to detect and extract the affected region such as RGB extraction, grey scale, sharpening filter to improve the accurate details of the grey scale image, median filter to remove all noise from the image, smooth filter, and binary filter which is the basis of extraction of the affected region. The extracted diseased region is then converted to a feature vector that is fed to an artificial neural network which is a machine learning algorithm. The network is evaluated using a dataset of 813 images referring to the 5 diseases to predict (eczema, psoriasis, impetigo, melanoma, and scleroderma), split into training, validation, and test sets with 70:15:15 ratio. The method achieves a prediction accuracy of 90% [6].

X. Shen et al. have presented a CNN based diagnosis method of facial acne vul-

garis. They locate first the ROI using a binary classifier with CNN that detects skin and non-skin areas from the images. Then they use a seven-classifier with CNN to classify the images into one of seven classes: six types of acne, and healthy skin. They use pre-trained VGG16 and also construct their own CNN network as binary and seven classifiers. Two datasets are gathered for this purpose. The first one, for binary classification, includes 3000 skin images and 3000 non-skin images, where 80% of the dataset is used for training, 10% for validation and 10% for test. For the seven-classifier model, they augment the dataset to 6000 images in each of the seven classes: blackhead, white- head, papule, pustule, cyst, nodule, and normal skin. The experiments show that the pre-trained VGG16 performs better in both binary and seven classifiers, achieving average accuracies of 87% and 92% respectively [7].

A Multi-Class Multi-Level classification algorithm has been suggested by **N. Hameed et al.** to improve the accuracy of the classification of multiple diseases. The skin lesion images are classified using two methods: a traditional machine learning based algorithm, and an advanced deep learning based algorithm. The first method based on machine learning consists of many phases: preprocessing, segmentation, feature extraction and finally classification. The preprocessing stage includes image resizing, and artifacts (hair, black frame and circle) removal when capturing images. The segmentation aims to extract the region of interest (ROI). They extracted 36 features referring to color and texture categories to classify the images and store them in a feature vector. The classification is achieved by an artificial neural network composed of three layers: an input layer, a hidden layer, and an output layer. The images are classified using two algorithms: Multi-Class Multi-Level (MCML) algorithm and Multi-Class Single-Level (MCSL) algorithm. In MCSL classification, the feature vector is presented to the ANN that classifies the images into healthy, benign, malignant, and eczema. While in MCML method, the classification is done in many levels. The images are first classified into healthy and unhealthy. Then, the unhealthy images are categorized into melanoma and eczema. Finally, in the last

level, melanoma images are classified as malignant or benign. The second approach is based on deep learning. They use deep convolutional neural network, AlexNet, for MCSL and MCML classifications. The proposed methods were evaluated on 3672 images, with 918 images in each class (healthy, benign, malignant and eczema), divided into two sets: 860 images for training and test and 58 images for comparing MCSL and MCML algorithms. The training/test set is split into 70:30 ratios. In both methods, MCML classification algorithm achieves a better performance with an accuracy of 96.47% using the deep learning technique [8].

M. A. Al-masni et al. have presented a deep learning system for segmentation and classification of skin lesions. The skin lesion boundaries are first segmented from the whole images using full resolution convolutional network (FrCN), and then the segmented lesions are transferred to convolutional neural network classifiers: Inception-v3, ResNet-50, Inception- ResNet-v2, and DenseNet-201, to be classified. The segmentation is performed to help the classifier learn only significant features of different types of skin lesions from the image and thus improve its classification performance. The model is evaluated using three datasets: International Skin Imaging Collaboration (ISIC) 2016, 2017, and 2018 including two (benign and melanoma), three (benign, seborrheic keratosis, and melanoma), and seven (benign, seborrheic keratosis, basal cell carcinoma, actinic keratosis, dermatofibroma, vascular lesion, and melanoma) skin disorders, respectively. All datasets have imbalanced classes and include RGB labelled images (1279, 2750, and 10015 images, respectively) of different sizes. They are split into training and test sets. Inception-v3, ResNet-50, Inception-ResNet-v2, and DenseNet-201 achieve accuracies of 77.04%, 79.95%, 81.79%, and 81.27% for two classes of ISIC 2016, 81.29%, 81.57%, 81.34%, and 73.44% for three classes of ISIC 2017, and 88.05%, 89.28%, 87.74%, and 88.70% for seven classes of ISIC 2018, respectively. The results show that ResNet-50 has the best performance [9].

E. Goceri has worked on an automated deep learning based approach to classify

facial dermatological diseases using color digital images. The method comprises two stages. In the first stage, lesions are detected and extracted with a fully automated updated extension of the Automated Detection of Facial Disorders (ADFD) technique, named F-ADFD. This technique consists of several steps. First, the type of the noise is determined by computing the statistical features of noise signals: kurtosis and skewness. That shows that the noise in the images is Gaussian. This noise is reduced using pixel based linear filtering, specifically Wiener filtering. Then the images undergo a contrast enhancement. The images are converted from RGB space to $L_a \times b^*$, and then the double-type data is scaled to the range $[0 \ 1]$ by dividing the values in the luminosity channel by 100 since they vary between 0 and 100. The pixels in the 1% area from the top and bottom part are saturated to apply contrast enhancement only to the values in the luminosity channel by retaining the values in $a \times$ and $b \times$ channels identical. At last, the images are converted from $L_a \times b^*$ to RGB. To set automatically the active contours, an image with the variance of the values in red and green color is generated. After the intensity normalization step, the regions with lesions are of higher intensity than the other pixels. With a k -means based clustering, a new binary image is produced and used to set the active contour. In the second stage, a deep learning based classification of the segmented lesions is performed using a pre-trained DenseNet 201. The images are classified into five facial skin diseases classes: acne vulgaris, psoriasis, hemangioma, seborrheic dermatitis and rosacea. The network is trained with 80 images from each class and tested with 21 images from each class. The lesions are classified with an accuracy of 95.4% [10]. Researchers have also developed some techniques based on deep convolutional neural networks for classification purposes, and not requiring features extraction.

Z. WU et al. have used five pertained CNN models: ResNet-50, Inception-v3, DenseNet121, Xception, and Inception-ResNet-v2 to classify six facial skin diseases: seborrheic keratosis, actinic keratosis, rosacea, lupus erythematosus, basal cell car-

cinoma, and squamous cell carcinoma. They create a dataset containing 2656 face images referring to the six diseases to identify. For all models, the images are first randomly reversed and cropped and then presented to the networks. To remedy the problem of classes imbalance, various weights in the cost function for different lesions are used. The networks are given images of size 300 x 300 as inputs and trained with facial images. The results show that the Inception-ResNet-v2 has the best performance with an average precision of 63.7% and an average recall of 67.2%. Then, the different models are evaluated with a new dataset used including images from different body parts of the same diseases mentioned previously. Inception-ResNet-v2 also procures the best performance with an average precision of 70.8% and an average recall of 77% [11].

H. Wu et al. have developed an artificial intelligence dermatology diagnosis assistant (AIDDA) based on CNN model, EfficientNet B4, that classifies skin images into 3 classes: psoriasis, eczema and atopic dermatitis, and healthy skin. They have built a dataset including 4740 skin diseases images grouped in sets of images belonging to the same disease but from different angles, or various images of identical skin disorders for the same person. Before being stored in the dataset, the images undergo a data cleaning where duplicate case and blur images are removed from the dataset, and then undergo data structuring and standardizing. After cleaning, images are managed as cases instead of individual images. These images are divided into training and validation sets. Five-fold cross-validation is used to evaluate the algorithm. The validation set is randomly picked and contains 20% of the cases. The proposed system achieves an accuracy of 95.8% [12].

Shanti et al. have proposed an automatic skin disease classification approach using Convolutional Neural Network. They have used AlexNet architecture. The method can detect four skin diseases: acne, keratosis, eczema herpeticum and urticaria. The network is trained with 105 images and tested with 69 images of different skin tones, location of the disease and different acquisition systems, from DermNet dataset refer-

ring to the diagnosed diseases. The model achieves an accuracy of 98.6% to 99.04% [13].

A deep learning based classification method has been proposed by **I. Iqbal et al.** They have developed a deep Convolutional Neural Network model, being called Classification of Skin Lesions Network (CSLNet), to classify eight skin diseases: melanoma, melanocytic nevus, basal cell carcinoma, actinic keratosis, benign keratosis lesion, dermatofibroma, vascular lesion, and squamous cell carcinoma. CSLNet contains less filters and parameters to enhance efficacy and performance. The proposed network is formed of many layers and numerous filter sizes gathered into four units. Each unit comprises a different number of convolutional layers of various filter sizes, preceded by a batch normalization layer, and leaky ReLU activation function. The units are separated by a block containing a batch normalization layer, a leaky ReLU activation function, a convolutional layer where the filter size is 1×1 , and an average pooling layer of size 2×2 . The model is fed with labelled dermoscopic images referring to the eight diseases to classify, acquired from ISIC-17, ISIC-18, and ISIC-19 dataset, and of different resolutions. The classes are imbalanced. The images undergo some pre-processing techniques such as cropping to transform them into square images with centered lesion, and rescaling to 64×64 . The artifacts like hair, gel bubbles, ruler markers, ink markers, patches, and dark borders, are kept in the images to balance the classes. They perform data augmentation by rotation, translation and flipping. The proposed method can classify the skin lesions with a 94% precision, 93% sensitivity, and 91% specificity [14].

The majority of the methods mentioned in literature have focused on the detection of acne or a definite number of diseases, maximum eight classes. Some require features extraction and others don't. In this thesis, we have worked on developing a computer-aided diagnostic and classification system based on deep convolutional neural networks, classifying facial skin images into ten classes, including 8 face skin pathologies (Acne, Actinic keratosis, Angioedema, Blepharitis, Eczema, Melasma,

Rosacea, Vitiligo), normal skin, and no-face class. The classification process doesn't require the extraction of the ROI from face images since the system is trained regardless of face pose, illumination, image resolution, etc.

1.3 Facial skin diseases

Skin lesions are defined as the variation of a region of the skin by comparison to the surrounding tissue. They can originate in the different layers of the skin, and occur in childhood, adulthood or even born with. Facial skin lesions are caused by an infection, allergens, medications or system disorders. These lesions could be transitory or permanent, benign or malignant, hereditary or acquired. They have different locations, colors, sizes, and textures that can help to diagnose the disease and the underlying cause. Facial skin lesions are classified into primary or secondary. Primary lesions appear at birth or over a lifetime as a reaction to the internal or external environment (such as acne, rosacea, etc.). Secondary lesions are developed from primary lesions or result from an infection or scratching (like atopic dermatitis, etc.). Hundreds of facial skin disorders affect humans. In our work, we have chosen to classify eight diseases that are mostly spread and happen to all age groups. These disorders are acne, actinic keratosis, angioedema, blepharitis, eczema, melasma, rosacea, and vitiligo. In the remainder of this section, we will define these diseases, their symptoms and causes.

1.3.1 Acne

Acne is a skin disorder, affecting 3 in every 4 persons, males and females in the age group 11 to 30 years. It is a chronic, inflammatory, but not dangerous skin condition where the pores of the skin are clogged by hair, sebum, dead skin cells and bacteria. It is caused by the elevation of androgen hormone, an active hormone that increases during puberty. High androgen levels cause the growth of oil glands under the

skin, that secrete more sebum. The excessive production of sebum destroys cellular walls in the pores and thus bacteria grows. Hormones increase, surface bacteria, and sebum combined together cause acne. Other factors can cause or trigger acne such as oily or greasy cosmetics, stress, some medications, hormonal variations, menstruation, genetics, humidity [15]. Acne occurs mainly on the face, forehead, chest, and back. It has numerous forms [16]:

- Blackheads are open and dark pores at the surface of the skin.
- Whiteheads are close pores at the surface of the skin, due to oil and dead skin.
- Papules are red inflamed bumps with no visible fluid.
- Pustules are large bumps filled with yellow or white pus with red base.
- Nodules are large, hard and painful pimples under the skin.
- Cysts are large, painful pimples filled with pus. They can leave scars on the skin (Figure 1.1).



Figure 1.1: Different types of acne: (a) whitehead, (b) blackhead, (c) pustule, (d) papule, (e) cyst, (f) nodule [58].

1.3.2 Actinic keratosis

Actinic keratosis or AK, known as solar keratosis too, is a skin disorder characterized by a scaly, rough spots on the skin, due to chronic exposure to ultraviolet rays of sunlight or indoor tanning. It is a type of precancer, and if not treated, 5 to 10% of AKs could turn into a type of skin cancer named squamous cell carcinoma. It affects people most often after age 40 and younger persons who live in mild year round climate areas, who don't protect their skin from sun exposure especially persons having light skin, red or blond hair, green or blue eyes, weak immune system, and history of sunburns. The lesions appear on sun exposed regions such as the face, neck, lips, ears, scalp and hands. They are of various colors red, white, light or dark tan, pink, or the same color as the skin, and gray, but all have a yellow or brown crust on top (Figure 1.2). AKs have different symptoms: pain or tenderness, bleeding, itch, burn, and dry, scaly and colorless lips [17].



Figure 1.2: Examples of actinic keratosis on the face [58].

1.3.3 Angioedema

Angioedema is a swelling under the skin or mucosa. It is mainly due to a severe allergic reaction, that leads to histamine release by the body, that causes dilation of blood vessels and fluid leakage. The fluid accumulates and causes swelling. Angioedema is triggered by different allergens as insect bites, intolerance to food, medication, animal dander, and pollen. It also can be caused by an infection or disease like leukemia or lupus, or simply can be hereditary. The swelling can affect especially loose regions of tissue as the lips, tongue, eyes, as well as hands, feet, genitals, and respiratory and gastrointestinal mucosa (Figure 1.3). It has multiple symptoms: welts, itching, pain and warmth in swollen areas, swelling and redness around the eyes and lips. In severe cases, it can cause breathing problems, dizziness and abdominal pain [18].

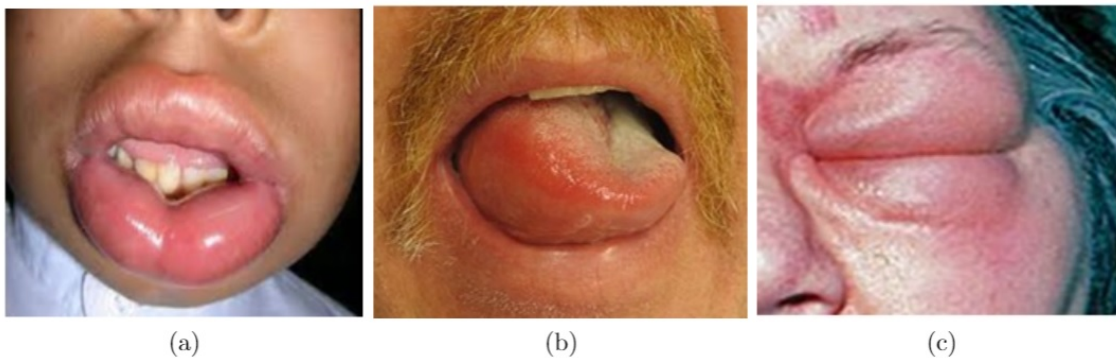


Figure 1.3: Angioedema in face: (a) lips, (b) tongue, (c) eyes [58].

1.3.4 Blepharitis

Blepharitis is the inflammation of the eyelid margin. It exists in two types of blepharitis: anterior blepharitis occurring at the external eyelid where the eyelashes attach, and posterior blepharitis affecting the inner eyelid in contact with the eyeball (Figure 1.4). Blepharitis happens when oil glands in the eyelids are blocked, provoking irritation and redness. It also may be caused by bacteria settling in the eyelashes, skin condition (dandruff of the scalp), rosacea, contact allergy to substances such as

cosmetic products and contact lens solutions. Persons with blepharitis suffer from different symptoms in the eyes including burning feeling, redness, dryness, and excessive tearing, and in the eyelids such as inflammation, itching, swelling and crusting. They may also experience blurry vision, sensitivity to light and inflammation of the cornea [19].

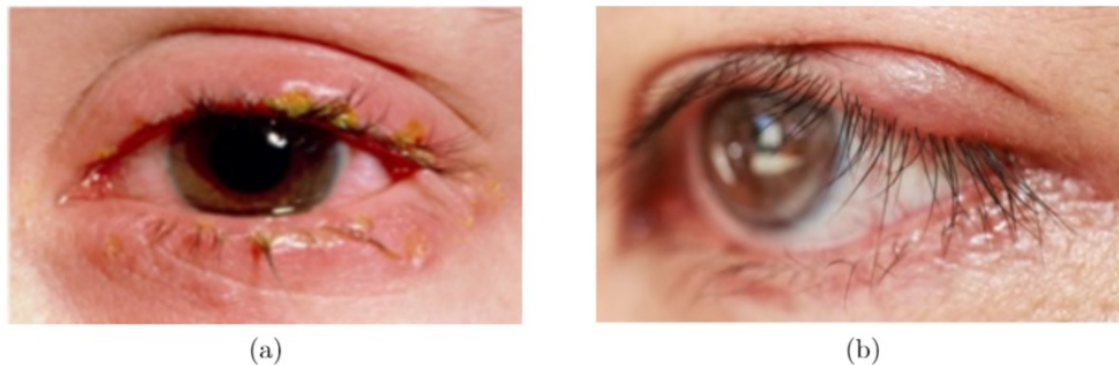


Figure 1.4: Blepharitis in face: (a) Posterior blepharitis, (b) Anterior blepharitis [58].

1.3.5 Eczema

Eczema is a skin disease characterized by red, inflamed, and itchy skin rash [20]. It can appear in any part of the body including the face. Facial eczema is classified into three groups (Figure 1.5):

- Atopic dermatitis is the most prevalent form of eczema affecting mainly children. It appears around the cheeks and chin. It also affects adults around the eyes, on the eyelids and around the lips [21].
- Seborrheic dermatitis is the most widespread form of facial eczema in adults. It happens around the ears, in the eyebrows and on the extremities of the nose [22].
- Contact dermatitis exists in two forms: Irritant contact dermatitis caused by cosmetics, toiletries detergents, solvents and friction. It leads to dryness and

pain of the skin around the eyes and around the hairline. The second form is allergic contact dermatitis provoked by the contact of the skin with substances usually non allergenic such as perfumes, jewelry, hair dye, rubber. . . it is found on the neck, and face [23].

All facial eczema types have almost identical symptoms including red spots, itching, burning, inflammation, dryness, swelling, rashes, and pain. Many factors can contribute to the development of eczema such as family history, allergies, autoimmune diseases and the age. Environmental irritants, hormone variations, food allergens, stress, and temperature trigger eczema.



Figure 1.5: Eczema images: (a) Atopic dermatitis in children; (b) Atopic dermatitis in adults; (c) Seborrheic dermatitis; (d) Allergic contact dermatitis; (e) Irritant contact dermatitis [58].

1.3.6 Melasma

Melasma is a skin pigmentation disorder characterized by brown or gray patches on the skin. It affects women more than man. 90% of persons developing melasma

are women. The patches appear on the cheeks, chin, forehead and bridge of the nose (Figure 1.6). It can also occur on other parts of the body such as the neck and forearms. The pigmentation is caused by the overproduction of melanin by the melanocytes, the color making cells. Sun exposure, skin care products, family history, hormone fluctuations, and pregnancy trigger melasma. When it occurs in pregnant women, melasma is called “mask of pregnancy”, and in this case, it disappears on its own [24].



Figure 1.6: Examples of Melasma: (a) on the cheeks; (b) on the forehead [58].

1.3.7 Rosacea

Rosacea is a chronic disease that mainly affects the face. It can be mistaken for eczema, acne or allergic skin reaction. Rosacea affects mainly more middle aged women with light skin than men. It causes redness across the cheeks, chin, nose and forehead. It also causes also pimples, dry skin, swelling, burning when using water or cosmetics, visible blood vessels, and thick skin in advanced phases (Figure 1.7). It may also cause eye problems. Rosacea could be a hereditary disease or due to the dilation of blood vessels leading to flushing and redness. It is triggered by spicy foods, alcohol, hot drinks, environmental factors, emotions, medications and cosmetic products [25].



Figure 1.7: Rosacea images: (a) acne-like rosacea; (b) rosacea with visible blood vessel [58].

1.3.8 Vitiligo

Vitiligo is a chronic disease caused by the deficiency of melanin, the pigment in skin, due to melanocytes death or stop functioning. The depigmentation is due to an autoimmune disease, stress, trauma or simply a family history. It mainly occurs on the face, neck and hands. Facial vitiligo is characterized by white patches developed on the skin, lips, and in the mouth. It affects all races and genders and mainly in persons between 10 and 30 years. Besides white patches, the most common symptoms of vitiligo include whiteness of eyelashes, eyebrows and beard hair, change of color of the retina, lighting of the tissues in the nose and mouth [26]. In some cases, persons may have pain, and itching (Figure 1.8).



Figure 1.8: Vitiligo Images [58].

1.4 Conclusion

Computer-aided diagnosis systems are computer-based techniques to assist doctors to interpret medical images and thus detect and diagnose the abnormalities in the images. They are widely used in different medical fields including dermatology. Computer-aided diagnosis systems used for facial skin diseases process images, extract features and then pass these features to a classifier to assign the skin lesion to the most likely disease. In this chapter, we have showed some automated techniques that have been developed by researchers to classify facial skin diseases images. Some methods are based on image processing techniques and pattern recognition. With the advances in technologies, high performance computers, and advances in machine learning and deep learning, artificial intelligence (AI)- based approaches have been suggested to classify skin lesions. We have proposed a deep learning based identification method capable of classifying several facial skin diseases. We have also presented the different diseases to be classified in our method. In the next chapter, we will provide an overview of artificial intelligence, machine learning and deep learning.

Chapter 2

Deep learning based skin diseases classification

2.1 Introduction

Deep learning is a subset of machine learning, which in turn, is a subset of artificial intelligence. The relationship between the three terms is represented in (Figure 2.1). The deep learning revolution started with the necessity of high accuracy predictive models for unstructured data like images, audio and natural language. Breakthroughs in artificial neural networks lead to the explosion of deep learning. Deep learning algorithms have ameliorated the capability to classify, identify, and detect. They can process huge number of features which makes them very efficient when working with unstructured data. Deep learning is currently used in several fields such as customer experience like chatbots, aerospace and military, industry, text generation, healthcare and medical research, and also in computer vision as image classification, speech recognition, object detection, and natural language processing. Big data, new machine learning methods, algorithmic ameliorations, novel neural networks models, powerful computing systems and also human-to-machine

development are all factors advancing deep learning.

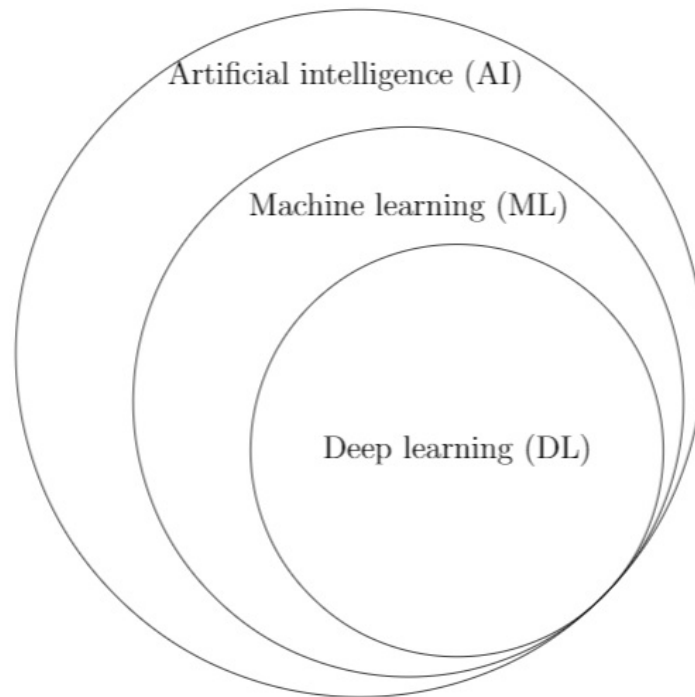


Figure 2.1: AI vs Machine learning vs Deep learning: A diagram describing the relation between artificial intelligence, machine learning and deep learning [27].

In our work, we are using deep learning to classify images, especially facial skin diseases images. Deep learning is considered as an efficient and an important machine learning tool in computer vision and image analysis. The principle idea of deep learning is that the model learns from large amount of data to classify images. It eliminates hand-engineered feature extraction. Deep learning models usually use neural network architectures, and that is why they are often called deep neural networks. In this chapter, we will define the terms artificial intelligence, machine learning and deep learning, show the difference between machine learning and deep learning and explain how deep learning works. Then, we will explore the fundamentals of artificial neural networks and how they work to reach deep neural networks. In this section, we will describe deep neural networks, discover the concept “feature hierarchy” that makes deep learning a powerful tool in computer vision, and present the different architectures of deep neural networks. In the rest of the chapter, convo-

lutional neural networks (CNN) will be explored: the blocks and layers constituting the network, its different architectures, as well as its training. Finally, we move to image classification using CNN. An overview of image classification definition will be provided, along with the challenges the network may encounter. The chapter ends by explaining the steps of training a CNN for image classification and how the classification is performed.

2.2 Artificial intelligence

Artificial intelligence (AI) is the ability of machines to mimic the work of the human brain in learning and processing data to perform recognition, identification tasks and make decisions without human intervention. The term AI was brainstormed in the 1950s, but has become very popular nowadays due to huge data volumes, advanced algorithms and enhancement of computers potential and storage. It has wide applications and in different fields such as image recognition, chatbots, speech recognition, sentiment analysis, chess playing computers, self-driving cars. . . .

2.3 Machine learning

Machine learning is a sub-field of AI enabling systems to automatically learn and recognize patterns on the basis of existing algorithms and datasets: learn from experience, and thus solve problems adequately. It is considered as the brain of AI systems. Machine learning systems are trained rather than programmed using structured and semi-structured data. These systems require a manual feature extraction, and then these features are used to create the model performing the desired task. Machine learning has numerous applications: automation, natural language processing, computer vision and image processing. In addition, machine learning algorithms are divided into two large classes: supervised and unsupervised learning. Supervised learning learns a function that maps an input (X) to an output (Y). Su-

ervised machine learning algorithms are given a known labeled set of inputs called training data and the desired output values. The algorithm learns automatically the patterns to make the correct prediction of the new data. This type of learning uses classification and regression techniques [28]. Unsupervised learning learns from unlabeled data and without predefined target values. The algorithm tries to identify discriminating characteristics in the input data. Clustering is the most common method used in unsupervised learning [29].

2.4 Deep learning

Deep Learning is a type of machine learning inspired by the architecture of the human brain, enabling the automatic learning [30]. It can work and learn from unstructured and unlabeled data. It trains the system by huge amounts of data (images, videos or texts) to learn the features from them using many layers of processing. Deep learning has found applications in the fields of computer vision: audio/speech recognition, automated driving, aerospace and defense, medical research, industrial automation, electronics. . . . Deep learning algorithms use a hierarchical level of artificial neural networks, comprising tens or hundreds of layers stacked on top of each other, to extract automatically the features from the data and improve the identification and the classification without human supervision. Deep learning models are capable of unsupervised learning.

2.4.1 Deep learning versus machine learning

Despite being a subfield of machine learning, deep learning differentiates itself from traditional machine learning algorithms by the type of data that it deals with and the learning method. Deep learning cancels some pre-processing of data that is commonly associated with machine learning. These methods can deal and process unstructured data such as audio, video, images. . . and extract automatically the

features without human intervention. Also, deep learning algorithms achieve end to end learning: the network is fed with raw data and learns automatically how to perform a given task, like classification (Figure 2.2).

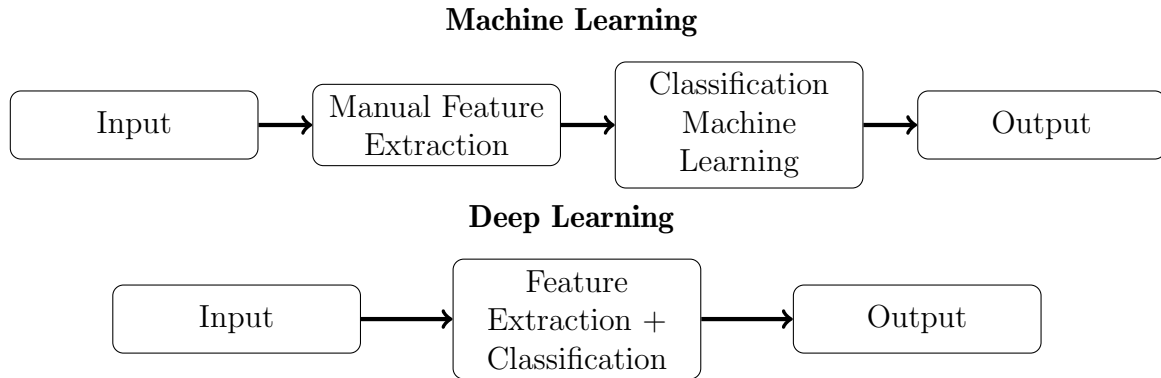


Figure 2.2: Machine learning and Deep learning technique: In machine learning technique, the features are manually extracted from the input, and then fed to a machine learning classifier to predict the output. While in deep learning, the network extracts automatically the features and learn them to perform the classification and predict the output.

Another major difference is the performance of deep learning algorithms improves as the size of data increases, while in traditional ML techniques the performance plateau to a certain level (Figure 2.3).

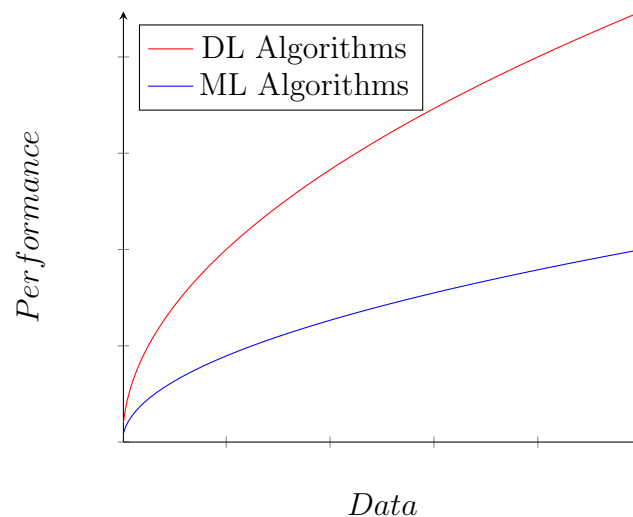


Figure 2.3: ML vs DL algorithms performance: The performance in deep learning algorithms (red curve) increases with the amount of data, while in traditional machine learning algorithms the performance plateau after it attains a threshold of data.

2.4.2 Comparison between machine learning and deep learning

Comparison between classic machine learning algorithms and deep learning algorithms (Figure 2.4).

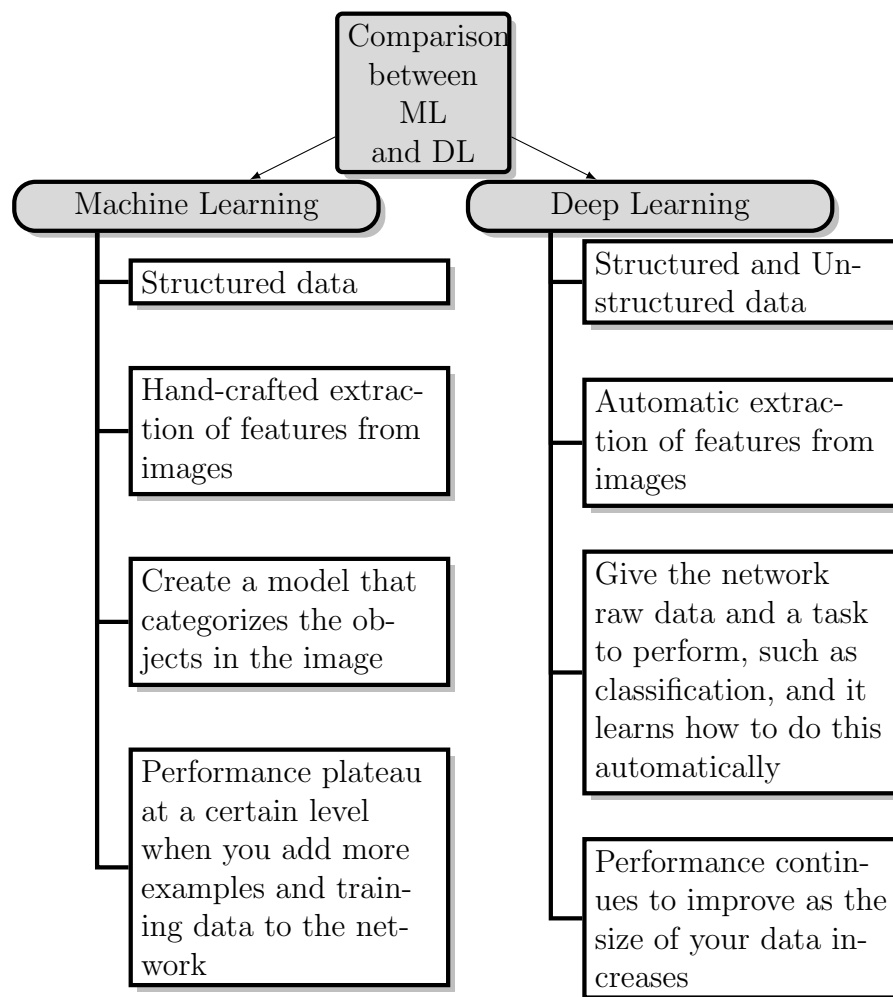


Figure 2.4: Comparison between classic machine learning algorithms and deep learning algorithms.

2.4.3 How deep learning works?

Deep learning methods are usually based on artificial neural networks. That’s why we refer to deep learning methods as deep neural networks. They use a hierarchical level of neural networks mimicking the human brain. The term “deep” in deep

learning stands for consecutive hidden layers of representations. Deep neural networks (DNN) are formed of numerous layers, tens or even hundreds, stacked on top of each other's, and they are all learned automatically during the training process. The hierarchical structure of deep networks allows data processing with a nonlinear approach. Each layer in the network utilizes the output of preceding layers to learn more complex concepts. First layers detect low level features as edges and the deeper you go in the DNN, the nodes identify more complex features by combining features from the preceding layer. This hierarchical learning eliminates the hand-crafted feature extraction in the network (Figure 2.5).

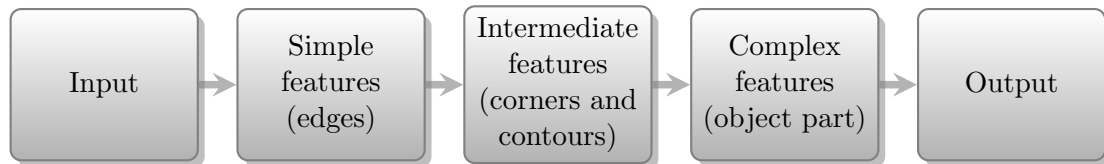


Figure 2.5: Deep neural network consisting of stacking layers learning more complex and distinct features as we go deeper in the network. Given an input image, the lower level layers detect the edges. The next layers detect the corners and contours. Combining the features extracted in the previous layers lead to learn more complex features. The output layer classifies the input.

Deep learning methods compute and predict regularly, learn the features during the training process and improve the accuracy over time, within each layer. Deep learning systems necessitate an enormous amount of data to train the model and thus obtain reliable results, and a powerful hardware to process this data rapidly.

2.5 Neural networks basics

Artificial neural networks (ANN) are computing systems built like the biological neural network constituting the human brain, with hundreds or thousands of interconnected neuron nodes. Artificial neural networks are trained by data to learn to classify data, recognize patterns and predict future events. Artificial Neural net-

works are made up of many processing layers: an input layer receiving data from the external world, one or more hidden layers processing the data, and an output layer performing the desired task (classification, recognition, clustering...) (Figure 2.6). Each layer is composed of several neurons and uses the output of the preceding layer as its input, so the layers are interconnected by the neurons.

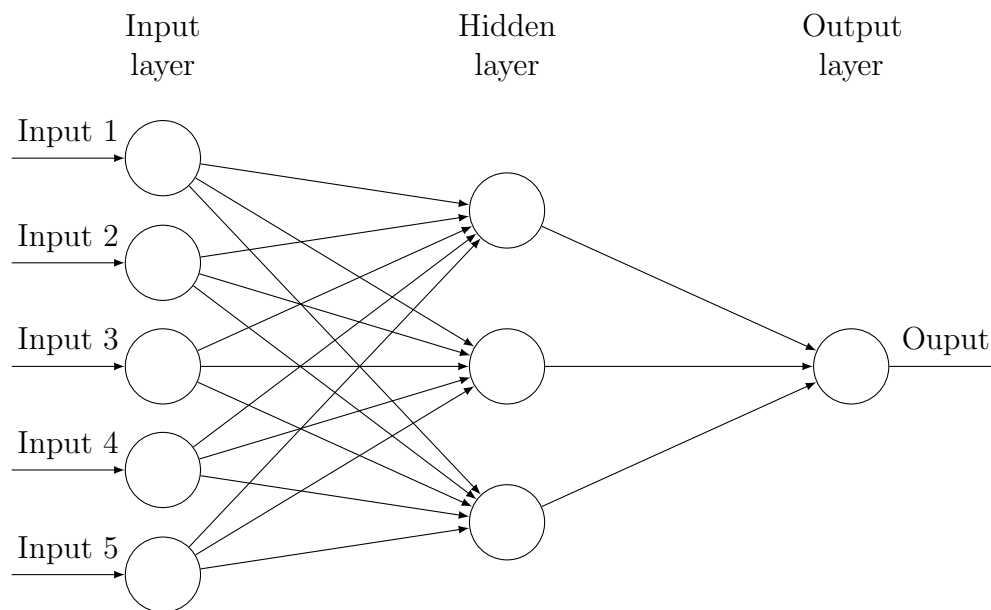


Figure 2.6: Artificial neural network architecture: the network consists of an input layer, hidden layer and an output layer. Each layer contains several neurons receiving inputs from previous neurons.

2.5.1 How artificial neural networks work?

The network receives an input signal (image, audio, text...) in the form of vector defined by $x(n)$. Each input is multiplied by a corresponding weight representing the strength of the signal of the neuron, connected to a neuron. All the weights are added in the computing unit. In case the weighted sum is zero, we add a bias to adjust the system's response. The weighted sum passes through an activation function f and then the output y is generated (Figure 2.7), and given by equation (2.1).

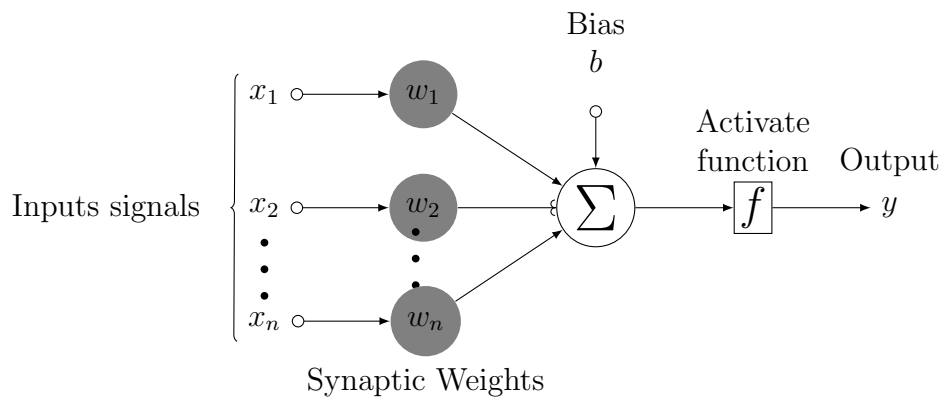


Figure 2.7: Structure of ANN: the inputs are multiplied by the weights and then summed. A bias can be added and then the weighted sum is activated by an activation function to generate the output. The activation function determines if the neuron should be activated or not.

Neural networks have two propagation functions: the forward propagation that helps in predicting the output and the backward propagation that minimizes the error between the actual output and the predicted one. Actually, these two functions depend on each other when training the network. The forward propagation is the computation and storage of variables for an ANN from the input to the output layer. While the backpropagation is the computation of the gradient of the ANN parameters with respect to all weights in the network. The forward propagation is the computation and storage of variables for an ANN from the input to the output layer. While the backpropagation is the computation of the gradient of the ANN parameters with respect to all weights in the network.

$$y = f\left(\sum_{i=1}^n x_i w_i + bias\right) \quad (2.1)$$

2.5.2 Activation function

Activation functions are mathematical models that define the output of a neural network. They decide if the neuron should be fired or not. They are a non-linear

transformation used to prevent the convergence of the network, normalize the output between $[0,1]$ or $[-1,1]$ and reduce the computation time. There are many types of activation types. The most common ones are: step function, sigmoid function, hyperbolic tangent and rectified linear unit.

1- Step function

It is the simplest activation function and defined by:

$$U(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

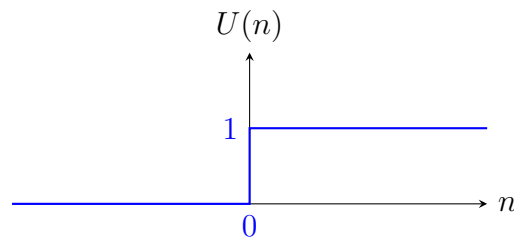


Figure 2.8: Step function.

As seen from the equation above, we can see that this function makes the output 1 when the weighted sum is greater or equal to zero and 0 otherwise (Figure 2.8). It is mostly used in binary classification problems. The step function is non differentiable which causes a problem during the training process and thus the weights can't be updated.

2- Sigmoid function

It is a non-linear, real and differentiable activation function characterized by:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

The output varies between 0 and 1 (Figure 2.9). The main disadvantage of the

sigmoid function is that it is not zero centered and the neurons saturate which kills the gradient and thus the training can not be achieved.

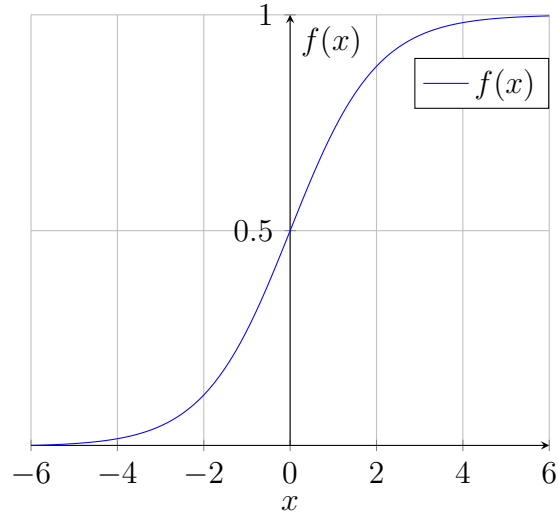


Figure 2.9: Sigmoid function.

3- Hyperbolic tangent [31]

It is a continuous zero-centered function ranging between -1 and 1 (Figure 2.10). It is represented by:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.4)$$

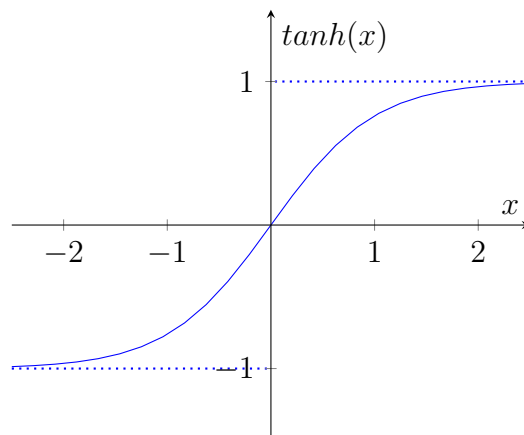


Figure 2.10: Hyperbolic tangent [31].

4- Rectified Linear Unit (ReLU)

ReLU is the most generally used function in neural networks because it speeds up the learning process and improves the performance. Despite its linear form, ReLU is not linear and is differentiable [32]. It gives an output 0 if the input is negative and x for positive values (Figure 2.11). It is designated by [33]:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (2.5)$$

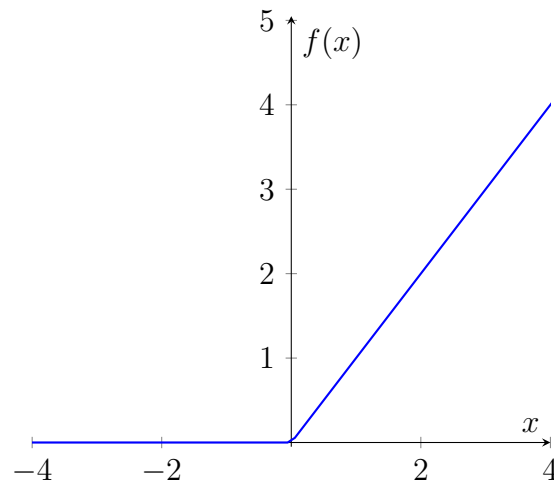


Figure 2.11: Rectified Linear Unit (ReLU) [33].

5- Cost function

The cost function is an important component of a neural network. It determines how badly the network is performing. It measures the error between the predicted output and the actual output. This error is back propagated in the network and minimized during the training process by adjusting the weights and biases repetitively.

2.6 Deep neural networks

Deep neural networks are large artificial neural networks composed of several hidden layers between the input and output layers [34]. The layers in a deep neural

network are interconnected. Each is built upon the preceding layer to process data and optimize the classification or prediction. This evolution of calculations through the network is named forward propagation. The data is fed in the input layer and the output layer makes the prediction or classification. The error in predictions is computed by backpropagation, such as gradient descent, and then the weights and biases are adjusted by moving backward throughout the layers during the training process. With time, the accuracy of the model improves. Each layer is trained with the representations of the preceding layer to learn a more complex abstraction (Figure 2.12). The layers in DNN learn incrementally the features from data. Early layers detect low level features and following layers aggregate features from previous layers in a more integrated and full representation. That's what we call feature hierarchy (Figure 2.13). DNN can deal with unstructured and unlabeled data, which is the majority of data in real world such as pictures, audio recordings, texts, and video.

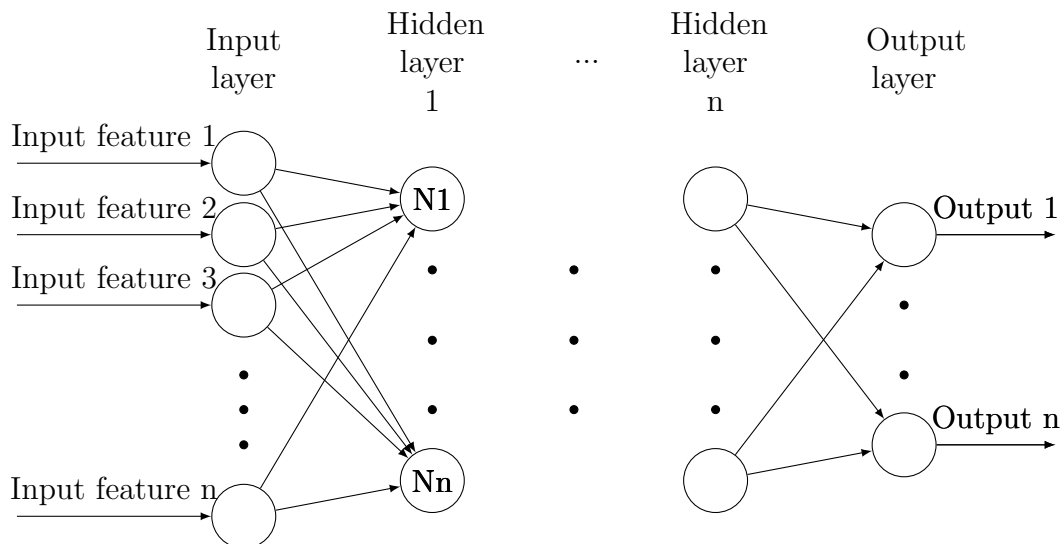


Figure 2.12: Deep neural network architecture: the network is composed of an input layer, n hidden layers and an output layer.

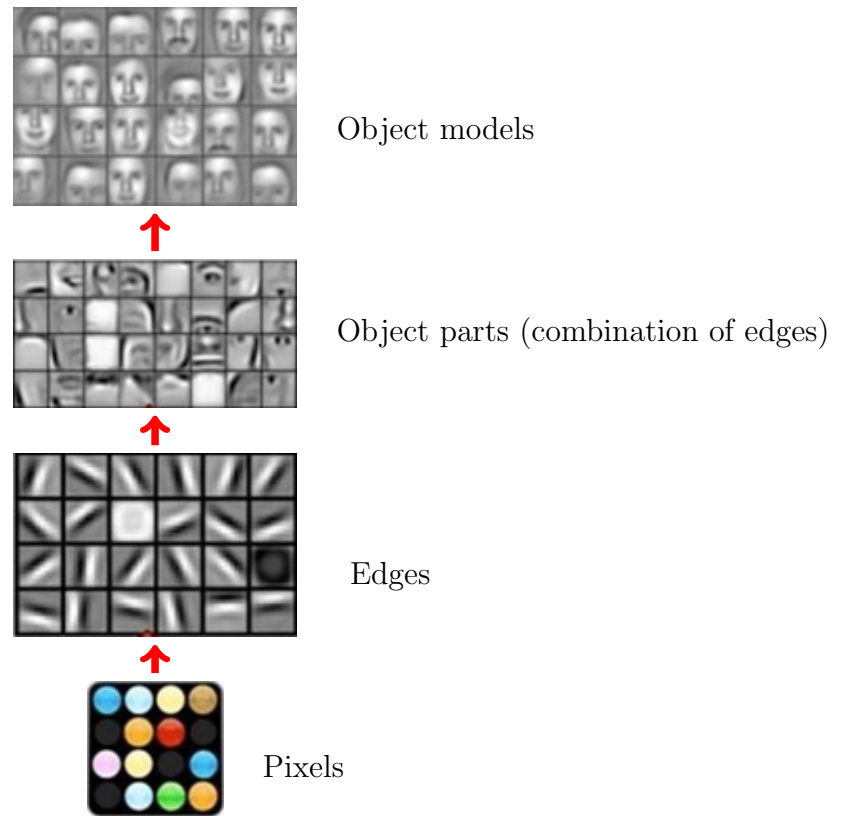


Figure 2.13: Example of feature hierarchy learned by a deep neural network on faces: the network learns first to recognize pixels and then edges and contours. These features are combined and mid-level features are learned such as eyes, noses, ears... In the last layer, high-level features are extracted, and the network learns which forms and objects can be used to identify a human face [34].

2.6.1 Architecture of deep neural networks

Deep learning architectures are divided into supervised and unsupervised learning.

1- Unsupervised learning: In unsupervised learning, the deep neural network learns patterns from unlabeled data. The most common unsupervised deep learning architectures are autoencoders [35] and deep belief networks [36].

2- Supervised learning: In supervised learning, the deep neural networks are trained using labelled data. The most used supervised deep learning architectures are recurrent neural networks [37], and convolutional neural networks [27].

In our work, we use convolutional neural networks that will be described in details in the next section.

2.7 Convolutional neural networks

Convolutional neural networks, also known as CNN, are a class of feed-forward deep neural networks mostly used in image analysis. They take images as input. They are widely used in different computer vision tasks and find interest across various domains. Unlike traditional feed-forward networks, CNNs use fully connected layers only in the very last layer. The fully connected layer is substituted by a convolutional layer for at least one of the layers of the network [27]. The convolution layers are followed by a nonlinear activation function, such as ReLU, till the end of the network where we can add one or two fully connected layers to obtain the final output classifications. The core of a CNN is the convolutional layer that gives its name to the network. The term “convolution” refers to the mathematical function of convolution. In mathematics, the convolution is a linear operation on two functions that provides a third function that indicates how the shape of one is changed by the other. Whereas in deep learning, it is a dot product of two matrices followed by a sum. In terms of CNN, it is the multiplication of the input with a series of weights. In case of a 2D input, the multiplication is done between an array of input data and a 2D array of weights, denoted filter or kernel. The convolution between 2D input image I and 2D kernel K is:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n K(i - m, j - n)I(m, n)) \quad (2.6)$$

Since its size is smaller, this filter is multiplied many times by the input array at different points. Hence, the filter is applied systematically from left to right and from top to bottom to cover the whole image. Actually, the filter is conceived to detect a specific type of feature in the input and the systematic application across

the whole image and this enables it to detect that feature anywhere in the image. This is known as translation invariance. The result of multiplication between the filter and the input one time gives a single value. As the filter is applied numerous times to the input, we obtain a 2D output array called “feature map” (Figure 2.14).

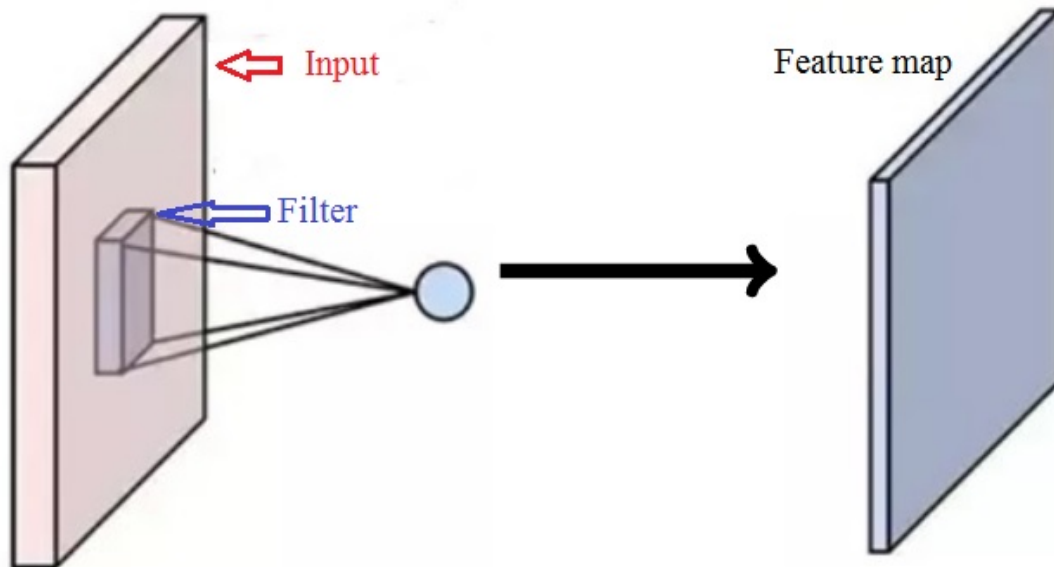


Figure 2.14: Example of a filter applied to an input to produce a feature map

2.7.1 CNN blocs

CNNs are built using many types of layers. These layers are arrayed in a 3D volume in three dimensions: width, height and depth, where depth refers to the number of channels in the image or the number of filters in the layer. The architecture of CNN consists of an input layer, output layer and hidden layers. (Figure 2.15). The hidden layers consist of convolutional layers followed by an activation function, pooling layers, fully connected layers and normalization layers stacked in a specific manner. Only convolutional and fully connected layers contain parameters that are learned during the training of the network. The convolutional and pooling layers are responsible of feature extraction while the fully connected layer map these features to predict the output (Figure 2.16).

Convolutional Neural Network

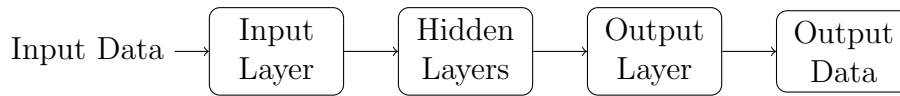


Figure 2.15: General block diagram of a CNN. The network is composed of 3 main blocks: input layer, hidden layer and output layer.

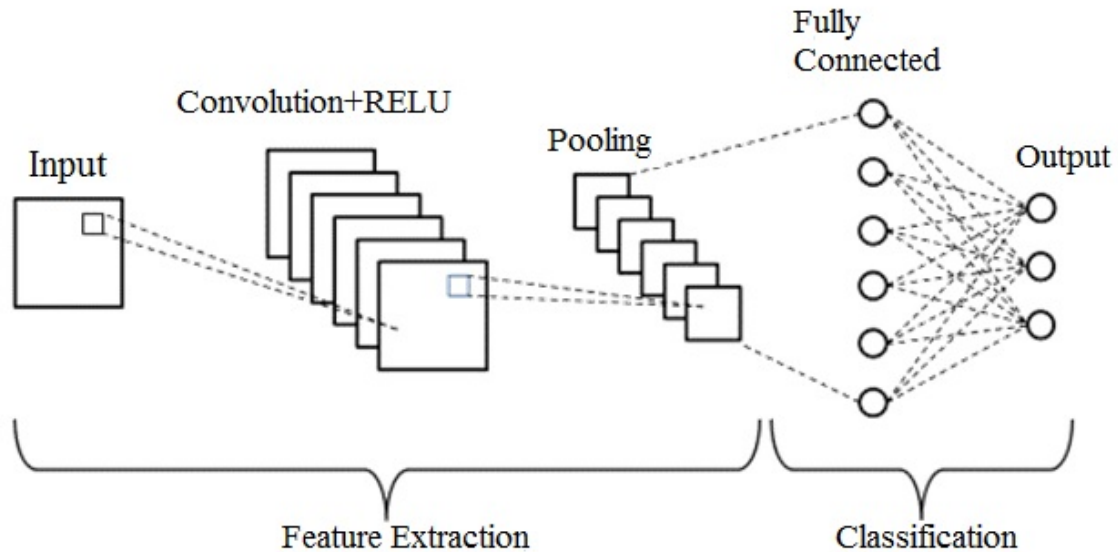


Figure 2.16: Basic CNN architecture. The network is mainly composed of an input layer, a convolution layer, a pooling layer, a fully connected layer and an output layer [38].

In the remainder of this section, each of these layer types will be reviewed in detail and the parameters associated with each one and how to set them, and how to train CNNs will be discussed as well.

1- Convolutional layer

The convolutional layer, or CONV layer, is the core of a CNN and is used to extract features from the input. The CONV layer has a set of learnable filters or kernels, each having a width and a height. These filters are small but applied through the full depth of the volume. The convolution is done between the input image and K kernels of size $M \times M$. Each kernel is slid across the input image and convolved with it. The result of each convolution is a 2D output called feature map giving

specific information about the image. Once the K kernels are applied to the input, we obtain K , 2D feature maps, stacked along the depth size of the array to form the final output volume that is fed to the next layer in the network. Each feature map represents specific characteristics of the input (Figure 2.17). The kernels are thus considered as feature extractors.

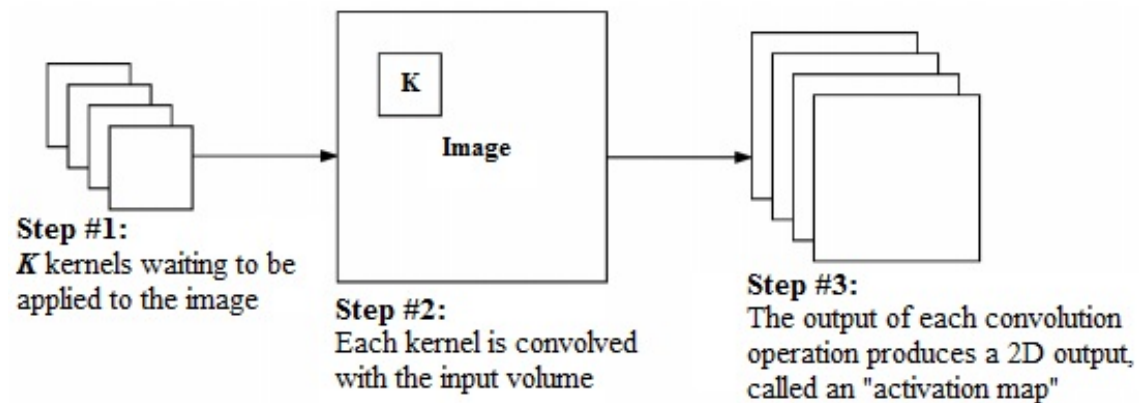


Figure 2.17: K kernels are applied to the input at each convolutional layer. Each one is convolved with the input to produce a feature map. After application of all kernels, we obtain K feature maps [38].

Every map in the output is an output of a neuron looking at a small zone of the input. In this way, CNN learns filters that trigger when they detect a specific feature at a spatial region in the input. In low layers, filters are triggered when they detect low-level features such as edges or corners. While in deeper layers, they are activated when seeing high-level features like object parts. In CNNs, each neuron is connected to a spatial region of the input. The size of this spatial region is called the receptive field F of the neuron. The size of the output is controlled by three parameters: the depth, stride and zero padding. The depth of the output is a hyperparameter, equal to the number of filters applied in the current layer. The stride S refers to the distance between two consecutive slides of the filters. It affects the size of the output volumes. Large strides contribute to less interfered receptive fields and small output volumes, while smaller strides result in an interfered receptive field and large output volumes (Figure 2.18).

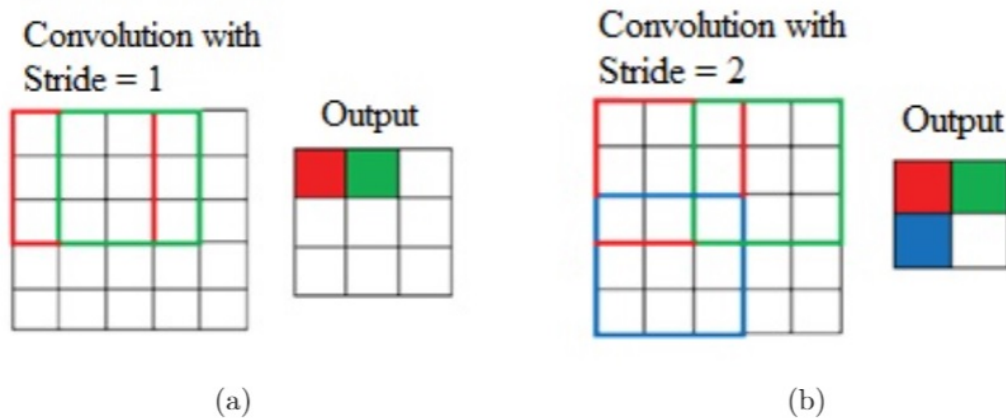


Figure 2.18: Example of stride. Given an input image 5x5 along with a 3x3 kernel. (a) When the stride is 1, the filter jumps one pixel at a time from left to right and top to bottom producing an output 3x3. (b) When the stride is 2, the filter slides two pixels at a time along the whole image providing a smaller output of size 2x2 [38].

Thus, beside feature extraction, we can observe that the convolutional layer also reduces the spatial size of the output by simply modifying the stride of the filter. The zero padding is a technique used to retain the spatial size of the input so that the input and output have the same height and width after applying the convolution. It consists of padding the borders of the input volume with zeros. It is a hyperparameter denoted by P . The zero padding protects the input volume from decreasing rapidly which makes us unable to train the network (Figure 2.19). The size of the output volume can be computed in function of the input volume size W , the receptive field of the neurons F , the stride S , and the value of zero padding P . It is given by integer of:

$$(W + 2P - F + 1) \tag{2.7}$$

If the result is not an integer, this means that the stride is incorrect and thus the neurons don't fit efficiently and uniformly across the input. In summary, the convolutional layer takes an input of dimension $W_1 \times H_1 \times D_1$ and necessitates four hyperparameters: the number of kernels K , the receptive field F , the stride S and

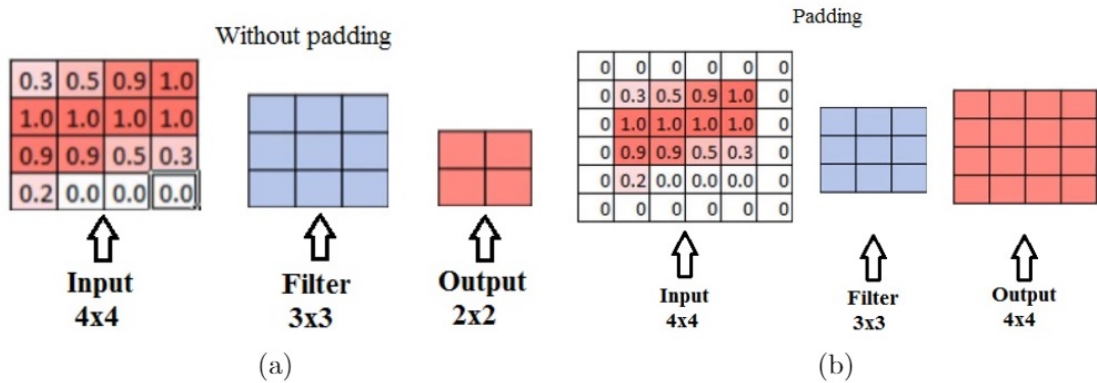


Figure 2.19: Example of zero padding. Given an input image 4x4 convolved with a 3x3 kernel. (a) Without padding, the convolution results in an output of size $(4-3+1) = 2$, 2x2. (b) with padding $P = 1$, the spatial size of the output volume increases to $(4+2-3+1) = 4$, 4x4. The input and the output have the same size [38].

the zero padding P .

It provides an output volume of size $W_2 \times H_2 \times D_2$, where [38]:

$$W_2 = \left(\frac{W_1 - F + 2P}{S} \right) + 1; H_2 = \left(\frac{H_1 - F + 2P}{S} \right) + 1; D_2 = D_1 = K \quad (2.8)$$

2- Activation layer

The convolutional layer is usually followed by a nonlinear activation function. It determines which info of the model should fire at the end of the network and which ones should not. There are many activation functions as seen previously, but recently the most commonly used one is ReLU since it almost has a better performance than others and speeds the training of the network.

3- Pooling layer

The pooling layer is embedded between two consecutive CONV layers. It decreases the size of the input of width and height to lessen the number of parameters and the computational costs, and also controls overfitting. It works separately on each feature map. There are two types of pooling: max pooling and average pooling. Usually, the max pooling is performed in the middle of the network to decrease spatial dimension, while average pooling is done in the final layer in case we don't

want to use fully connected layers entirely. Max pooling is the most common type of pooling. Average pooling calculates the average of all the values from the section of the image covered by the filter. Whereas in max pooling, the largest value is taken. Pooling layer takes an input of size $W_1 \times H_1 \times D_1$. It requires two hyperparameters: the pool size F , and the stride S .

It provides an output of size $W_2 \times H_2 \times D_2$ where:

$$W_2 = \frac{W_1 - F}{S + 1}; H_2 = \frac{H_1 - F}{S + 1}; D_2 = D_1 \quad (2.9)$$

Usually, we use a pool size of 2×2 or 3×3 in networks using larger input images and a stride $S = 1$ or $S = 2$. When we increase the stride, the size of the output decreases (Figure 2.20).

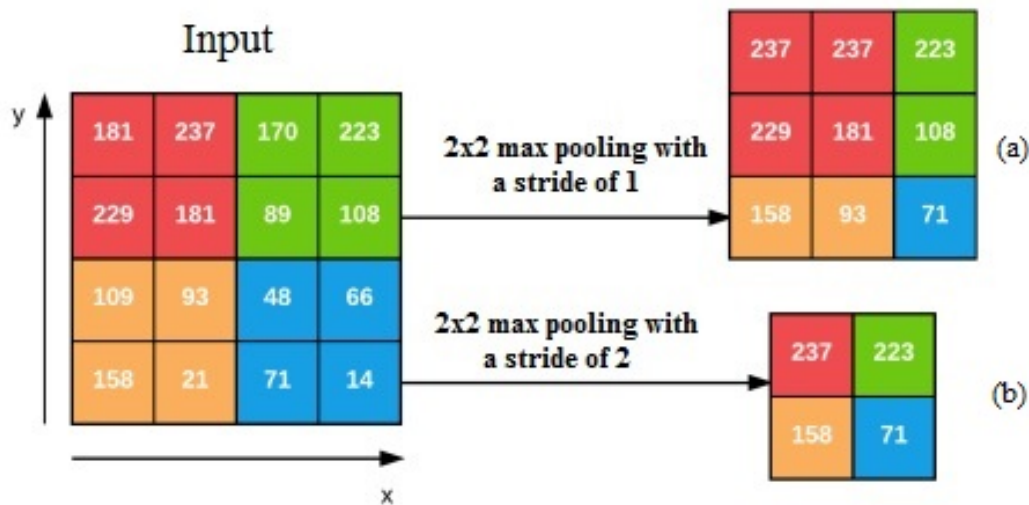


Figure 2.20: Example of max pooling. Given an input image 4×4 . (a) we apply a 2×2 max pooling with a stride of 1. (b) we apply a 2×2 max pooling with a stride of 2. The spatial size of the input is reduced [38].

4- Fully connected layers

Fully connected layers (FC) are the last few layers of the network, generally put before the output layer. They consist of weight, biases and neurons that are fully connected to all activations in the preceding layer. The FC layers are fed with flat-

tened output from the previous layer. They are used to classify images between different classes. FC are followed by a softmax classifier that computes the probability of each class (Figure 2.21).

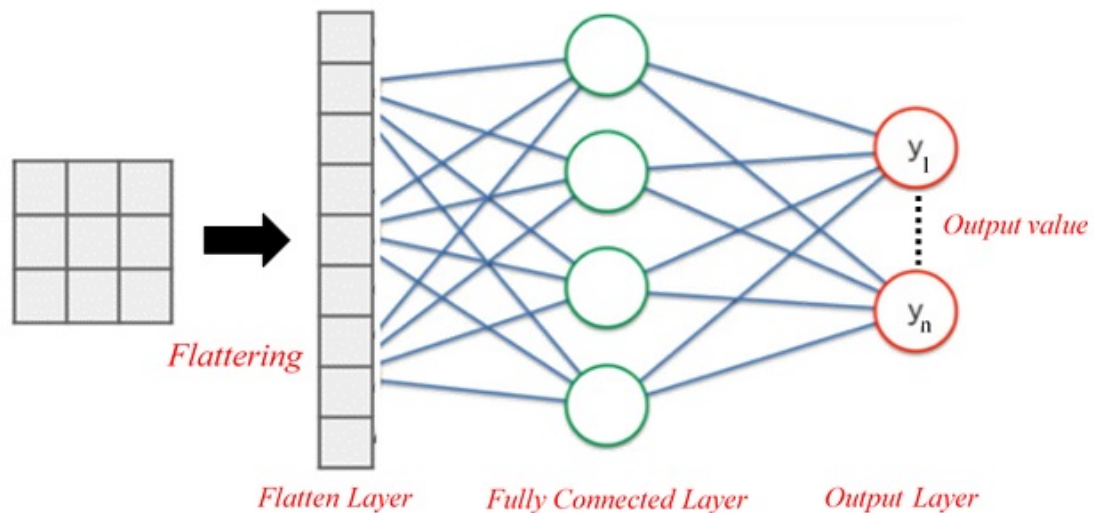


Figure 2.21: Example of fully connected layer. The FC layer is preceded by a flatten layer that converts the output of the convolutional layers into 1 dimensional array. The neurons in the FC layer are fully connected to all activations in the flatten layer. FC layer is followed by an output layer that computes the output.

5- Output layer

It is the last layer of the network and contains the label in one-hot encoded form. In practice, other layers could be added to control the network. The most common ones are batch normalization and dropout layers.

6- Batch normalization layer

Batch normalization (BN) layers are used to stabilize and accelerate learning of the network by diminishing the number of training epochs required to train the network. They can be inserted before or after the activation. This can be determined by experiment. BN layers normalize the input values of the next layer for each mini batch. Furthermore, they can prevent overfitting and help attain high classification

accuracy in less epochs compared to the same model without batch normalization layer [39].

7- Dropout

Dropout is a regularization technique used in CNN to prevent overfitting caused by the connection of all the features to the FC layer. Overfitting happens when a model performs well on the training set provoking a negative effect in the model's performance when applied on a new data. The dropout layer randomly selects neurons and sets them to zero during the training (Figure 2.22). Therefore the size of the model is reduced [40].

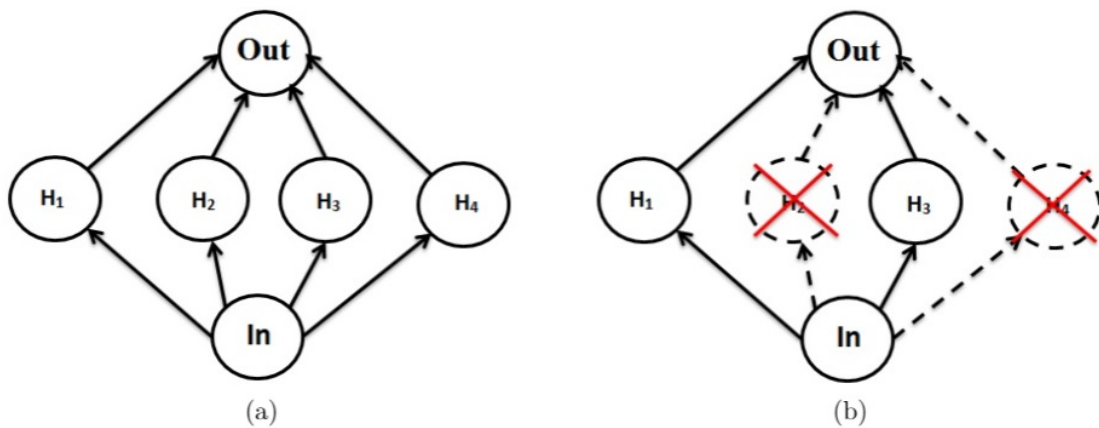


Figure 2.22: Example of dropout: (a) A standard network with one hidden layer without dropout. (b) The same network after dropout of probability $p = 0.5$. 50% of the neurons is randomly dropped out from the model [40].

There are several CNN models differing by their architecture and the number of layers in the network. The most common ones are: LeNet [41], AlexNet [42], GoogleNet (also called Inception) [43], and VGGNet [44].

LeNet is the first successful model of CNN developed by Yann LeCun in 1998. It is a straightforward and small network composed of 5 layers with learnable parameters. This architecture is used for image classification and especially recognition of handwritten and machine printed characters. It takes a grayscale image in input.

It consists of 3 convolution layers, 2 average pooling layers, and 2 fully connected layers with a softmax classifier [41].

Alexnet is proposed in 2012 by Alex Krizhevsky and his colleagues. It has the same architecture as LeNet but deeper with 8 layers with learnable parameters. The input to the network is RGB images. This model is composed of 5 convolution layers, max-pooling layers, and 3 fully connected layers with a Softmax classifier. They use ReLU as activation function. The network includes also two Dropout layers [42].

GoogleNet is a network proposed by Szegedy et al. from Google in 2014. It uses inception modules, which permit to choose between several filter sizes in each convolution block. It is formed of 22 layers but with reduced number of parameters compared to previous models [43].

VGGNet is introduced in 2014 by K. Simonyan and A. Zisserman (Very Deep Convolutional Networks for Large-Scale Image Recognition, conference paper at ICLR 2015). It is one of the most popular networks in deep learning applications. VGGNets are fed with RGB images of fixed size 224x224. They replace large size filters (11 x11 and 5 x 5) in Alexnet with smaller ones of size 3x3 which results in reducing the number of parameters. They show that the performance of the network increases with depth. There is a different architecture of VGGNet such as: VGG-11, VGG-13, VGG-16 and VGG-19 where the number within each name refers to the number of deep layers in the network [44].

ResNet or Residual Network is developed by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang in 2015. It is one of the most used and successful models. ResNet is based on the concept of skip connections and the use of batch normalization. Its architecture is inspired by the VGG-19 but doesn't include FC layers at the end of the network. They are replaced by an average pooling layer. All ResNet models have only one max pooling layer after the first one. It There are several

architectures of ResNet differing by the number of layers that vary from 18 to 152 [45].

2.7.2 CNN training

CNN training consists of finding kernels in CONV layers and weights in FC layers that reduce changes between output predictions and real labels on a training set. The training comprises two stages: a forward propagation and a backpropagation (Figure 2.23). In the forward phase, the input is propagated through the network to produce an output. In this stage, each layer will store all variables computed and used that will be needed in the backward phase. In the backward phase, the gradients are back propagated and the weights are updated. Backpropagation algorithm is the approach generally used to train and optimize neural networks. The weight matrix and the bias vector are the learnable parameters of the classifier that should be optimized to minimize the loss function and thus increase the performance of the network. The most prevalent optimization algorithm used is the gradient descent.

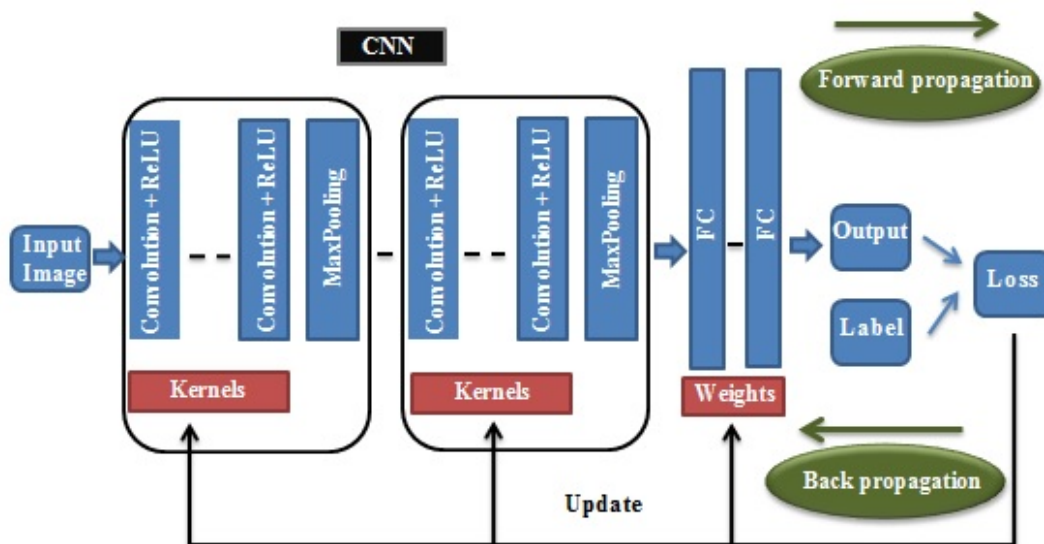


Figure 2.23: Training process of CNN. The performance of the model, with specific kernels and weights, is computed with a loss function, by forward propagation, quantifying the agreement between the predicted class labels and the ground truth ones. The kernels and weights are updated by backpropagation using gradient descent optimization algorithm [46].

1- Loss function

The loss function or cost function measures the similarity between predicted class labels through forward propagation and ground truth labels. The greater the compatibility between the two sets of labels, the less the loss function. During training our network, we aim to minimize the loss function, so the accuracy of classification increases.

2- Gradient descent

Gradient descent is the most common optimization algorithm used to train deep neural networks. It updates iteratively the parameters of the network, such as kernels and weights in CNN, to minimize the loss function by iteratively moving in the direction of steepest descent. Hence, each parameter is updated in the negative direction of the gradient with a random step called learning rate. Mathematically, it is the partial derivative of the loss with respect to the learnable parameter. The single update of a parameter is given by:

$$w - \alpha * \frac{\partial L}{\partial w} \tag{2.10}$$

Where w is the learnable parameter, α the learning rate and L the loss function. The learning parameter is a hyper parameter that should be determined before starting the training. Actually, the gradient descent is slow to run on big datasets. Instead, we use Stochastic Gradient Descent (SGD), a modification of the gradient descent algorithm, that calculates the gradients and updates the parameters on mini batches of the training dataset, rather than the whole training set. The mini-batch size is a hyper parameter. The typical batch sizes are 32, 64, 128, and 256. SGD is the most commonly used algorithm to train deep neural networks [47] (Figure 2.24). Numerous improvements have been applied to the gradient descent algorithm and are widely used such as momentum, Nesterov acceleration, RMSprop [47], and Adam [48].

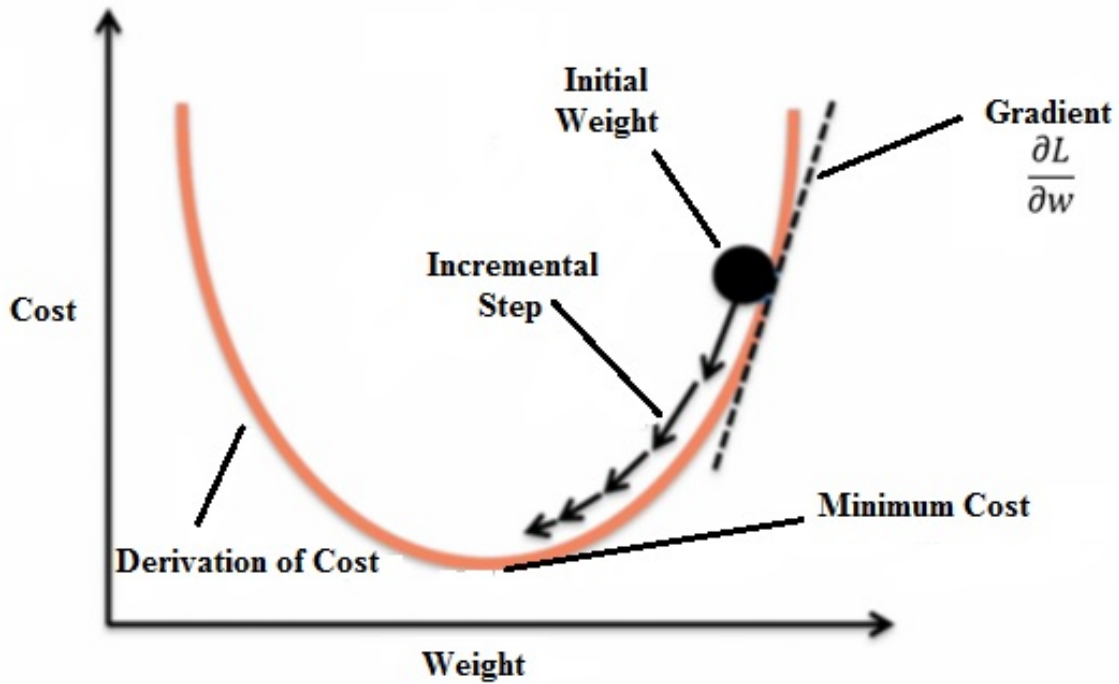


Figure 2.24: Gradient descent algorithm: it consists of iteratively updating the learnable parameters to minimize the loss, which calculates the distance between a predicted class label and a ground truth label. The gradient of the loss function determines the direction in which the function has the steepest descent. The parameters are updated in the negative direction of the gradient with a definite learning rate [47].

3- ADAM

Adaptive Moment Estimation or Adam optimization algorithm is an extension of SGD for training deep neural networks. It was proposed by Diederik Kingma and Jimmy Ba in 2015 [48]. It is mostly used when dealing with problems comprising a great number of data or parameters, and has limited memory requirements. Adam is a combination of SGD with momentum and RMSprop. It computes the exponential moving average of the gradient as SGD with momentum and the squared gradients like RMSprop, resulting in a more optimized gradient descent. This algorithm involves two parameters β_1 and β_2 called decay rates of average of gradients. The moving averages m_t and v_t are initialized to 0, and β_1 and β_2 are close to 1 which, leads to a bias towards 0.

They are given by:

$$m_t = \beta m_{t-1} + (1 - \beta) \left[\frac{\delta L}{\delta w_t} \right] \quad (2.11)$$

$$v_t = \beta v_{t-1} + (1 - \beta) * \left[\frac{\delta L}{\delta w_t} \right]^2 \quad (2.12)$$

Where δL is the derivative of Loss Function, δw_t is the derivative of weights at time t , and β is the moving average parameter of value 0.9. This bias is fixed by calculating bias corrected. This also helps to reach global minimum efficiently with minimal oscillations. The formulas are given by:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.13)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.14)$$

Where \hat{m}_t and \hat{v}_t are the bias corrected m_t and v_t , β_1^t and β_2^t are the decay rates of average of gradients at time t . The gradient descent is adapted after each iteration on a current mini-batch to be controlled and unbiased during the process, from where the naming Adam comes. Finally, the weight update is performed following the equation:

$$w_{t+1} = w_t - \hat{m}_t \left(\frac{\alpha}{\sqrt{\hat{v}_t} + \varepsilon} \right) \quad (2.15)$$

Where w_t is the model weight at time t , α is the learning rate, and ε is a small positive constant.

2.8 CNN and Image Classification

Image classification is a very interesting and large field of study, involving an enormous variety of approaches, including deep learning. It is a sub-field of computer vision permitting computers to see the world as we do. The invention of deep learning made image classification more widespread. Image classification has various applications in healthcare, industry, social networks, marketing and others.

2.8.1 What is image classification?

Image classification is the labelling of images into a predefined set of classes. Previous image classification techniques relied on raw pixel data. The problem is that images of the same thing could be different. Many factors vary such as: backgrounds, sizes, viewpoint, deformation, illumination, intra-classes variation, and poses. This makes it challenging for a computer to correctly see and classify images. This problem is solved with deep learning. Deep learning approaches have a robust learning ability, that incorporates the feature extraction and classification process to perform the image classification test, that enhances the accuracy of classification. Image classification in deep learning is a data driven approach. It is a supervised learning, consisting of giving examples of each class and teaching our model to identify the difference between the classes based on these examples. These examples are called training dataset, where each example in the dataset is an image with a label. The labels are used by the algorithm to teach itself how to identify each class.

2.8.2 Image classification based on Deep learning

To construct a deep learning based image classifier, four steps are required. First, we have to gather our dataset. The dataset comprises labelled images. The labels define a finite set of classes (cat, dog, cup, car...). The number of images within

each class should be identical. Unbalanced classes lead the classifier to overfit over fit. Second, the dataset is created, we split it into two parts: a training set, and a test set. The training set is used to train the classifier to learn how each class looks. It predicts the input data and then adjusts itself when the predictions are incorrect. After being trained, the performance of the classifier is evaluated on the testing set. The training set and testing set should be different and don't overlap to evaluate correctly the performance of the classifier. If the testing set is a part of the training set, the classifier would have already seen the example and learned from it. We have many split cases. Actually a third set could be added, the validation set. It is a part of the training set and used as a fake test set. Generally, it forms 10-20% the training set. The third step is training the network using the training set. The model learns how to identify each class in the labeled data. CNN detects low level characteristics like edges in the first layer, and the following layers combine the detected features previously to detect higher-level ones such as shapes in the second layer and objects in the highest layers. The last layer utilizes the higher level features to make the predictions. In case of a misclassification, it learns from this error and enhances itself using gradient descent algorithms. Last, we evaluate our trained model using the test set. The network predicts the class of each image in the set. These predictions are compared to the ground truth labels, representing the correct labels, from the test set. The performance is evaluated by computing the number of correct predicted labels, and classification metrics such as precision, recall, accuracy and F1-score.

2.8.3 Convolutional Neural Networks for image classification

CNNs show a huge advance in image recognition. They are used to analyze images and are widely used in image classification due to many reasons. CNNs are end-to-end models which allows the user to skip the hand-engineered feature extraction

stage. Instead, the network learns the features. The network is fed by raw input data and then learns filters that can differentiate between classes in the hidden layers. CNNs reduce the number of parameters without losing the features. The number of parameters increases with the number of layers which can slow down the training process and thus makes parameters tuning a massive task. CNNs decrease the tuning time. Layers of a CNN have numerous convolutional filters that work and scan the entire feature matrix and perform dimensionality reduction. This makes CNN a suitable and appropriate network for image classification.

2.9 Conclusion

In this chapter, we have explained what deep learning is, showed how it is correlated to the terms AI and machine learning and how it works. Deep learning is implemented using neural network architectures, hence the nomenclature of deep neural networks (DNN). They comprise several layers stacked on top of each other, and learn in a hierarchical mode. It exists comes in different architectures of DNN divided into supervised and unsupervised learning. We also discussed CNNs, belonging to supervised learning, that are mainly used in image classification. CNNs are composed of a several layers where each one has a specific task.. CONV layers, the core of CNNs, learn a set of K kernels, each of size $F \times F$. They are followed by activation layers to get a non-linear transformation. The spatial dimensions of the input are reduced by pooling layers. FC layers end the network, in which the neurons are fully connected to all activations in the preceding layer. They are followed by a softmax classifier to predict the output. Batch normalization and dropout layers could be added to the network to normalize inputs before passing it to the next layer, and to prevent overfitting, respectively. Backpropagation algorithm is mainly used for training CNNs. During training, we compute the loss which is the difference between the predicted labels and the ground truth labels, and the gradient. The weights are thus updated using the gradients to minimize the loss through

forward propagation on a training set. Finally, we learned what image classification is and how it is performed by deep learning. Deep learning classification consists of gathering the dataset, splitting data into training, testing and validation sets, training the network, and at last evaluating the model. CNNs, end-to-end models, are ideal for image classification. They eliminate manual feature extraction. Instead, the network learns the features that can differentiate amongst image classes. In the next chapter, we will present the different CNNs architectures that we have used to classify facial skin diseases images and the results obtained from each one.

Chapter 3

Convolutional neural networks architectures

3.1 Introduction

There are various popular CNN architectures. Generally, most of deep CNNs are composed mainly of basic layers such as a convolutional layer, pooling layer, fully connected or dense layer, and softmax classifier. The different deep networks are generally made up of numerous CONV layers, Maxpooling layers, followed by one or many fully connected layers and finally a softmax layer. The most common networks are LeNet [41], AlexNet [42], VGG [44], and later more advanced networks have been suggested such as GoogLeNet or Inception [43], ResNet [45], DenseNet [49], EfficientNet [51], and many others. In all these networks, the basic components, convolution and pooling, are almost the same, but the difference lies in their topology. VGG, Inception v3, and EfficientNet B0 are some of the most popular architectures widely used for image recognition and classification due to their good performance and availability of use on different applications for object identification tasks.

VGG is seen as a general architecture and considered a baseline for many tasks and datasets outside of ImageNet, the dataset on which the network is trained initially, and widely used for image identification and classification.

Inception v3 is the third edition of Google's Inception CNN. While most popular CNNs work on increasing the depth of the network, Inception models used other techniques to enhance the performance of the network in terms of speed and accuracy. Many versions have been created, where each version is an improvement of the previous.

EfficientNet models are recently developed from Google presenting a new concept in networks scaling to improve the performance, called Compound Scaling. Previous CNN models follow the classic method of arbitrary single dimension scaling and adding more layers. EfficientNet models scale uniformly all dimensions (depth, width, and resolution) using a compound coefficient. They improve the accuracy and the efficiency of the model. In this chapter, we will review the architecture of the different networks that we have used in our thesis.

We will also explain how we can apply these pre-trained networks on ImageNet to classify other images of different categories than those of ImageNet. The chapter ends in a conclusion.

3.2 Visual Geometry Group (VGG)

Visual Geometry Group, or VGG, is a convolutional neural network introduced by K. Simonyan and A. Zisserman [44]. The main contribution of this model is to show that the depth is a significant factor to enhance classification accuracy. They have been widely used for object recognition. The VGG family is distinguished from other CNN architectures previously proposed by the use of small receptive fields of size 3×3 and stride 1 in all CONV layers, and many stacked CONV with ReLU layer sets followed by a pooling operation.

VGG is composed of several sets of 3 x 3 convolutional layers stacked on top of each other of different depth in various architectures, of stride 1 and padding 1, and ReLU activation function followed by a Maxpooling layer of size 2 x 2 with a stride 2. The number of layers in each set increases as we go deeper in the network.

The CONV layers are followed by 3 fully connected layers where the first two layers contain 4096 nodes, and the third one contains 1000 nodes referring to 1000 classes.

The network is ended by a softmax layer for classification (Figure 3.1).

The VGG family includes four architectures: VGG-11, VGG-13, VGG-16, and VGG-19 where the numbers 11, 13, 16, and 19 stand for the number of weight layers in the network. These models follow the general architecture of VGG, they all have 3 FC layers but differ in the number of CONV layers; where VGG-11 has 8 CONV layers, VGG-13 has 10, VGG-16 has 13 and VGG-19 has 16. All CONV layers are followed by ReLU activation function except one network containing Local Response Normalisation (LRN) normalisation [42]. Also, in one of the models, they use 1 x 1 convolution kernels, that act as a linear transformation of the input, followed by ReLU. The architectures of the different models are presented in (Table 3.1).

We will focus on VGG-16 network that is the backbone of the network we have proposed in our work to classify facial skin diseases.

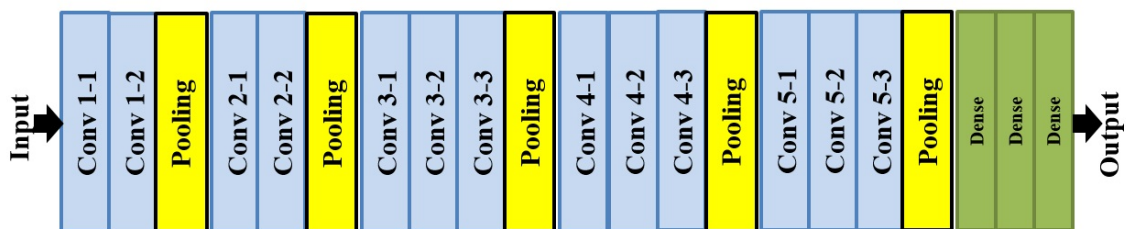


Figure 3.1: VGG architecture. The model includes an input, several stacked CONV layers of different depths, Maxpooling layers, three fully connected layers, and softmax layer performing the classification [44].

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 x 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 3.1: VGG configurations. Each column represents a configuration. The depth increases from left to right from A with 11 weight layers to E with 19 weight layers. The added layers are represented in bold. The CONV layer parameters are designated as conv “receptive field” “width of CONV layers”. Each CONV layer is followed by a ReLU activation function [44].

3.2.1 VGG-16

VGG-16 is considered one of the most important pretrained models for image classification. It has 16 weight layers; 13 CONV layers and 3 FC layers. It includes only CONV layers of fixed kernel size 3 x 3 and a stride of 1 and 2 x 2 Maxpooling

layers of stride 2 inserted between CONV layers to down sample the input size. The network has 138 million parameters and trained on 1.2 million images from Imagenet dataset to classify objects into 1000 different classes. The VGG-16 could classify images with an accuracy of 92.7%. The architecture of VGG-16 is presented in Figure 3.2. The VGG-16 takes as input an image of size $224 \times 224 \times 3$ referring to height \times width \times depth where depth is the number of RGB channels. As mentioned previously, VGG-16 is composed of several blocks of CONV followed by ReLU activation functions and Maxpooling layers. The first block has 2 CONV layers of size $224 \times 224 \times 64$ each one, and a Maxpooling layer reducing the spatial dimension of the images to $112 \times 112 \times 64$. The second is formed of 2 CONV layers of size $112 \times 112 \times 128$ each, and a Maxpooling layer that reduces the volume size to $56 \times 56 \times 128$. The third constitutes of 3 CONV layers; each of size $56 \times 56 \times 256$, followed by a Maxpooling layer that reduces the image size to $28 \times 28 \times 256$. In the fourth block, there are also 3 CONV layers of size $28 \times 28 \times 512$ each and a Maxpooling layer reducing the image to $14 \times 14 \times 512$. The fifth block has 3 CONV layers too of size $14 \times 14 \times 512$ each one, and a Maxpooling layer with $7 \times 7 \times 512$. These sets are followed by two FC layers with 4096 nodes each one, and one FC layer with 1000 nodes and a softmax classifier (Figure 3.2).

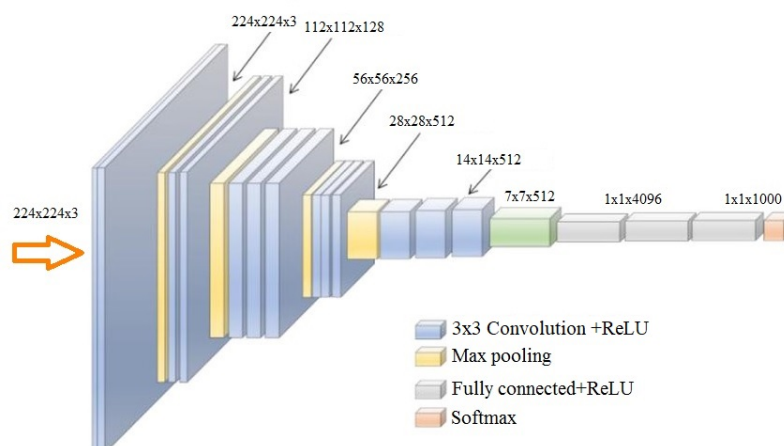


Figure 3.2: VGG16 architecture. The network is composed of 13 convolutional layers with filter size 3×3 , 5 Maxpooling layers embedded between CONV layers, 3 fully connected layers, and softmax layer arranged in a specific architecture. The input is an RGB image of size 224×224 [44].

VGG-16 is trained by optimizing the multinomial logistic regression using mini batch SGD based on back propagation with momentum. They use a batch size of 256, and a momentum of 0.9. The training is regularized by L2 weight decay and dropout for the first two FC layers of 0.5. The learning rate is initially set to 0.01, and when the validation accuracy stops enhancing, it is dropped by factor of 10. VGG-16 has a larger number of parameters and depth than previous deep neural networks, but requires less epochs for loss function to converge and this is due to the use of small convolutional filter sizes and regularization by large depth, and pre-initialization of some layers [44]. The network is trained with images of size 224 x 224. The network has learned plenty of feature representations for a large variety of images. The pre-trained VGG-16 is an open-source model, so it can be generally used out of the box for different applications.

3.3 Inception-v3

Inception-v3 is a CNN architecture from the inception family [51]. It is widely used in image identification and classification, and pre-trained on ImageNet dataset. The Inception module is based on the idea of running several operations of pooling, and convolution with different filter sizes as 3 x 3, and 5 X 5 in parallel. The inception family includes 4 versions. The first version is Inception-v1, known also as GoogleNet [43]. Later, this architecture has been modified and improved by introducing batch normalization resulting in a new model called Inception-v2 [39]. The third generation is Inception-v3 which especially uses factorization ideas. The last version is Inception-v4 which is a simplified architecture involving more inception modules than Inception-v3 [52]. Inception-v3 is built on factorization ideas. Its architecture is built as follows.

3.3.1 Factorized convolutions

Factorizing convolutions aim to diminish the number of parameters in the network without deteriorating the efficiency. It consists of: factorization into smaller convolutions, and factorization into asymmetric convolutions.

a- Factorization into Smaller Convolutions

The benefit of smaller convolutions is to fasten the training. 5×5 filter is factorized into two 3×3 filters. One 5×5 filter has 25 parameters, while two 3×3 filters have 18 parameters ($3 \times 3 + 3 \times 3$). Hence, this technique has reduced the number of parameters by 28%. Using this technique, the Inception Module A has been created (Figure 3.3).

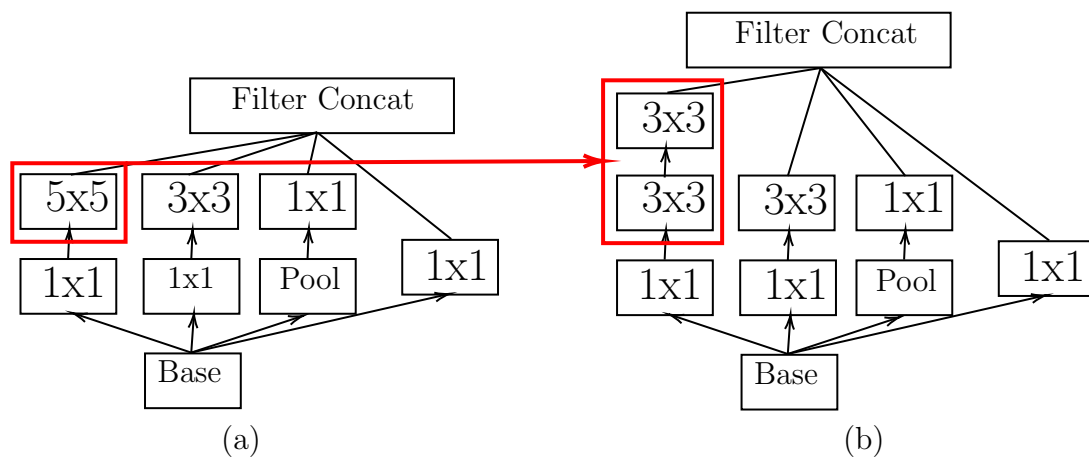


Figure 3.3: Factorized convolutions: (a) original inception module including 5×5 convolutions; (b) new factorized module, called Module A, where 5×5 filter in the original module is replaced by two 3×3 filters. Pool denotes pooling layer, and $m \times n$ denotes a CONV layer, where m and n determine the size of the convolution.

b- Factorization into Asymmetric Convolutions

Asymmetric convolutions consist of factorizing $N \times N$ convolutions into $1 \times N$ and $N \times 1$ convolutions. A 3×3 convolution is replaced by a 1×3 convolution followed by a 3×1 convolution. The 9 parameters resulting from a 3×3 filter are reduced to 6 parameters when using 3×1 and 1×3 filters. The reduction is of 33%. The

3×3 convolution has not been replaced by two 2×2 ones because the number of parameters will be reduced only by 11%, since the new parameters will be 8 ($2 \times 2 + 2 \times 2$). This technique results in two modules Inception Module B and Inception Module C. According to authors, the Inception Module C is developed to enhance large-scale representations (Figure 3.4).

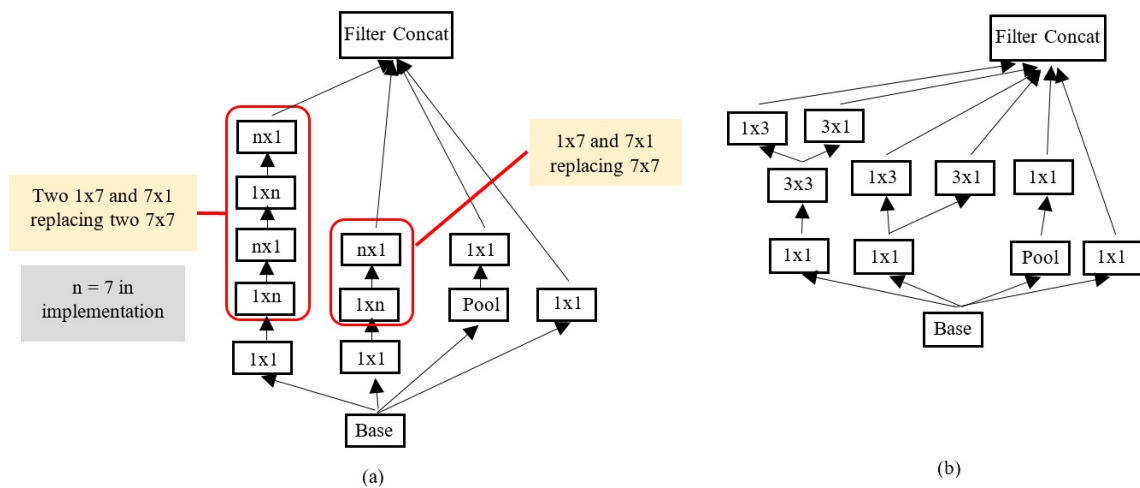


Figure 3.4: Asymmetric Convolutions: (a) Inception Module B where 7×7 filter is replaced by 1×7 followed by 7×1 filters; (b) Inception Module C where 3×3 filter is replaced by 1×3 followed by 3×1 filters. Pool denotes pooling layer, and $m \times n$ denotes a CONV layer, where m and n determine the size of the convolution.

The authors have developed three inception modules based on the idea of factorization, that is a method that allows the reduction of the number of parameters for the entire network; which makes overfitting less likely, and thus the depth of the network can be augmented.

3.3.2 Auxiliary Classifier

Auxiliary Classifiers are small CNNs placed between layers and have different purposes. In Inception-v1, two auxiliary classifiers are used to deepen the network. Whereas, in inception-v3, one classifier is used that act as a regularizer (Figure 3.5).

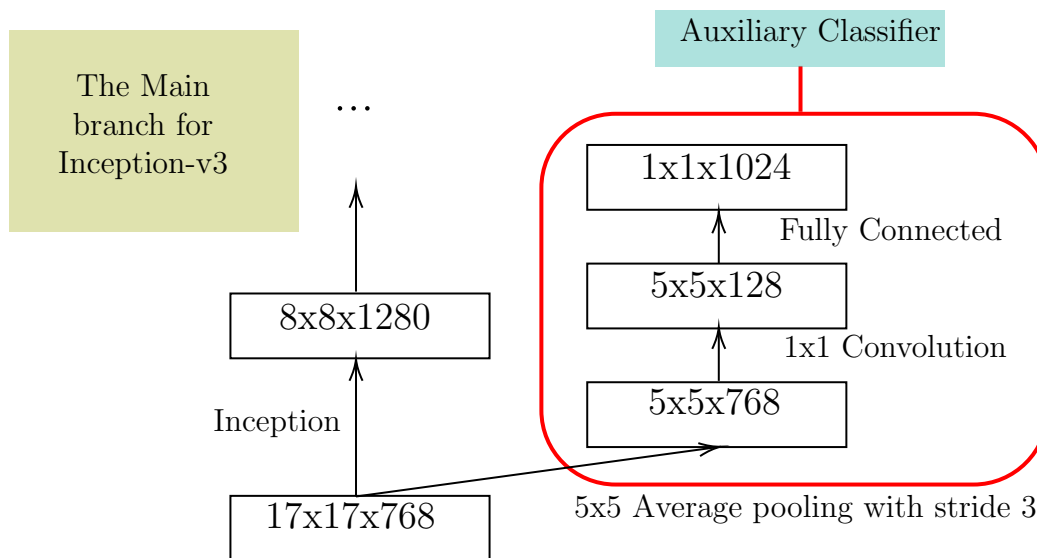


Figure 3.5: Auxiliary Classifier in Inception-v3 acting as regularizer that consists of average pooling, convolution, FC, and softmax layers.

3.3.3 Efficient grid size reduction

The size of the feature maps is usually reduced by max pooling. This method has a computational cost. Hence, an efficient grid size reduction is suggested. 640 feature maps are obtained by concatenation of 320 feature maps that are obtained by convolution with stride 2, and 320 feature maps that are got by max pooling. The concatenated feature maps are presented to the next level of inception module. This method is cheap and cancels the bottleneck representation. It is presented in Figure 3.6. All these concepts presented above are integrated into the final architecture of Inception-v3, consisting of 42 layers. It includes several layers, such as convolution, average pooling, max pooling, concat, dropout, fully connected, and softmax. Batch normalization and ReLU are used after CONV layers. It has 11 inception modules. The general architecture is shown in Figure 3.7. Inception v3 is trained on ImageNet dataset with stochastic gradient using a batch size of 32 for 100 epochs. They use initially momentum with a decay of 0.9, and then used RMSProp with a decay of 0.9, and $\epsilon = 1$ which achieve th best model. The learning rate was set to 0.94, and

decayed after each two epochs with an exponential rate of 0.94.

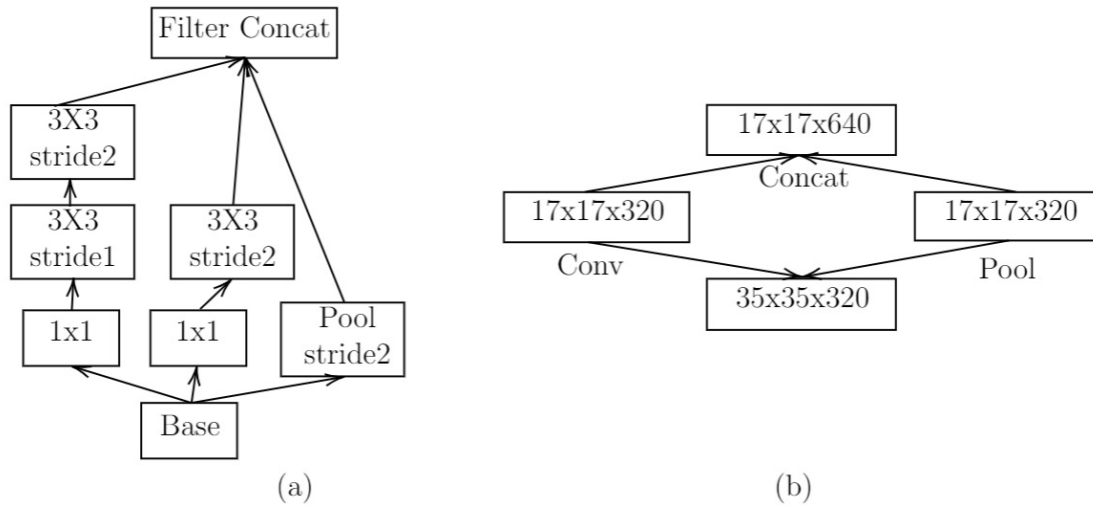


Figure 3.6: Efficient grid size reduction: (a) Detailed architecture of efficient grid size reduction that reduces the grid size while augments the filter banks; (b) Efficient grid size reduction diagram representing the feature maps sizes resulting from each operation.

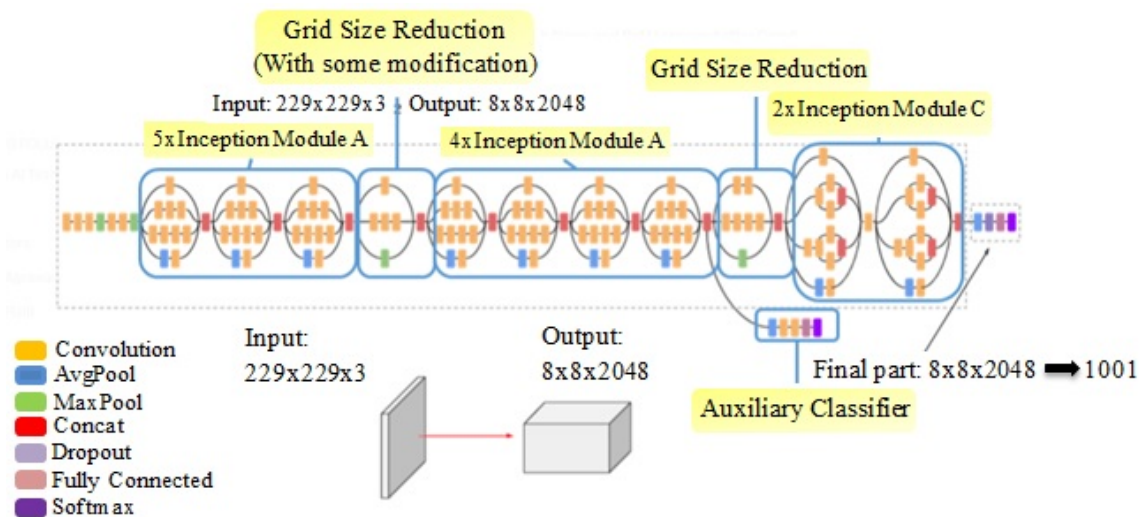


Figure 3.7: Architecture of Inception-v3: Inception-v3 is made up of different type of layers with a specific topology and gathered in blocks, each performing a specific task. It includes 11 Inception modules: 5 Inception modules A, 4 Inception Modules B, and 2 Inception Modules C, 2 Grid Size Reduction blocks, and 1 Auxiliary Classifier. It takes in input an image of size $229 \times 229 \times 3$ [52].

3.4 EfficientNet

EfficientNet is a CNN model proposed by Mingxing Tan and Quoc V in 2019 and scaling method that evenly scales all dimensions of the network, depth, width, and resolution, using a compound coefficient [50]. EfficientNet models achieve better accuracy and efficiency than previous CNNs. Scaling up CNNs is usually used to improve the accuracy. It consists of increasing the model's depth, or width, or image resolution for training and test. Increasing the depth means adding more CONV layers to the network; which allows the extract of more complex features, but the model becomes hard to train. Width scaling; increasing the number of channels in a CONV layer, results in more feature maps. The model captures accurate features but saturates fast. Resolution scaling simply increases the image resolution that is being fed to the CNN. Generally, depth scaling is the most used amongst all. These manual scaling methods improve the accuracy of the model, but they are tedious and also after a few level, the performance saturates or even degrades. A new scaling method is suggested, called compound scaling, based on scaling up the three dimensions simultaneously with a fixed ratio (Figure 3.8). It is justified by the fact that if the input image is bigger, the model requires more layers to increase the receptive field, and more channels to detect fine-grained features in the bigger image. The compound scaling best improves the performance than the individual scaling. The relation between the different scaling dimensions of the baseline network is determined using grid search algorithm [53]. It computes the coefficient of each dimension, and then these coefficients are applied to scale up the baseline network. The three dimensions are scaled uniformly as follows:

$$(\text{Depth} : d = \alpha^\phi); (\text{Width} : w = \beta^\phi); (\text{Resolution} : r = \gamma^\phi) \quad (3.1)$$

With $(\alpha) \cdot (\beta^2) \cdot (\gamma^2) \approx 2, \alpha \geq 1, \beta \geq 1, \gamma \geq 1$. ϕ is a coefficient determined by the user that controls the number of available resources for model scaling, whereas α, β , and γ are constants determined by grid search that define how to designate these resources to the depth, width, and resolution of the network. Based on the compound scaling method and automated machine learning, the researchers have generated a family of EfficientNet models, smaller and faster than state-of-the-art CNN models, achieving up to ten times better accuracy and efficiency.

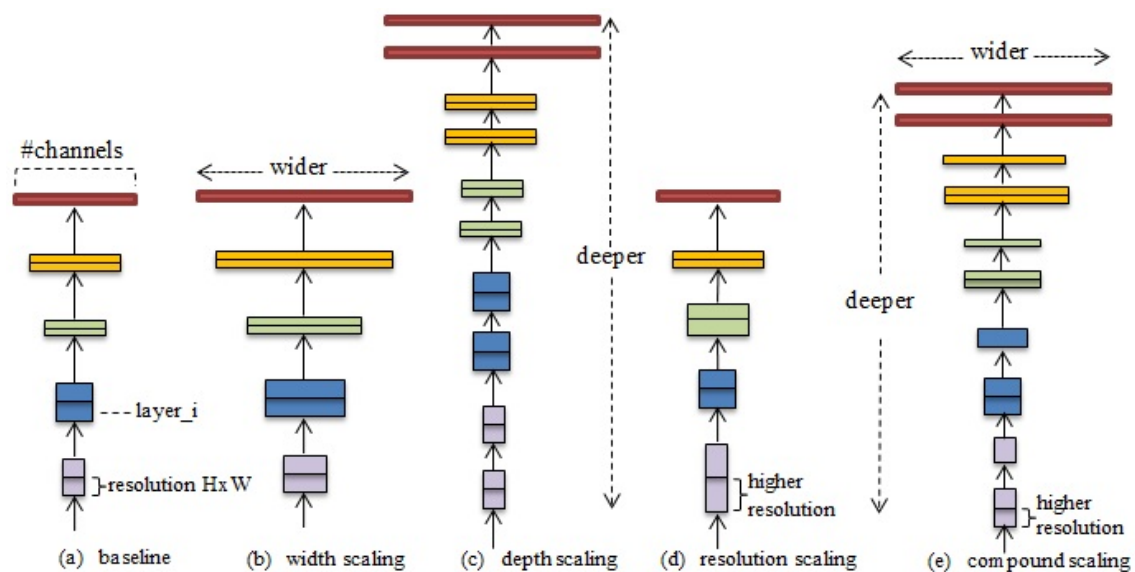


Figure 3.8: Model scaling: (a) baseline network; (b), (c), (d) single dimension scaling method where width, depth, and resolution, are scaled up respectively; (e) compound scaling method increasing depth, width, and resolution, with a fixed ratio.

3.4.1 EfficientNet architecture

Scaling improves accuracy but doesn't change the function of the layers. The first step was finding the baseline network and then scaling it up using compound scaling. The authors have developed a new mobile-size baseline, called EfficientNet B0 that optimizes the accuracy and FLOPS (floating point operations per second) that measure the number of operations needed to run the network model. The architecture of the baseline network is presented in Table 3.2.

EfficientNet B0 is mainly composed of mobile inverted bottleneck MBConv used in MobileNetV2 [54] besides squeeze and excitation block with swish activation function. It has 7 blocks of different settings and 11 million trainable parameters. It is developed to be used in image classification.

Stage	Operator	Resolution	#Channels	#layers
1	Conv3x3	224x224	32	1
2	MB Conv1,k3x3	112x112	16	1
3	MB Conv6,k3x3	112x112	24	2
4	MB Conv6,k5x5	56x56	40	2
5	MB Conv6,k3x3	28x28	80	3
6	MB Conv6,k5x5	14x14	112	3
7	MB Conv6,k5x5	14x14	192	4
8	MB Conv6,k3x3	7x7	320	1
9	MB Conv1x1/Pooling/FC	7x7	1280	1

Table 3.2: EfficientNet B0 baseline network architecture. Each row presents the type of layer in the block, its input resolution, its output channels, and the number of layers in each block [54].

Swish activation is an activation function proposed by Google Brain team for deep networks [55]. It is the multiplication of the input x with the sigmoid activation. The form of the Swish function and its derivative are shown in (Figure 3.9).

$$Swish(x) = x * sigmoid(x) = x * \left(\frac{1}{1 + e^{-x}}\right) \quad (3.2)$$

It is a smooth function that looks like ReLU, but differs in the domain around 0. ReLU changes direction near $x = 0$. It cancels all negative values, and thus all their derivatives are zero, while swish activation twists from 0 toward negative values and then upwards. It is non-monotonic. The authors show that swish surpasses ReLU in deep neural networks. For example, when replaced with ReLU on InceptionResNetV2, the performance of the network has improved by 0.6% on ImageNet dataset.

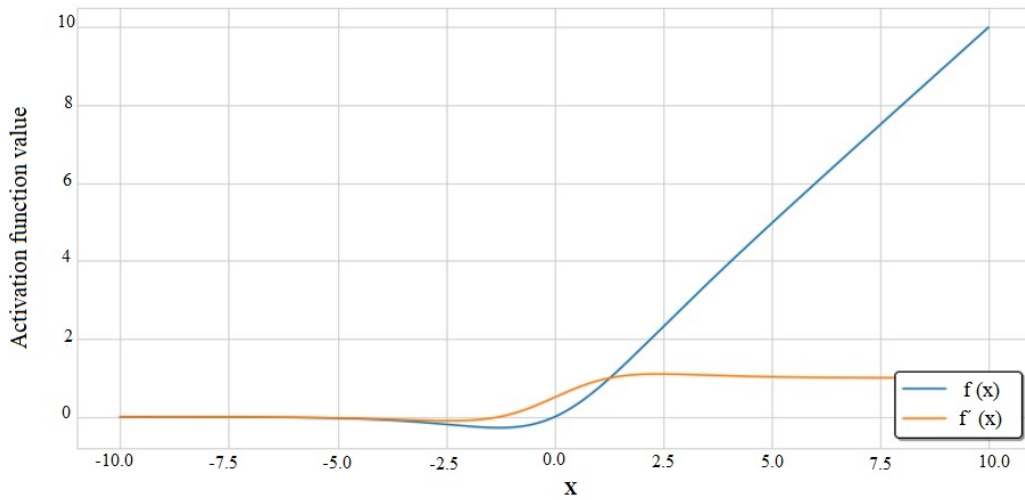


Figure 3.9: The Swish activation function [55].

3.4.2 Inverted Residual Block

Inverted Residual Block, or MBConv Block, is a class of residual block using an inverted architecture for efficiency purposes. It was initially used in MobileNetV2 network [54], and reused in numerous mobile extension networks. Residual blocks create a shortcut connection between the start and end of a convolutional block. They connect the wide layers having the high number of channels. They follow a wide \rightarrow narrow (bottleneck) \rightarrow wide approach. Inverted Residual blocks are the opposite. They have a narrow \rightarrow wide \rightarrow narrow structure, inspired by the fact that bottlenecks include all the significant information. They have less parameters (Figure 3.10).

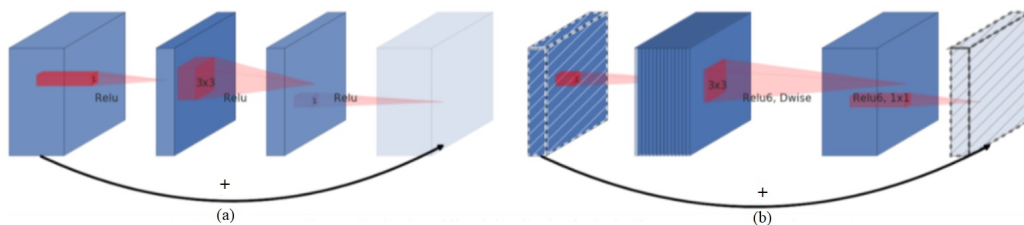


Figure 3.10: A comparison between a residual block and an inverted residual block: (a) A residual block connects wide layers that have large number of channels. Layers in between are bottleneck; (b) An inverted residual block connects bottleneck layers, while layers in the center are wide [54].

3.4.3 Squeeze and Excitation Block

In CNNs, the channels of the feature maps created from a CONV layer have equal weights. Squeeze and excitation (SE) block is an approach that assigns a specific weight to each channel. It focuses on only the significant features. It procures an output of size $1 \times 1 \times \text{channels}$. This weightage is determined by the network as other parameters [56].

SE block, fed with a convolutional block, is composed of a global average pooling layer that squeezes each channel into a single value, an FC layer with C/r (where C is the number of channels, and r is a hyper parameter set to 16) and neurons followed by ReLU activation to reduce the complexity of the output channel. An FC layer with C neurons followed by a sigmoid activation provides a smooth gating function to each channel and scaling where we give a weight to each feature map in the convolutional block depending on the side network. This is called excitation (Figure 3.11).

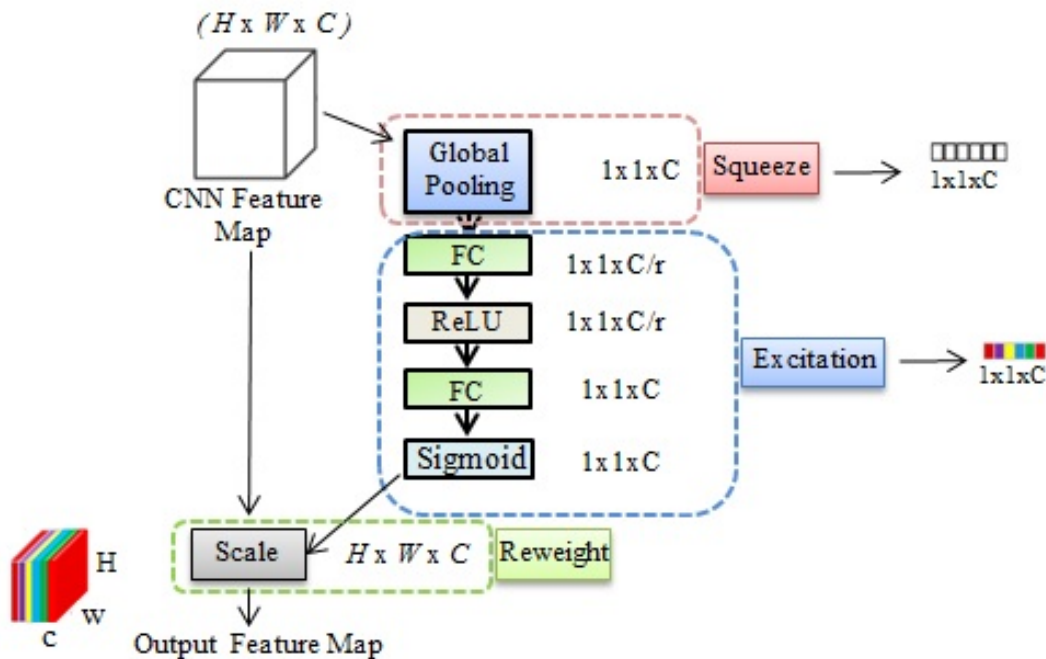


Figure 3.11: SE block structure: Each feature map resulting from CONV layer is squeezed, excited, and then scaled to produce a weighted feature map [56].

Being the baseline network, EfficientNet B0 is scaled up using a compound scaling method at two stages. In the first stage, the authors fix the value of ϕ to 1, and determine the values of α , β , and γ using a small grid method that is set to 1.2, 1.1, and 1.15 respectively for EfficientNet B0. In the second stage, α , β , and γ are fixed as constants, and the base network is scaled up with a different ϕ , to generate EfficientNet B1 to B7. All EfficientNet models include 7 blocks, but with a different number of sub-blocks that increases as we go from EfficientNet B0 to B7. They are all trained on ImageNet database using similar settings, an RMSProp optimizer with decay 0.9 and a momentum 0.9, a batch norm momentum 0.99, a weight decay $1e-5$, and an initial learning rate 0.256 decayed by 0.97 every 2.4 epochs. The dropout is linearly increased from 0.2 to 0.5 for EfficientNet B0 to B7 respectively, since bigger models require more regularization.

3.5 Transfer learning with CNN

3.5.1 What is a pre-trained model?

A pre-trained model is a trained model on a large dataset that has learned to extract efficient features from images and are then used as the starting point to perform a new task. The pre-trained network can be downloaded and used as-is to classify new images. There exists a standard benchmark dataset for image classification in the computer vision and machine learning literature that are used to train the models. The most famous ones are:

- MNIST (Modified National Institute of Standards and Technology dataset) is a dataset constituted of labelled gray scale images of handwritten single digits between 0 and 9. It includes 60,000 training images and 10,000 testing images of size 28×28 .
- CIFAR-10 (Canadian Institute For Advanced Research) dataset consists of

60000 labelled colored images (RGB) of size 32 x 32 referring to 10 classes, including: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each class contains 6000 images; 5000 training images and 1000 test images.

- ImageNet is a large dataset of annotated images for computer vision research. It includes more than 14 million images organized in approximately 22000 categories. It is useful for object recognition and localization, and image classification tasks. This dataset is used in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [57] that aims to train a deep neural network to classify an image into 1000 object classes that we meet in daily life. It uses approximately 1.2 million training images, 50,000 validation images and 100,000 testing images.

3.5.2 Transfer learning

Training Deep CNNs on large datasets from scratch can take days or weeks, and is also computationally expensive due to the depth and number of fully connected nodes in the network. To minimize these huge time and compute resources, and thus accelerate the process of image classification, the model weights of pre-trained networks on benchmark datasets are used as the starting point. These models can be used directly, or implemented in a new model for other classification tasks. This is called transfer learning. Thus, transfer learning is the most common technique used in deep learning, where a model trained on a task is reused as an initialization for the task of interest. It aims to decrease the training time and enhance the performance of the model. Transfer learning is also beneficial when there is not enough training data to train a network from scratch. In Figure 3.12, we show how transfer learning enhances the performance of the network during training.

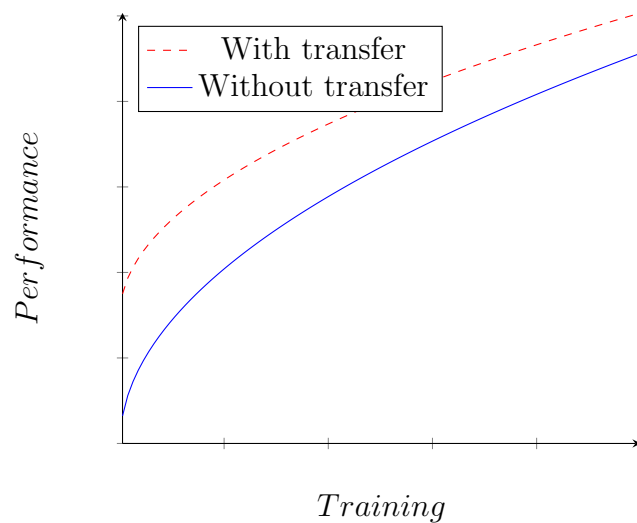


Figure 3.12: Benefits of transfer learning for deep learning with CNN: The y axis represents the performance, and the x axis the number of training samples. When using transfer learning, the performance has a higher start, higher slope, and higher asymptote which leads to a better performance.

In deep learning, there are two types of transfer learning (Figure 3.13):

- CNN is considered as feature extractor. In this case, we remove the output of the pre-trained model, and then use the rest of the network as a fixed feature extractor for the new dataset. The initial layers of the pre-trained model are frozen to retain the contained information during the retraining. A new classifier is added on top of the frozen layers, that will learn to convert old features into predictions on a new dataset. The classifier is retrained on the new dataset.
- Fine-tuning consists of removing the fully-connected layers of the pre-trained network, replacing them by a new FC layer set, retraining them, and also fine-tuning the weights of the pre-trained network. We can fine-tune all the layers of the network, or freeze the weights of initial layers for overfitting concerns and then fine-tune those of higher layers. This is justified by the fact that early layers extract low level features (edges, corners, blobs. . .) that are useful to many tasks, while later layers extract task specific features. The number

of frozen and fine-tuned layers can be determined by testing. The model is retrained on the new dataset with a very low learning rate.

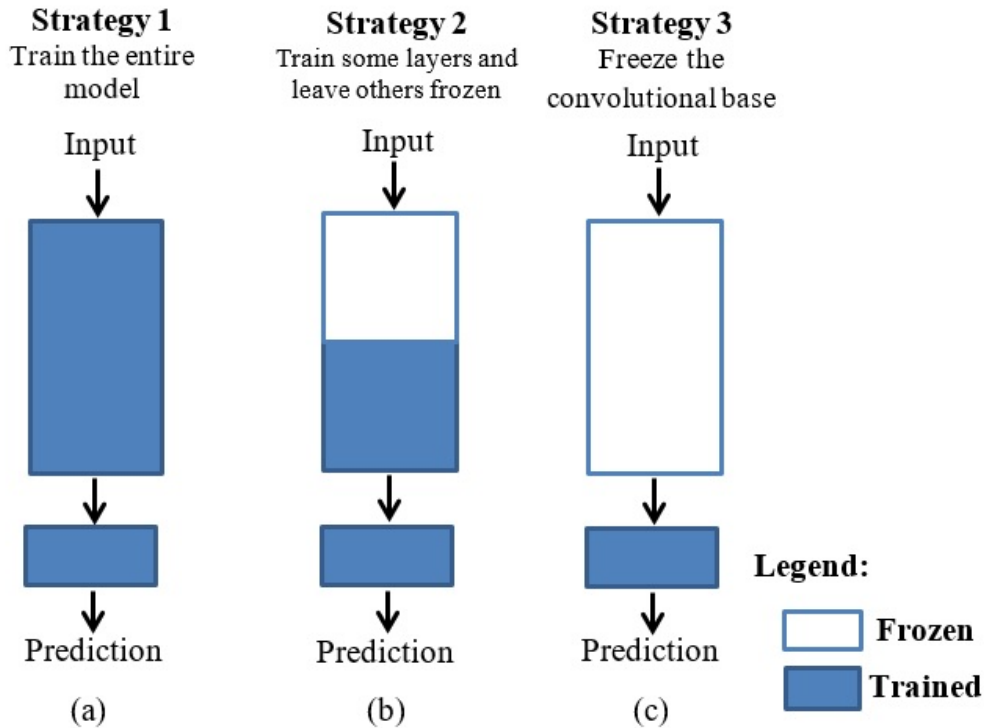


Figure 3.13: Transfer learning scenarios: (a) train the entire model, (b) fine-tuning method: some layers (convolutional, and the classifier) to be retrained, while some to be kept frozen, (c) Feature extractor: applying a new classifier on top of the pre-trained network and retraining it, while freezing the convolutional base.

3.5.3 Transfer learning scenarios

Transfer learning methods depend on several factors, especially the size of the new dataset, and its resemblance to the prototype dataset. There are 4 scenarios that help us to decide which transfer learning method will be used (Figure 3.14).

Scenario 1: Size of the new dataset is small, with a high similarity to the original dataset. In this case, the pre-trained network is used as a feature extractor. We only replace the output layers with a new classifier and retrain it. The convolutional base is frozen. In this case, the fine-tuning is not a good choice because it may lead to overfitting.

Scenario 2: Size of the new dataset is small, with a low similarity to the original dataset. In this case, we use the fine-tuning method. A new classifier is added on top of the pre-trained network. Some convolutional layers are retrained in addition to the classifier, and what remains is frozen.

Scenario 3: Size of the new dataset is large, with a low similarity to the original dataset. The best idea is to train the network from scratch since we have a large dataset, and the new data is different.

Scenario 4: Size of the new dataset is large, with a high similarity to the original dataset. This case requires fine-tuning. The whole model is retrained, using the initial convolutional layer weights as a starting point.

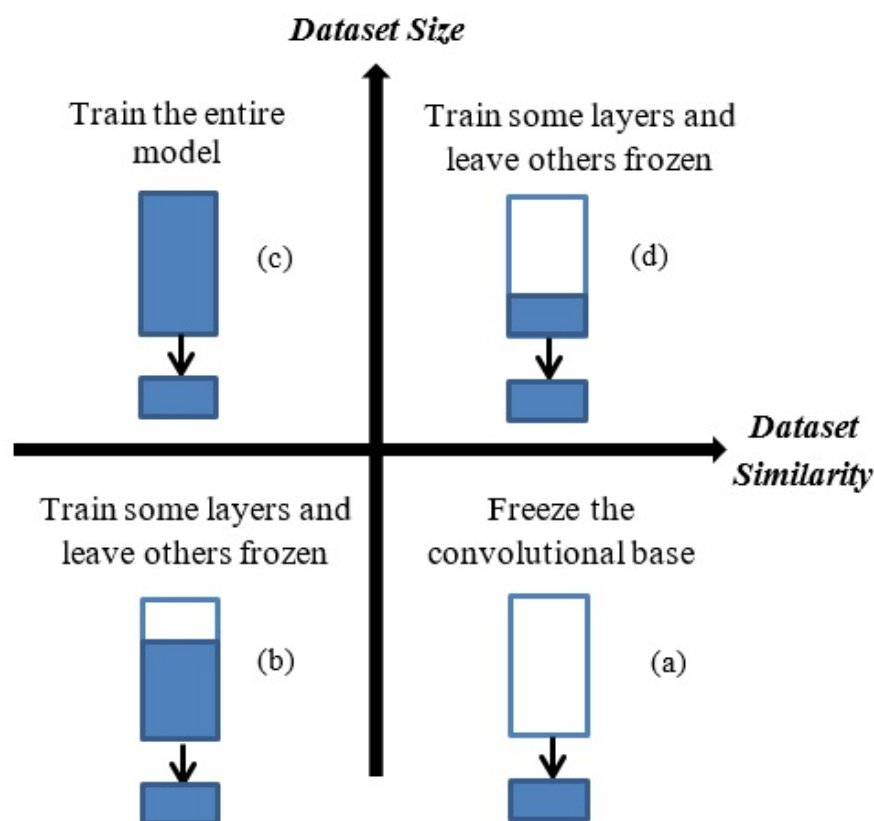


Figure 3.14: Transfer learning methods: There are four scenarios: (a) the new dataset is small, with a high similarity to the original dataset: feature extraction method, (b) the new dataset is small, with a low similarity to the original dataset: fine-tuning method, (c) the new dataset is large, with a low similarity to the original dataset: training from scratch, (d) the new dataset is large, with a high similarity to the original dataset: fine-tuning method.

3.6 Conclusion

In this chapter, we presented an overview of the three CNN models that have been used in our method: VGG, Inception v3, and EfficientNet B0.

VGG is a deep pre-trained ConvNet widely used for image classification and identification. It forms a baseline network for many tasks and datasets beside ImageNet. The main contribution is to show in depth the benefits on the performance of the network. It is characterized by the use of a small receptive field 3×3 of stride 1. VGG has different configurations: VGG-11, VGG-13, VGG-16, and VGG-19. All have the same architecture but differ in the number of layers. VGG-16 is composed of 16 weight layers; 13 CONV layers and 3 FC layers. This model includes CONV layers followed by a ReLU activation function, max pooling layers, and FC layers. It takes as input an RGB image of size $224 \times 224 \times 3$. It is pre-trained on ImageNet ILSVRC dataset and achieves a top-5 test accuracy of 92.7%.

Inception v3 is an optimized version of GoogleNet belonging to the Inception family. Inception models rely on the Inception modules. The contribution is the use of factorization ideas to build the network. It has been developed using different techniques including factorizing convolutions, dimension reduction, and one auxiliary classifier. It has 11 inception modules of 3 kinds; Inception Module A, Inception Module B, and Inception Module C. Inception v3 is made up of 42 layers of different types; CONV, average pooling, max pooling, concat, dropout, FC, and softmax layers forming several blocks having different functions. It takes an image of size $229 \times 229 \times 3$ as input. Inception v3 is pre-trained on ImageNet ILSVRC and reaches a classification accuracy greater than 78.1%. It achieves the lowest error rates with comparison to the state-of-the-art.

EfficientNet models presented a new technique to scale up CNNs to enhance the performance of the network and denoted compound scaling method. It consists of balancing the three dimensions: depth, width, and resolution at one time by scaling

each of them by a constant ratio. EfficientNet B0 is the baseline network of this method on which the introduced scaling method is applied. EfficientNet B0 is scaled from B1 to B7. It is principally made up of MBConv of different settings, as well as a squeeze and excitation block, and a swish activation function gathered in 7 blocks. It is pre-trained on ImageNet dataset and achieves a classification accuracy of 84.4%.

These pre-trained networks can be used to perform image classification on different tasks and datasets than ImageNet. This is called transfer learning. This technique is commonly used in deep learning because it speeds the training time, enhances the performance of the model, and doesn't require a lot of data. The pre-trained model is trained on new data but initialized with pre-trained weights. Also, these pre-trained models can be fine-tuned by making some modifications to the model and then retrain it on the new data using a small learning rate. Using transfer learning or fine-tuning depends on the size of the new dataset, and the similarity with the original dataset.

In the next chapter, we will present the results obtained when applying transfer learning on these three pre-trained networks to classify facial skin diseases images into ten classes, that is 8 facial skin pathologies, normal class, and no-face class.

Chapter 4

Facial Skin Disease Identification AI Based Approach

4.1 Introduction

In this chapter, we will be building a complete end-to-end method that can detect facial skin diseases in RGB images using deep learning along with computer vision techniques. To accomplish this task, we will be using transfer learning with the three pre-trained models VGG-16, Inception v3, and EfficientNet b0 and training them on a dataset of images containing faces of people affected with a skin disease, normal face skin, and no face images. In the remainder of the chapter, we will explain in details our proposed approach. We will also present the results obtained in each model.

4.2 Facial skin diseases identification method

In this study, we proposed an approach that can identify probable dermatological facial pathologies from a single face image. Only the phenotypes of the face are used

in different face poses, illumination and image resolution. Our approach does not require extraction of Region of Interests (ROI). Our method could classify the images into ten classes that are 8 facial skin diseases, normal face class, and no face class. The classification is achieved through three based deep CNN architectures: VGG-16, Inception v3, and EfficientNet b0 while using transfer learning and fine-tuning and has yielded different results. These models were trained and validated using a dataset, containing 20000 facial skin images gathered from various sources, that we have created due to the unavailability of any public database including the identified diseases in our work. The proposed approach is composed of four phases which are gathering and labeling facial skin disease images, image preprocessing including resizing of images and data augmentation, network training and validation, and identification of the lesions. The general block diagram of the method is illustrated in Figure 4.1.

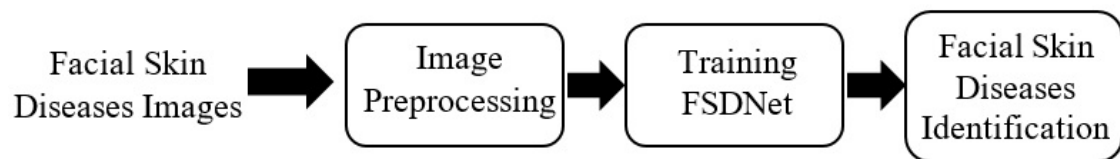


Figure 4.1: Bloc diagram of facial skin diseases identification. Facial skin images are gathered and labelled. They undergo some preprocessing such as resizing, and data augmentation. Then, these images are used to train the network to finally identify the probable diseases.

We will be implementing our models network using the Keras framework with Tensorflow as backend. The system ran on an Intel® core™ i7 7700HQ, 2.8 GHz CPU, 1060 GPU GTX.

4.3 Facial skin disease dataset

Facial skin disease dataset consists of labelled RGB images of faces affected by one of the eight skin diseases used in our approach, normal faces, and images of

various objects such as animals, cups, cans, trees, cars, food, etc. . . collected from different dermatological sites. Initially, the dataset is formed of 2000 labeled images belonging to ten classes that are 8 facial skin diseases (Acne, Actinic keratosis, Angioedema, Blepharitis, Eczema, Melasma, Rosacea, and Vitiligo), normal skin class, and no-face class. All the classes are balanced, and contain 200 images of different resolutions and brightness, for males and females of different age groups, and in several face poses. Deep neural networks have a lot of parameters, so in order to have good performance we should present a proportional amount of data. Since the performance of networks enhances with the amount of data available, we perform data augmentation to increase the size and the diversity of the available data, without having to gather new data. Data augmentation can also help reduce overfitting and enhance the generalization of the model due to the increase in the diversity of our dataset. In our work, the augmentation is performed before feeding the data to the network. It is called offline augmentation. New samples generated by horizontal flip and rotation are used. Images are rotated at different angles (5° , -5° , 10° , -10° , 30° , -30° , 45° , and -45°) (Figure 4.2). Thus, our dataset has in total 20000 images and 2000 images in each class divided randomly into training and validation sets where each image is associated with one of ten labels from 0 to 9 (Figure 4.3). The training dataset is used to train our classifiers to learn what every class looks like while. While the validation set is used to evaluate the model while tuning the model hyper-parameters. We also created a small dataset that includes 20 images in each of the ten classes in our facial skin diseases dataset and is used to assess the performance of our fully trained classifiers. These images are collected from Dermweb [58]. These images are of different resolutions. Since our classifiers take an RGB image of size $224 \times 224 \times 3$, we rescale all the images in our dataset to the appropriate size before feeding them to the network.

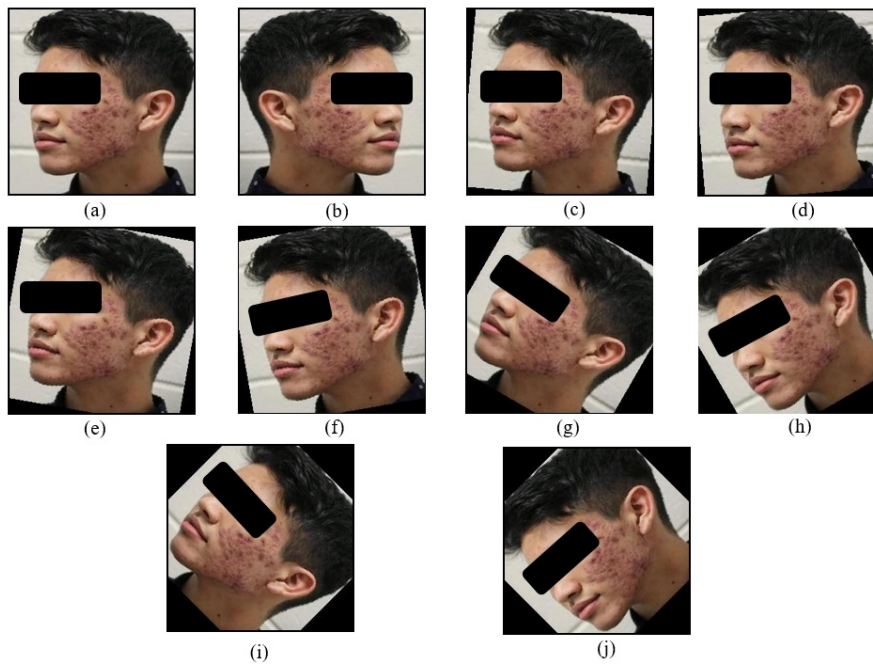


Figure 4.2: Sample images from our database: (a) original image, (b) horizontal flip image, rotated images with angles (c) -5° , (d) 5° , (e) -10° , (f) 10° , (g) -30° , (h) 30° , (i) -45° , (j) 45° .



Figure 4.3: Images from our dataset referring to the ten classes: 8 facial skin diseases (Acne, Actinic Keratosis, Angioedema, Blepharitis, Eczema, Melasma, Rosacea and Vitiligo), normal skin class, and no face class including different images such as animals, cups, coins, phones...

4.4 Facial skin diseases identification using VGG-16

To classify facial skin diseases, we use pre-trained VGG-16 based models. We have applied two models of VGG-16; VGG-16 in its original architecture, and transfer learning with VGG-16 which generates a new adapted architecture to facial skin diseases identification to evaluate performances on Facial Skin Disease Dataset (FSDD). Before diving into our approaches, we will introduce some terminologies that will be seen in the next sections.

1- Overfitting

Overfitting occurs when a network models well the training data, but can't generalize on unseen data. It can be noticed when the validation or testing error is greater than the training error. To prevent overfitting, we can reduce the complexity of the network, or apply regularization techniques such as weight decay, dropout, data augmentation, early stopping, etc..., or train with more data.

Underfitting happens when the network has not learned the basic features, and thus can neither model the training nor generalize to unseen data. Underfitting is remedied by simply using another model. Actually, when we train a network we aim to lessen the training loss as much as we can, and keep the gap between the training and testing loss small.

2- Batch

The batch size is a hyper-parameter of gradient descent, the learning algorithm that updates the network using the training dataset, which defines the number of training samples used in one forward and backward pass. At the end of the batch, the error is computed by comparison of the predictions with the ground truth labels. Consequently, the update algorithm enhances the model.

There are three types of batches:

- Batch mode: the batch size is equal to the size of the dataset.
- Mini-batch mode: $1 < \text{batch size} < \text{size of the dataset}$, and is a number that can be divided into the total dataset size.
- Stochastic mode: the batch size = 1.

3- Epochs

An epoch is the one time forward and backward passage of the whole dataset through the neural network. The number of epochs is a hyper-parameter of gradient descent that determines the number of complete passes of the learning algorithm through the training dataset. This number of epochs is related to the diversity of data, and thus is determined by test. The best number is determined when the model error has been sufficiently minimized. A small number of epochs can lead to underfitting, while a huge number may cause overfitting.

4.4.1 Classification using pre-trained VGG-16

The first step in building our facial skin diseases model is to train the pre-trained VGG-16 on our dataset. The convolutional base and the FC layers are kept the same; we replace the original Softmax classifier by a 10-dimension output vector to fit the number of predicted classes. The original architecture of the pre-trained VGG-16 and the modified architecture for facial skin diseases identification are presented in (Figure 4.4) and (Figure 4.5) . All the layers are frozen during training, and only the new classifier is trained. Hence, among 134 million parameters, we train only 40970 parameters. The pre-trained VGG-16 takes an RGB image of size $224 \times 224 \times 3$ as input. First, the data is loaded into the model and the classes are mapped to an integer label between 0 and 9 (we have 10 classes) and then converted

to one-hot encoding. To train the network, the FSDD data is split into training and validation sets. We assess the performance of the network using 5 split cases 90:10, 80:20, 70:30, 60:40, and 50:50, for training versus validation respectively. The split of data helps to see if the network overfits and if we have to regulate some hyper-parameters such as to lower the learning rate, increase the number of epochs if the validation accuracy is higher than the training accuracy or stop the over-training if the training accuracy varies higher than the validation. We trained the model for 30 epochs with batch size of 16, compiled with categorical crossentropy loss function, and Adam optimizer with a learning rate of 0.0001. The five blocks including CONV2D and Maxpooling2D layers, Flatten layers, and FC layers are kept the same in both architectures. The Dense layer of 1000 nodes is replaced with a 10 dimension one to fit the number of predicted classes in our method.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 256)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 256)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 256)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
Predictions (Dense)	(None, 1000)	4097000

Figure 4.4: Model summary of the original architecture of pre-trained VGG-16.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 256)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 256)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 256)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
Predictions (Dense)	(None, 10)	40970

Figure 4.5: Model summary of the modified architecture adapted to facial skin diseases identification.

The average inference times for training and validating the images in each split case is shown in Table 4.1. After training the model in each split case, we obtain the following training accuracy and loss, as well as validation accuracy and loss presented in Table 4.2 and Table 4.3 respectively.

Split Case	Training Time	Validation Time
90%-10%	4171 s	14 s
80%-20%	4124 s	27 s
70%-30%	4297 s	41 s
60%-40%	4361 s	55 s
50%-50%	4196 s	70 s

Table 4.1: Training and validation times in the five data split cases. The training time in the different cases is approximately stable, while the validation time increases with the number of the validation images.

The accuracy is a metric that characterizes the performance of the model in all classes. It is the ratio of correct predictions to the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Split Case	Training accuracy	Training loss
90%-10%	99.91%	0.0219
80%-20%	99.89%	0.0255
70%-30%	99.81%	0.0289
60%-40%	99.85%	0.0328
50%-50%	99.85%	0.0374

Table 4.2: Training accuracy and loss in the five split cases. The accuracy and loss vary slightly between the split cases. The highest accuracy is 99.91% and the lower loss is 0.0219 in data split case 90:10 for training versus validation.

Split Case	Validation accuracy	Validation loss
90%-10%	96.9%	0.0987
80%-20%	95.77%	0.1198
70%-30%	95.05%	0.1381
60%-40%	94.85%	0.1559
50%-50%	93.06%	0.2003

Table 4.3: Validation accuracy and loss in the five split cases. The validation accuracy varies between 96.9% and 93.06% and the loss between 0.0987 and 0.2003. The highest accuracy and lowest loss are achieved in the split case 90:10 for training versus validation.

We plot the variation of training and validation accuracies through the 10 epochs as well as the training and validation loss, in each data split case in Figure 4.6. In all cases, we observe that the training and validation accuracy are both increasing, and the training and validation loss are decreasing together. Now that we have trained the classifier, we will compute the metric values such as precision, recall, and F-1 score, to evaluate the model performance.

The precision evaluates the accuracy of a model to classify a positive sample. It is the ratio of correct positive samples to the total number of positive predicted samples.

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

The recall determines the ability of a model to detect positive samples. The number of detected positive samples increases proportionally with the recall. It is given by:

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

The F1-score measures the accuracy of a model on a dataset. It assesses the strength and precision of the classifier. A good classifier with a good F1-score gives low false positives and low false negatives. It is comprised between 0 and 1.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (4.4)$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

We calculate these metrics to assess the performance of the classifier in each class for the different split cases shown in Figure 4.7. To see how well our model is performing, and what types of error it is making, we compute the confusion matrix that provides a summary of prediction results. Hence, the confusion matrix is a N x N matrix, where N is the number of predicted classes that reveal how the model is confused while making predictions. We also compute the confusion matrices in the five split cases shown in Figure 4.8. As we can see in the confusion matrices, the correct predicted values are way greater than the wrong predicted ones.

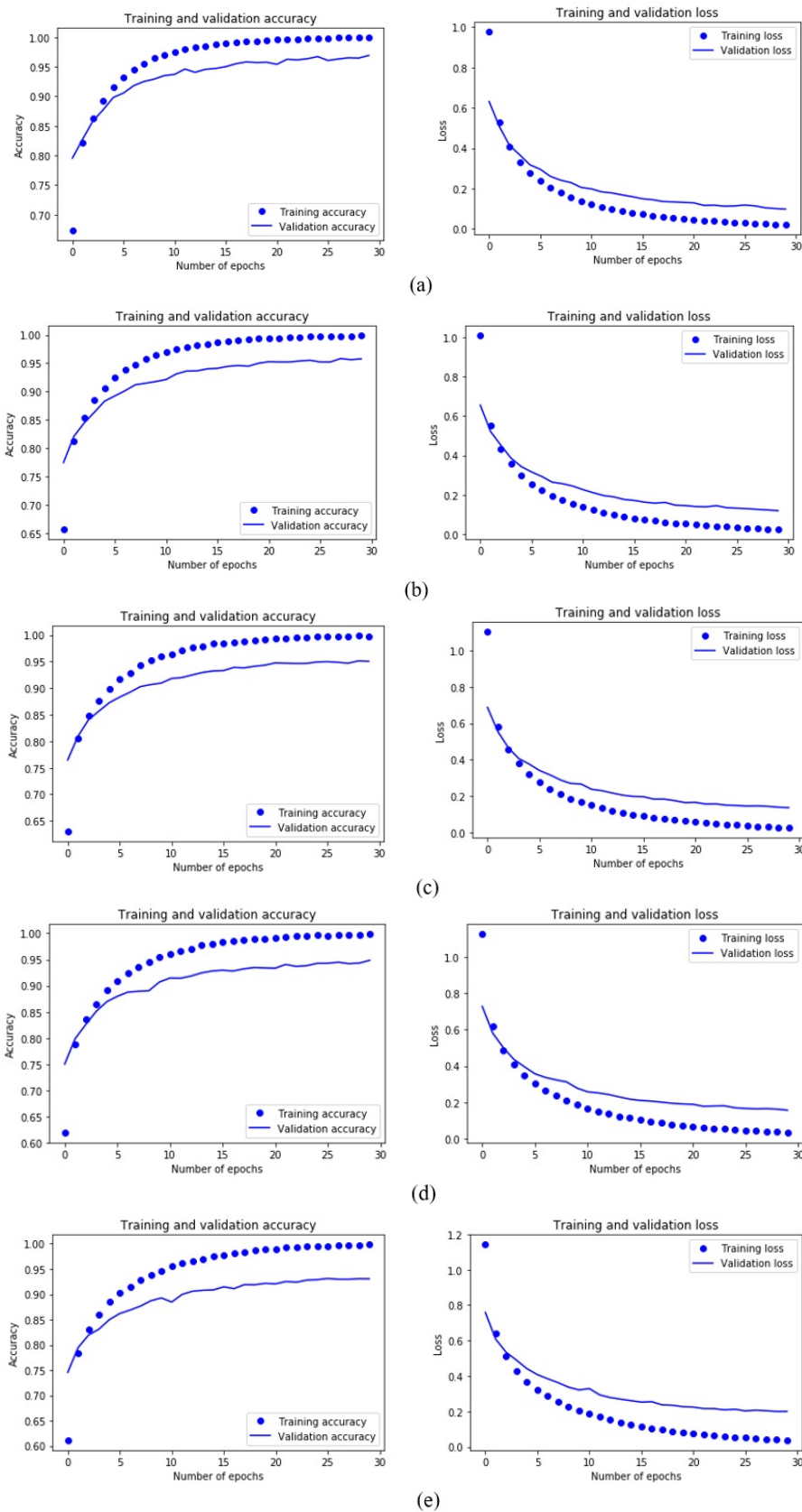


Figure 4.6: Plots of training and validation accuracy, and training and validation loss in the five split cases (a) 90:10,(b) 80:20,(c) 70:30,(d) 60:40,(e) 50:50. We observe that the training and validation accuracy are both increasing, and the training and validation loss are decreasing together.

	Precision	Recal	F1-score	Support		Precision	Recal	F1-score	Support
Acne	0.89	1.00	0.94	204	Acne	0.89	0.99	0.94	407
Actinic	0.97	0.98	0.97	189	Actinic	0.95	0.97	0.96	386
Angiodema	0.98	0.97	0.98	201	Angiodema	0.98	0.95	0.96	390
Blepharitis	0.99	0.99	0.99	196	Blepharitis	0.99	1.00	0.99	414
Eczema	0.95	0.90	0.93	189	Eczema	0.91	0.92	0.92	395
Melasma	0.95	0.90	0.92	213	Melasma	0.94	0.92	0.93	415
Normal	0.98	0.97	0.98	220	Normal	0.97	0.95	0.96	402
No-Face	1.00	1.00	1.00	207	No-Face	1.00	1.00	1.00	418
Rosacea	0.94	0.95	0.95	200	Rosacea	0.93	0.86	0.89	380
Vitiligo	0.99	0.97	0.98	181	Vitiligo	0.98	0.96	0.97	393
macro avg	0.96	0.96	0.96	2000	macro avg	0.95	0.95	0.95	4000
weighted avg	0.96	0.96	0.96	2000	weighted avg	0.95	0.95	0.95	4000

(a)

	Precision	Recal	F1-score	Support		Precision	Recal	F1-score	Support
Acne	0.88	0.99	0.93	620	Acne	0.85	0.99	0.92	809
Actinic	0.95	0.96	0.96	580	Actinic	0.95	0.95	0.95	777
Angiodema	0.97	0.94	0.95	599	Angiodema	0.96	0.92	0.94	803
Blepharitis	1.00	0.99	0.99	609	Blepharitis	0.99	0.99	0.99	816
Eczema	0.91	0.87	0.89	598	Eczema	0.92	0.85	0.88	788
Melasma	0.92	0.94	0.93	613	Melasma	0.91	0.91	0.91	818
Normal	0.96	0.95	0.96	591	Normal	0.96	0.97	0.96	788
No-Face	1.00	1.00	1.00	605	No-Face	1.00	1.00	1.00	803
Rosacea	0.91	0.85	0.88	576	Rosacea	0.91	0.88	0.89	792
Vitiligo	0.97	0.97	0.97	609	Vitiligo	0.97	0.95	0.96	806
macro avg	0.95	0.94	0.95	6000	macro avg	0.94	0.94	0.94	8000
weighted avg	0.95	0.95	0.95	6000	weighted avg	0.94	0.94	0.94	8000

(b)

	Precision	Recal	F1-score	Support
Acne	0.88	0.99	0.93	620
Actinic	0.95	0.96	0.96	580
Angiodema	0.97	0.94	0.95	599
Blepharitis	1.00	0.99	0.99	609
Eczema	0.91	0.87	0.89	598
Melasma	0.92	0.94	0.93	613
Normal	0.96	0.95	0.96	591
No-Face	1.00	1.00	1.00	605
Rosacea	0.91	0.85	0.88	576
Vitiligo	0.97	0.97	0.97	609
macro avg	0.95	0.94	0.95	6000
weighted avg	0.95	0.95	0.95	6000

(c)

	Precision	Recal	F1-score	Support
Acne	0.82	0.99	0.90	1012
Actinic	0.93	0.92	0.93	981
Angiodema	0.94	0.91	0.91	985
Blepharitis	0.98	0.98	0.98	1019
Eczema	0.93	0.78	0.85	1000
Melasma	0.88	0.91	0.89	1018
Normal	0.92	0.94	0.93	975
No-Face	1.00	1.00	1.00	1000
Rosacea	0.88	0.85	0.87	994
Vitiligo	0.96	0.93	0.94	1016
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000

(d)

	Precision	Recal	F1-score	Support
Acne	0.82	0.99	0.90	1012
Actinic	0.93	0.92	0.93	981
Angiodema	0.94	0.91	0.91	985
Blepharitis	0.98	0.98	0.98	1019
Eczema	0.93	0.78	0.85	1000
Melasma	0.88	0.91	0.89	1018
Normal	0.92	0.94	0.93	975
No-Face	1.00	1.00	1.00	1000
Rosacea	0.88	0.85	0.87	994
Vitiligo	0.96	0.93	0.94	1016
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000

(e)

Figure 4.7: Evaluation metrics in different split cases for each class (a) 90:10, (b) 80:20, (c) 70:30, (d) 60:40, (e) 50:50, where we compute for each class the precision, recall, and F1-score. The column support represents the number of test samples available for validation, while macro avg stands for macro average that calculates the average without taking into account the proportion of each class from the total number of samples, and weighted average calculates the average with consideration of the proportion. The performance of the model varies between the classes. Some classes are classified better than the others.

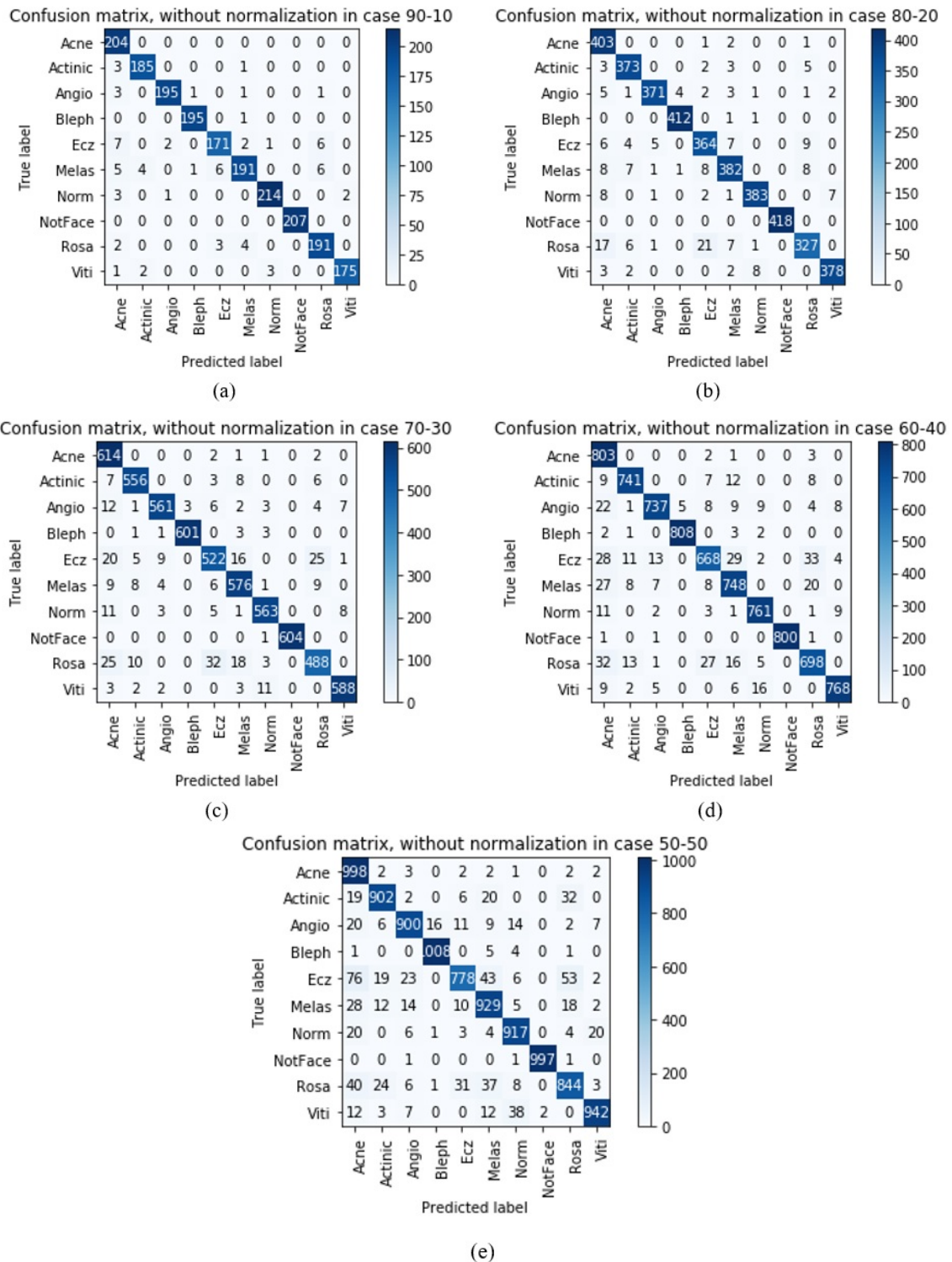


Figure 4.8: Confusion matrices in (a) 90:10, (b) 80:20, (c) 70:30, (d) 60:40, and (e) 50:50 for training versus validation respectively. The vertical axis of the confusion matrix represents the true label belonging to the ten classes (Acne, Actinic keratosis, Angioedema, Blepharitis, Eczema, Melasma, Rosacea, Vitiligo, Normal skin, and no face), and the horizontal axis is the predicted label. The diagonal of the matrix in dark color shows the number of correctly predicted samples while all the other numbers present the wrong predicted samples in each class.

In split case 90% for training and 10% for validation, we found 1928 correct labels and 72 incorrect labels, so 96.4% of the samples are correctly predicted. In case 80:20, the classifier predicts 95.2% of the samples correctly; 3811 correct labels and 189 incorrect labels. In case 70:30, 5673 labels are appropriately predicted equivalent to 94.55% of the samples, while 327 are wrongly predicted. For the split case 60:40, we obtained 7532 correct predictions, and 468 wrong ones, so 94.15% of the predictions are well classified. Finally for the case of 50:50, the model could classify 92.15% of the samples correctly; we have 9215 correct labels, and 785 wrong labels. From all the evaluation metrics presented above, we can observe that the pre-trained VGG-16 model has the best performance when splitting data into 90% for training and 10% for validation, and this is because the model is trained with more data. Once the model is trained and validated, we save the model to test it. For the test, we give the network images from outside the FSDD. For this purpose, we have created a small dataset containing 10 images for each class. By experiment, we have found that the best model to use for test is the split case 90:10.

4.4.2 Classification using transfer learning with VGG-16

The second method we have applied for facial skin diseases classification is transfer learning with VGG-16. We fine-tuned the model to adapt our image classification tasks. The backbone, the CONV base of the pre-trained model, is kept in its original form, while the classifier, consisting of Flatten, 2 FC and softmax layers, is removed. Our classifier is built using a global average pooling layer, dropout layer of 0.5, and a Softmax classifier of dimension 10. This network is called Facial Skin Diseases Network, FSDNet. The architecture of FSDNet is summarized in Figure 4.9. The convolutional base is used to extract the features that will be fed to the classifier that we want to train to identify to which class the facial skin images belong. We choose the global average pooling as classifier for many reasons. The global average

pooling (GAP) is a method that computes the average output of each feature map from the preceding layer. This operation tends to minimize the data considerably, and presents the model for the classification layer. The GAP has no trainable parameters. As seen in the architecture in (Figure 4.9), there are 512 averaging of dimension 7×7 . The GAP layer converts the dimensions from $(7, 7, 512)$ to $(1, 1, 512)$ by doing averaging across the 7×7 channels. By doing this, a large number of trainable parameters is eliminated from the model which speeds up the training. Also the removal of these parameters diminishes the probability of overfitting.

On the other hand, the removal of FC layers forces the feature maps to be more related to the identification classes. Lastly, GAP strengthens the model regarding spatial translations [59]. By experiments, we found that adding a dropout layer after the GAP layer improves the performance of the network. The best results are obtained for a dropout of 0.5. The FSDNet also takes in input as RGB image of dimension $224 \times 224 \times 3$. Data is initially loaded into the model, and the classes are mapped (from 0 to 9) and converted to one-hot encoding. As the aim of transfer learning, CONV weights are frozen which, means that they are set and are not updated during training. The features learned by the pre-trained network are preserved and can be useful for several computer vision issues, even when dealing with totally different categories than those of the initial task. You can keep all of them frozen or freeze some and train the others.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 256)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 256)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 256)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_1	(None, 512)	0
dropout_1(Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5130

Figure 4.9: FSDNet architecture. The convolutional base is the same as in the original architecture of the pre-trained VGG-16. The classifier consists of a global average pooling, a dropout, and a 10-dimension softmax denoted by dense_1.

Actually in our work, we choose to fine tune our model by training some layers and leaving the others frozen because this process provides a better accuracy than keeping all CONV weights frozen. Moreover, lower layers extract general features, while deeper layers learn the specific features of the problem. For facial skin diseases identification, we found that unfreezing only the fifth block is sufficient to obtain a good accuracy. The FSDNet is trained and validated using FSDD. The dataset is split into training and validation sets. We have used 5 split cases 90:10, 80:20, 70:30, 60:40, and 50:50, for training versus validation respectively. Many trials have been carried out to determine the corresponding number of epochs and batch size.

We have trained our model for 30 epochs using different batch sizes and compared their performances to decide the best configuration. Adam optimizer with a learning rate of 0.0001 is utilized. We have found that the model fits well in just 10 epochs for all batch sizes (Figure 4.10), so we have decided to train our model for 10 epochs to gain time. We have tried different batch sizes, 16, 32, and 64 and compare their performance to determine which value to be adopted in our model. The training and validation loss curves in each case are shown in Figure 4.11. Analyzing these curves, we have found that the training and validation loss in case of batch size 16 starts with the minimum loss value. Also, the training loss decreases the faster and this case provides the best validation accuracy. Hence, in our approach we have adopted the batch size 16. The weights from block 5 are trained for 10 epochs because we found that the model could fit very well in 10 epochs with a batch size of 16, compiled with a categorical crossentropy loss function, and Adam optimizer with a learning rate of 0.0001. We chose a small learning rate to preserve former knowledge, and avoid distorting the weights too early and too considerably.

By experiment, we have found that Adam optimizer works better than SGD especially when we test our model. Adam was more efficient and robust and could predict more correct labels.

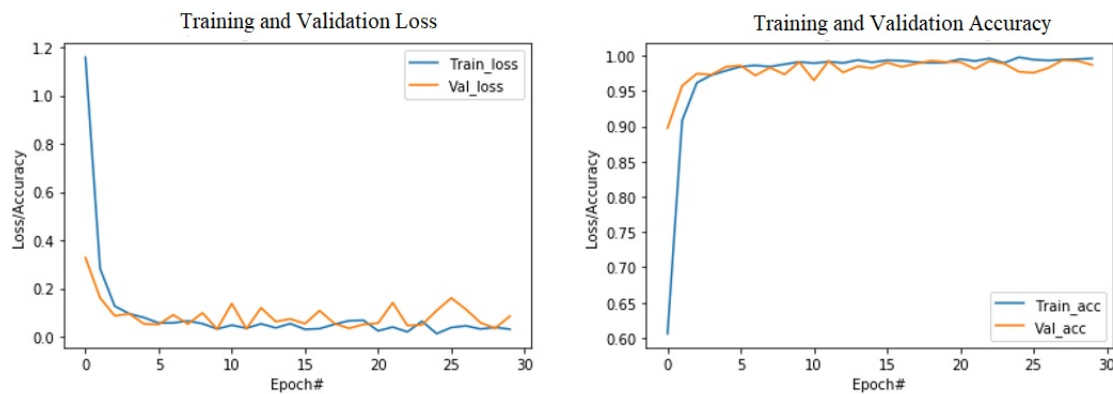


Figure 4.10: Training and validation accuracies and loss for 30 epochs. The training accuracy and loss are almost stable after 10 epochs. They do not vary significantly.

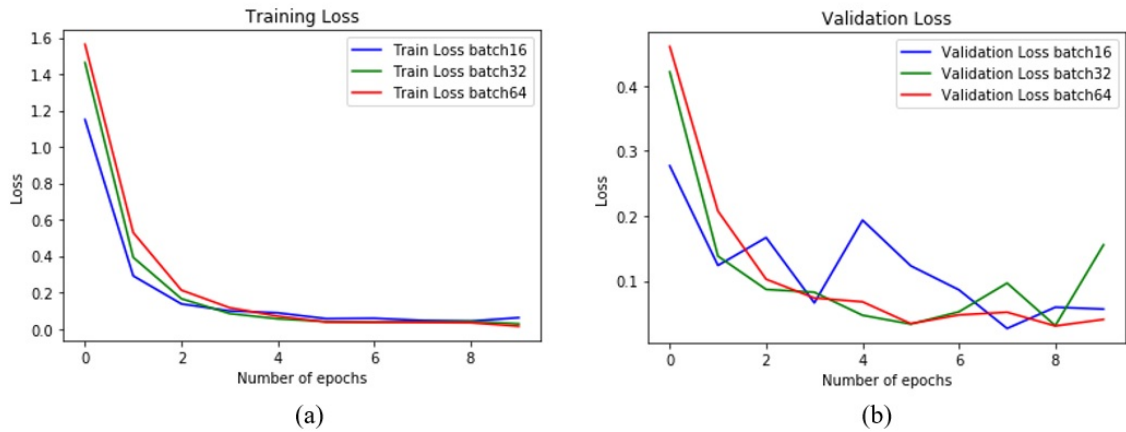


Figure 4.11: (a) Training loss, and (b) validation loss in different batch sizes 16, 32, and 64.

The average inference times for training and validating the images in each split case is shown in Table 4.4. The FSDNet is trained and achieves the following training and validation accuracies and loss as shown in Table 4.5.

Split Case	Training Time	Validation Time
90%-10%	1486 s	13 s
80%-20%	1559 s	26 s
70%-30%	1519 s	40 s
60%-40%	1484 s	52 s
50%-50%	1454 s	66 s

Table 4.4: Training and validation times in the five data split cases. The time required to train and validate the weights of block 5 and the new added layers added in the top of the network varies between the split cases.

Split	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
90%-10%	99.04%	0.0451	98.95%	0.0582
80%-20%	98.58%	0.0611	98.57%	0.0567
70%-30%	98.09%	0.0868	98.56%	0.0557
60%-40%	98.99%	0.0431	98.087%	0.0955
50%-50%	98.62%	0.0489	96.2%	0.1377

Table 4.5: Training and validation accuracies and loss in the five split cases. The accuracy and loss vary slightly between the split cases. The training accuracy varies between 98.09% and 99.04%, while the lowest loss is 0.0431 obtained in split case 60:40. The validation accuracies are smaller and are between 96.2% and 98.95% but the loss values are higher than in training, and the lowest loss is 0.0557. For training and validation, the highest accuracies are obtained in case of splitting 90% of data for training and 10% for validation.

One observes that the FSDNet achieves better performance than the pre-trained VGG-16 with an accuracy of 99.04% for 96.95%. We have plotted the learning curves (accuracy and loss) for the five split cases to see the changes in performance of the model over time and thus determine if the model overfit, underfit or well fit to data. The curves are presented in Figure 4.12. Comparing to the curves of the pre-trained VGG-16 model, we observe that the learning curves in FSDNet are improved considerably. The gap between training and validation loss has almost disappeared. Even the training and validation accuracy are very close to each other. We can say the model is well fitted to the data. To make sure that the model performs well, we have calculated the evaluation metrics, precision, recall, and F1- score. The results are presented in Figure 4.13. Also with comparison to the pre-trained VGG-16, the metrics have considerably improved. The averages have risen by approximately 8%. To evaluate the performance of the network, and find the errors it makes, we have calculated the confusion matrices presented in Figure 4.14. These matrices show that FSDNet has high number of correct predicted labels. The confusion with other classes is very low and in some cases it is null. The number of correct predicted labels has increased by 9%. The number of correct and incorrect predicted labels is presented in Table 4.6.

Split	Correct labels	Incorrect labels	Total number of classified images
90%-10%	1979	21	2000
80%-20%	3939	61	4000
70%-30%	5898	102	6000
60%-40%	7830	170	8000
50%-50%	9580	420	10000

Table 4.6: Number of correct and incorrect predicted labels in the different split cases of FSDNet. The number of correct labels forms approximately 98% of the total images to be classified in all cases.

The next step is to test the FSDNet using the test dataset, composed of 100 images with 10 images in each class, used in testing the pre-trained VGG-16. According to the carried experiments, we will use the FSDNet in split case 80:20 to predict the class of the different test images.

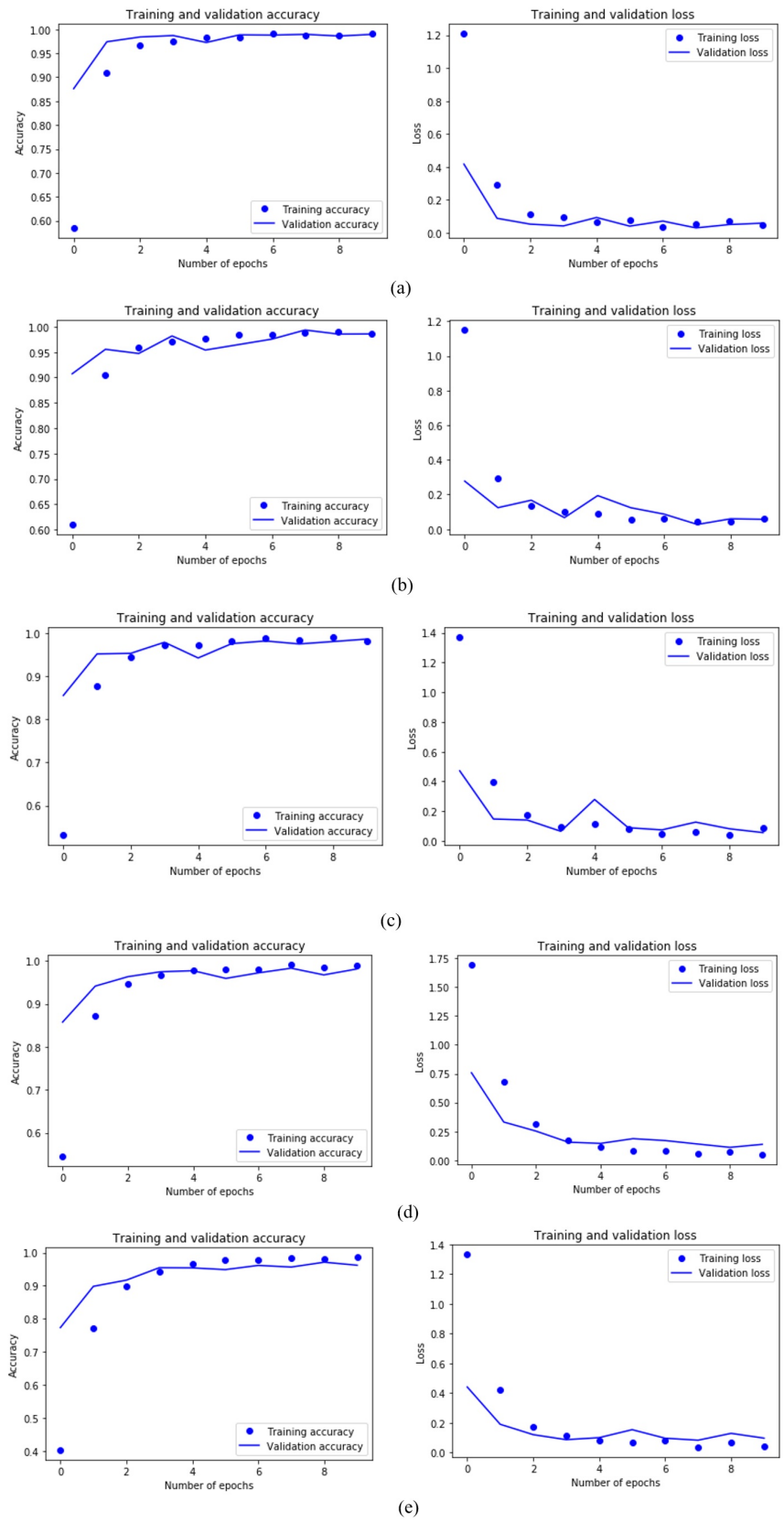


Figure 4.12: Training and validation accuracy, and training and validation loss curves in the five split cases (a) 90:10, (b) 80:20, (c) 70:30, (d) 60:40, (e) 50:50. In all cases, we observe that the training and validation accuracy are both increasing, and the training and validation loss are decreasing together.

	Precision	Recal	F1-score	Support		Precision	Recal	F1-score	Support
Acne	0.99	1.00	0.99	204	Acne	0.97	1.00	0.99	407
Actinic	0.97	0.99	0.98	189	Actinic	1.00	0.95	0.98	386
Angiodema	0.99	0.99	0.99	201	Angiodema	0.99	0.99	0.99	390
Blepharitis	1.00	1.00	1.00	196	Blepharitis	1.00	1.00	1.00	414
Eczema	0.99	0.95	0.97	189	Eczema	0.94	0.98	0.96	395
Melasma	1.00	0.98	0.99	213	Melasma	0.98	0.98	0.98	415
Normal	0.99	1.00	1.00	220	Normal	0.99	0.99	0.99	402
No-Face	1.00	1.00	1.00	207	No-Face	1.00	1.00	1.00	418
Rosacea	0.97	0.99	0.98	200	Rosacea	0.99	0.96	0.97	380
Vitiligo	1.00	0.99	1.00	181	Vitiligo	0.99	0.99	0.99	393
macro avg	0.99	0.99	0.99	2000	macro avg	0.99	0.98	0.98	4000
weighted avg	0.99	0.99	0.99	2000	weighted avg	0.99	0.98	0.98	4000

(a) (b)

	Precision	Recal	F1-score	Support		Precision	Recal	F1-score	Support
Acne	0.96	1.00	0.98	620	Acne	0.94	1.00	0.97	809
Actinic	0.95	0.96	0.96	580	Actinic	0.99	0.99	0.99	777
Angiodema	0.98	1.00	0.99	599	Angiodema	0.96	0.92	0.94	803
Blepharitis	0.99	0.98	0.98	609	Blepharitis	0.99	0.97	0.98	816
Eczema	0.98	0.96	0.97	598	Eczema	0.96	0.97	0.96	788
Melasma	0.99	0.96	0.98	613	Melasma	1.00	0.93	0.96	818
Normal	0.96	0.99	0.98	591	Normal	0.97	1.00	0.98	788
No-Face	1.00	1.00	1.00	605	No-Face	1.00	0.98	0.99	803
Rosacea	0.99	0.97	0.98	576	Rosacea	0.98	0.97	0.97	792
Vitiligo	0.99	0.99	0.99	609	Vitiligo	0.98	0.98	0.98	806
macro avg	0.98	0.98	0.98	6000	macro avg	0.98	0.98	0.98	8000
weighted avg	0.98	0.98	0.98	6000	weighted avg	0.98	0.98	0.98	8000

(c) (d)

	Precision	Recal	F1-score	Support
Acne	0.90	0.99	0.94	1012
Actinic	0.99	0.87	0.93	981
Angiodema	0.98	0.91	0.94	985
Blepharitis	0.98	0.99	0.98	1019
Eczema	0.97	0.93	0.95	1000
Melasma	0.93	0.98	0.95	1018
Normal	0.97	0.99	0.98	975
No-Face	0.99	0.98	0.99	1000
Rosacea	0.88	0.99	0.93	994
Vitiligo	0.99	0.96	0.98	1016
macro avg	0.96	0.96	0.96	10000
weighted avg	0.96	0.96	0.96	10000

(e)

Figure 4.13: Classification report in different split cases for each class (a) 90:10, (b) 80:20, (c) 70:30, (d) 60:40, (e) 50:50, where the precision, recall, and F1-score are calculated to assess the model. The column support represents the number of test samples available for validation, macro avg stands for macro average that calculates the average without taking into account the proportion of each class from the total number of samples, and weighted average calculates the average with consideration of the proportion. The classifier could identify some classes with higher accuracy than others.

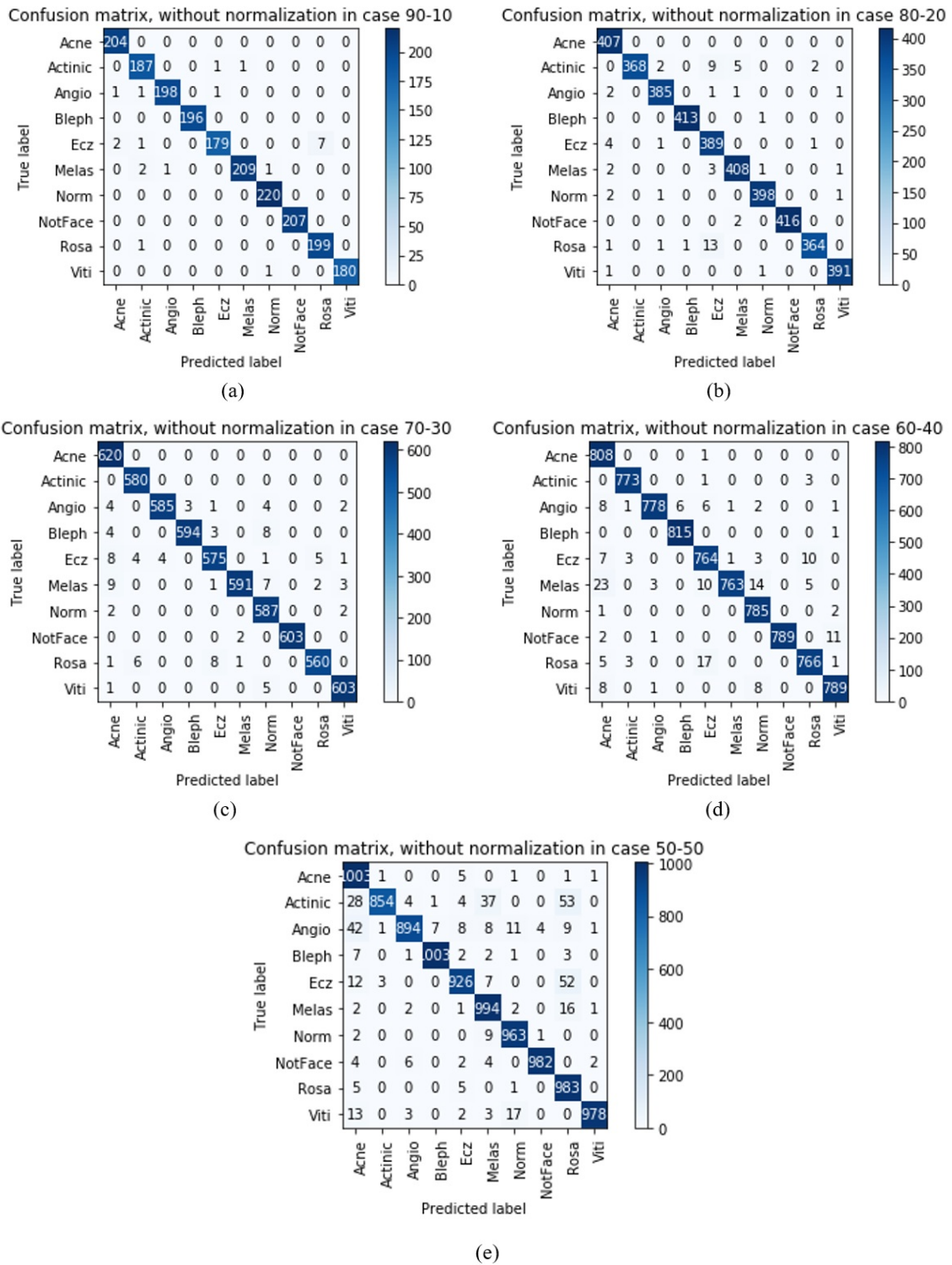


Figure 4.14: Confusion matrices of FSDNet in (a) 90:10, (b) 80:20, (c) 70:30, (d) 60:40, and (e) 50:50 for training versus validation respectively. The vertical axis of the confusion matrix represents the true label belonging to the ten classes (Acne, Actinic keratosis, Angioedema, Blepharitis, Eczema, Melasma, Rosacea, Vitiligo, Normal skin, and no face), and the horizontal axis is the predicted label. The diagonal of the matrix in dark color shows the number of correctly predicted samples and all the other numbers present the wrong predicted samples in each class.

4.4.3 Test

The pre-trained VGG-16 and the FSDNet have been tested using the same images. As described previously, 10 images from each class not included in the FSDD (not used for training nor validation), were used for test. In the following tables, we will present the results and the accuracy of prediction of images in each class.

Class: Acne

Acne images	Predicted class (FSDNet)	Predicted class (Pretrained VGG16)
Image 1	Eczema: 59.9194%	Angioedema: 95.1791%
Image 2	Acne: 98.9879%	Rosacea: 63.3976%
Image 3	Acne: 99.7595%	Angioedema: 99.1347%
Image 4	Acne: 100.0000%	Acne: 71.1650%
Image 5	Acne: 100.0000%	Acne: 58.0859%
Image 6	Acne: 99.9997%	Acne: 53.2475%
Image 7	Acne: 100.0000%	Acne: 86.9358%
Image 8	Acne: 100.0000%	Acne: 67.6664%
Image 9	Acne: 100.0000%	Acne: 78.3595%
Image 10	Acne: 100.0000%	Acne: 99.7977%

Class: Actinic keratosis

Actinic images	Predicted class (FSDNet)	Predicted class (Pretrained VGG16)
Image 1	Actinic: 99.9976%	Actinic: 96.9296%
Image 2	Actinic: 99.8859%	Rosacea: 99.4904%
Image 3	Rosacea: 99.9820%	Actinic: 99.5998%
Image 4	Acne: 48.0390%	Actinic: 67.7229%
Image 5	Actinic: 70.1654%	Rosacea: 56.9207%
Image 6	Actinic: 100.0000%	Rosacea: 99.0967%
Image 7	Actinic: 98.6569%	Actinic: 54.1970%
Image 8	Actinic: 88.5610%	Actinic: 99.7013%
Image 9	Actinic: 99.6006%	Rosacea: 77.0670%
Image 10	Actinic: 99.9916%	Actinic: 94.7248%

Class: Angioedema

Angioedema images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Angioedema: 100.0000%	Angioedema: 99.8815%
Image 2	Angioedema: 100.0000%	Angioedema: 65.5447%
Image 3	Angioedema: 99.2950%	Angioedema: 73.3742%
Image 4	Angioedema: 99.9999%	Normal: 51.7840%
Image 5	Angioedema: 100.0000%	Angioedema: 72.1420%
Image 6	Angioedema: 100.0000%	Melasma: 76.0969%
Image 7	Angioedema: 99.9995%	Angioedema: 94.6091%
Image 8	Angioedema: 99.9820%	Melasma: 76.5969%
Image 9	Angioedema: 100.0000%	Angioedema: 74.4274%
Image 10	Angioedema: 100.0000%	Melasma: 73.3384%

Class: Blepharitis

Blepharitis images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Blepharitis: 100.0000%	Blepharitis: 97.9182%
Image 2	Blepharitis: 100.0000%	Blepharitis: 99.7507%
Image 3	Blepharitis: 99.2950%	Blepharitis: 73.1964%
Image 4	Blepharitis: 99.9999%	Rosacea: 77.7452%
Image 5	Eczema: 99.7191%	Rosacea: 74.5383%
Image 6	Blepharitis: 100.0000%	Blepharitis: 99.8609%
Image 7	Blepharitis: 100.0000%	Blepharitis: 98.8120%
Image 8	Blepharitis: 100.0000%	Blepharitis: 99.9990%
Image 9	Blepharitis: 100.0000%	Blepharitis: 99.7662%
Image 10	Blepharitis: 100.0000%	Blepharitis: 95.9965%

Class: Rosacea

Rosacea images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Rosacea: 100.0000%	Rosacea: 99.2495%
Image 2	Rosacea: 100.0000%	Rosacea: 99.9424%
Image 3	Actinic: 89.6260%	Rosacea: 84.1789%
Image 4	Rosacea: 91.1533%	Rosacea: 97.7397%
Image 5	Rosacea: 98.5187%	Rosacea: 99.5054%
Image 6	Rosacea: 92.1533%	Rosacea: 98.7897%
Image 7	Rosacea: 100.0000%	Rosacea: 99.8428%
Image 8	Rosacea: 99.9768%	Rosacea: 99.5510%
Image 9	Rosacea: 99.9999%	Acne: 25.4341%
Image 10	Rosacea: 98.0187%	Rosacea: 97.4251%

Class: Vitiligo

Vitiligo images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Vitiligo: 91.5769%	Vitiligo: 64.9133%
Image 2	Vitiligo: 93.5769%	Normal: 90.0509%
Image 3	Vitiligo: 92.5769%	Vitiligo: 75.3988%
Image 4	Vitiligo: 95.5769%	Angioedema: 97.8977%
Image 5	Eczema: 92.3289%	Acne: 45.3988%
Image 6	Vitiligo: 88.3836%	Vitiligo: 68.9892%
Image 7	Vitiligo: 86.3836%	Angioedema: 99.9705%
Image 8	Vitiligo: 99.9998%	Vitiligo: 76.2728%
Image 9	Vitiligo: 95.5530%	Vitiligo: 79.0362%
Image 10	Vitiligo: 99.9671%	Vitiligo: 69.2660%

Class: Eczema

Eczema images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Eczema: 99.1136%	Eczema: 78.9951%
Image 2	Eczema: 95.0109%	Eczema: 92.2045%
Image 3	Eczema: 93.7058%	Vitiligo: 54.9187%
Image 4	Eczema: 94.3913%	Acne: 44.0073%
Image 5	Eczema: 99.9118%	Acne: 40.5535%
Image 6	Eczema: 100.0000%	Eczema: 77.2535%
Image 7	Eczema: 98.6569%	Rosacea: 52.0459%
Image 8	Eczema: 98.5610%	Acne: 86.4012%
Image 9	Acne: 25.6425%	Eczema: 77.5904%
Image 10	Acne: 44.2101%	Eczema: 75.4175%

Class: Melasma

Melasma images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Melasma: 91.6315%	Melasma: 94.9185%
Image 2	Melasma: 100.0000%	Rosacea: 89.8135%
Image 3	Melasma: 100.0000%	Melasma: 98.3580%
Image 4	Melasma: 100.0000%	Melasma: 99.8119%
Image 5	Melasma: 100.0000%	Acne: 33.4502%
Image 6	Melasma:99.9371%	Acne: 45.8840%
Image 7	Melasma: 83.7741%	Normal: 53.7066%
Image 8	Melasma: 100.0000%	Melasma: 96.8072%
Image 9	Melasma: 97.7482%	Melasma: 75.6706%
Image 10	Melasma: 99.8925%	Melasma: 75.6920%

Class: No-Face

No-Face images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	No-Face: 100.0000%	Angioedema: 68.5342%
Image 2	No-Face: 100.0000%	No-Face: 99.9086%
Image 3	No-Face: 100.0000%	No-Face: 99.9995%
Image 4	No-Face: 100.0000%	No-Face: 91.1643%
Image 5	No-Face: 100.0000%	No-Face: 99.9999%
Image 6	No-Face: 100.0000%	No-Face: 100.0000%
Image 7	No-Face: 100.0000%	No-Face: 99.9999%
Image 8	No-Face: 100.0000%	No-Face: 100.0000%
Image 9	No-Face: 100.0000%	No-Face: 100.0000%
Image 10	No-Face: 100.0000%	No-Face: 100.0000%

Class: Normal

Normal images	Predicted class(FSDNet)	Predicted class(Pretrained VGG16)
Image 1	Normal: 100.0000%	Normal: 87.1288%
Image 2	Normal: 100.0000%	Normal: 98.8503%
Image 3	Normal: 98.8923%	Normal: 82.7673%
Image 4	Normal: 93.4639%	Normal: 98.1720%
Image 5	Normal: 100.0000%	Melasma: 56.5076%
Image 6	Normal: 90.5722%	Normal: 53.7602%
Image 7	Normal: 99.7090%	Normal: 99.7691%
Image 8	Normal: 99.9991%	Normal: 86.7657%
Image 9	Normal: 99.9992%	Angioedema: 63.4595%
Image 10	Normal: 99.9992%	Normal: 99.5918%

From the results obtained above, one can observe the superiority of FSDNet in predicting all classes with high accuracy of 98%. This can be explained by the fact that FSDNet is more trained on our data. The number of trainable parameters in FSDNet is greater than in pre-trained VGG-16. In pre-trained VGG-16 only the new softmax classifier is trained, while in FSDNet the training starts from block 5 that includes 3 CONV layers, as well as the classifier so the network learn more complex features specific to facial skin diseases.

4.4.4 Performance evaluation of FSDNet

We assess the performance of FSDNet in different brightness conditions and face poses to see how they can affect the accuracy of the prediction.

1- Brightness effect

We have studied the effect of brightness on three classes: eczema, acne, and normal. We have chosen a test image from each class and applied the brightness variations on it. We have changed the brightness with different factors, 0.8, 0.6, and 0.5. The image gets darker. We have found that the images are still predicted correctly but with a slightly lower accuracy. The results are presented in Figure 4.15 , Figure 4.16 and Figure 4.17.

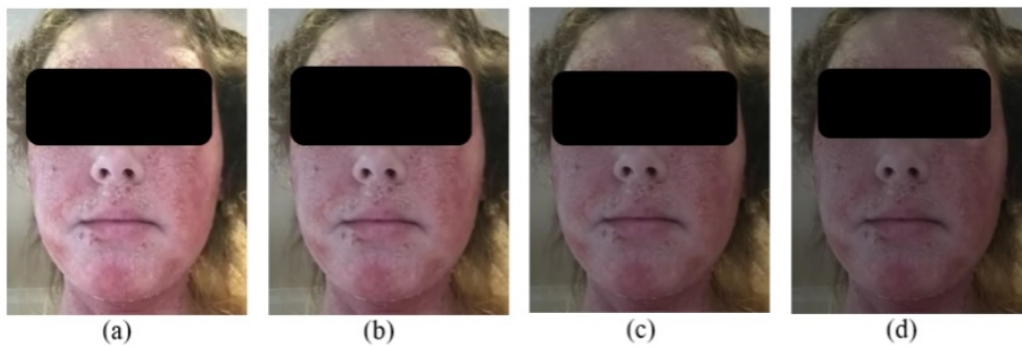


Figure 4.15: Study of the effect of brightness on the performance in class Eczema. (a) The original image is identified as Eczema with an accuracy of 99.9%. (b) The brightness is modified by a factor of 0.8, and Eczema is predicted with an accuracy of 99.1%. (c) The brightness is changed by a factor of 0.6, and the accuracy of identification of Eczema is 98.5%. (d) Finally, with a modification by a factor of 0.5, the image is classified as Eczema with an accuracy of 95.1%.

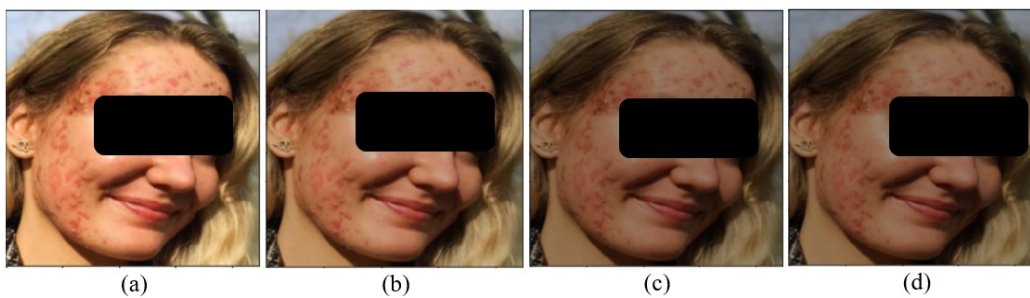


Figure 4.16: Study of the effect of brightness on the performance in class Acne. (a) The original image is identified as acne with an accuracy of 99.99%. (b) The brightness is modified by a factor of 0.8, and the identification accuracy of acne is 99.99%. (c) Acne is identified with an accuracy of 99.96% when the brightness is varied by a factor of 0.6. (d) The image becomes darker with a brightness modification by a factor of 0.5, and the image is predicted as acne with an accuracy of 99.93%.

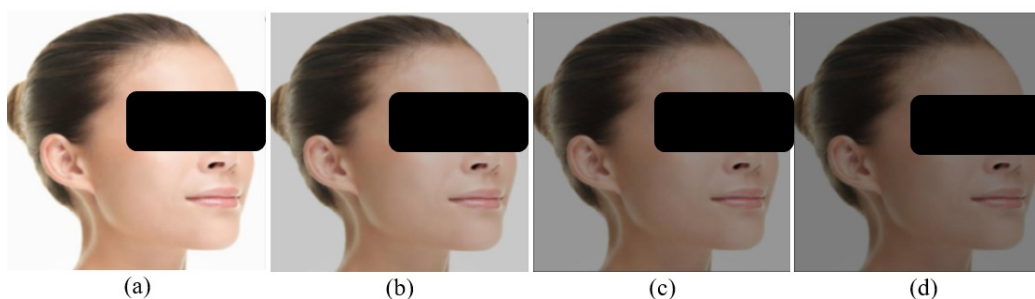


Figure 4.17: Study of the effect of brightness on the performance in class Normal. (a) The original image and images with brightness modified by a factor of (b) 0.8, (c) 0.6, and (d) 0.5. All images are correctly predicted as belonging to normal class with an accuracy of 99.99%, 99.99%, 99.87%, and 99.77%, respectively.

2- Face pose effect

We have evaluated the effect of the face pose on the identification accuracy in three classes: vitiligo, angioedema, and melasma. We have taken a test image, and we have applied two rotations of 30° and -30° on the image to change the face pose. All images are correctly predicted with accuracies varying according to the class as presented in Figure 4.18, Figure 4.19, Figure 4.20.

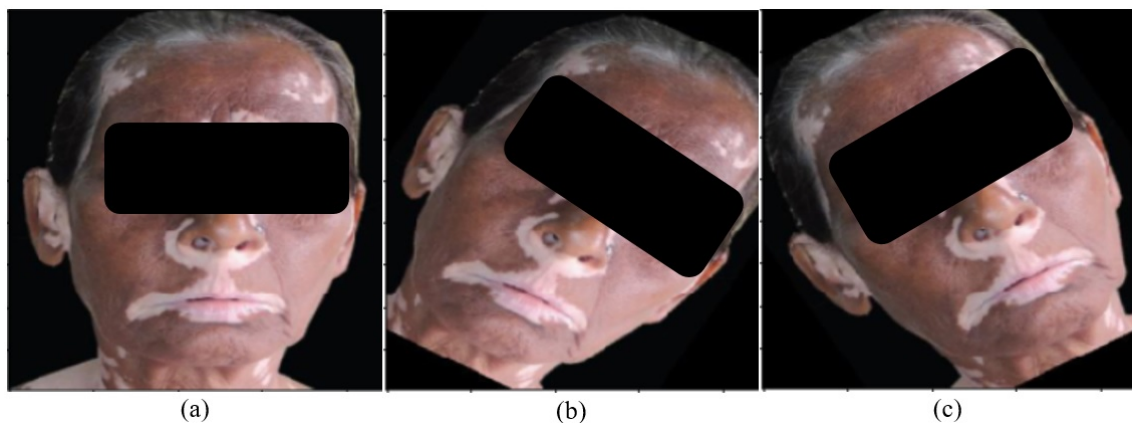


Figure 4.18: (a) The original image is identified as vitiligo with an accuracy of 100%. (b) When rotated 30° the image is identified also as having vitiligo with an accuracy of 99.99%. (c) The image is rotated with an angle -30° , and it is predicted as having vitiligo with an accuracy of 99.96%.

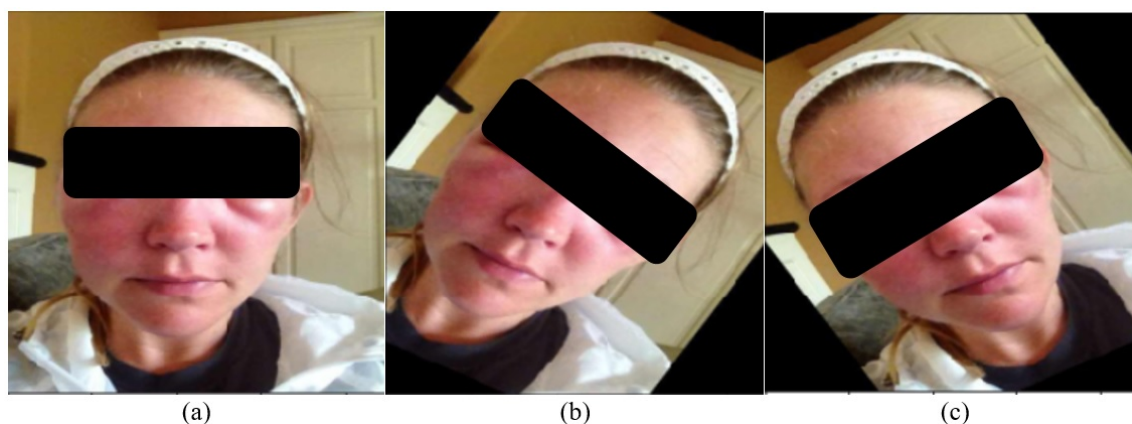


Figure 4.19: Angioedema: The 3 images are classified as angioedema with accuracies (a) 100% for the original image, (b) 98.81% when rotated 30° , and (c) 99.96% when rotated -30° .



Figure 4.20: (a) the original image is classified as melasma with an accuracy of 100%. (b) the image is rotated with an angle of 30° , and the accuracy of melasma prediction is 84.91%. (c) With a rotation of -30° , the image is classified as melasma with an accuracy of 99.89%.

In conclusion, we observe that the FSDNet performs better than the pre-trained VGG-16. FSDNet is a robust model that could identify the different handled classes with a high accuracy regardless of the face pose, the illumination, and image resolution since the model is trained on images of different resolutions, brightness and poses. Experimental results indicate that FSDNet can identify facial lesions, normal skin, and no face classes with high accuracy between 93.73% and 100%.

4.5 Facial skin disease identification using EfficientNet

b0

The second approach used to perform facial skin disease identification is transfer learning with efficientNet b0. The architecture of EfficientNet b0 has been modified. We have added the same layers we have used to build the FSDNet. We have added to the top of the model a global average pooling layer, a dropout of value 0.5, and a Softmax classifier of dimension 10. We have used the same architecture to make the comparison between the models meaningful. The architecture of the fine-tuned version of efficientNet b0 is presented in Figure 4.21.

Layer (type)	Output Shape	Param #
efficientnet-b0 (Model)	(None, 7, 7, 1280)	4049564
global_average_pooling2d_1 ((None, 1280)	0
dropout_1 (Dropout)	(None, 1280)	0
dense_1 (Dense)	(None, 10)	12810
Total params: 4,062,374		
Trainable params: 4,020,358		
Non-trainable params: 42,016		

Figure 4.21: Fine-tuned EfficientNet b0: The general architecture of EfficientNet b0 has been kept the same and 3 layers have been added to the top: a global average pooling, a dropout denoted as dropout_1, and a softmax of dimension 10 denoted dense_1. The total trainable parameters in the network is 4020358.

The fine-tuned EfficientNet bo takes in input an RGB image of dimension 224 x 224 x 3. Data are initially loaded into the model, and the classes are mapped (from 0 to 9) and converted to one-hot encoding. The model has been trained and validated using our dataset FSDD. We have tested the performance of the model in five data split cases 90:10, 80:20, 70:30, 60:40, and 50:50 for training versus validation using Adam and SGD optimizers to determine which one performs better. We have set the batch size to 16 and trained first the network for 50 epochs. We have found that around ten epochs the network fits well with the model (Figure 4.22), so we decided to train our model for 12 epochs to gain time. Also, the number of trainable parameters has been changed to study their effect on the performance. We have tried four cases: The network has been trained from block4, then block5, block6, and finally block7 to the end of the network. To be noted that efficientNet is composed of 7 MB blocks. Experimental results have shown that the best performance is achieved in split case 80:20 for training versus validation using Adam optimizer with a learning rate of 0.0001. High accuracies and very low loss are obtained. In Table 4.7, we present the training and validation accuracy and loss obtained in each training case for the split case 80:20.

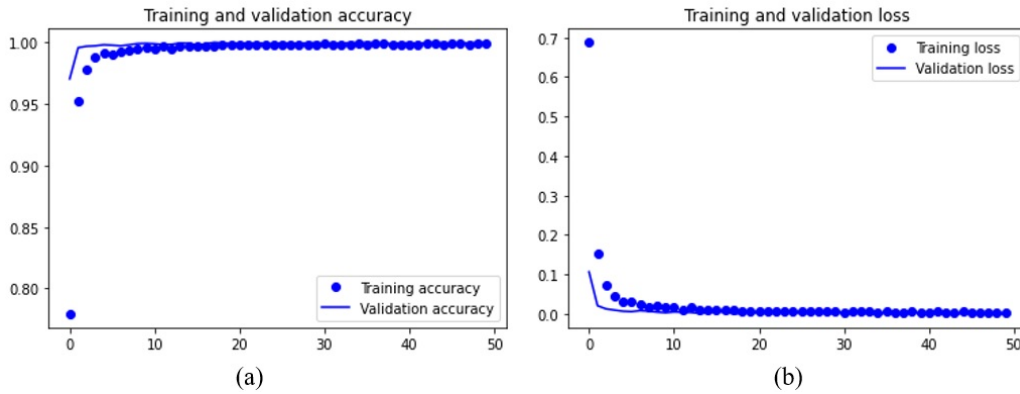


Figure 4.22: Curves of training and validation (a) accuracy, and (b) loss: We have observed the accuracy and loss of training and validation converge around the tenth epoch and their value is almost stable.

Trainable blocks	Optimizer	Training accuracy	Validation accuracy	Training loss	Validation loss
4, 5, 6, and 7	SGD	97.51%	99.53%	0.0886	0.0215
4, 5, 6, and 7	Adam	99.91%	99.97%	0.0034	0.0010
5, 6, and 7	Adam	99.85%	99.87%	0.0034	0.0053
5, 6, and 7	SGD	97.26%	99.38%	0.0998	0.0251
6, and 7	SGD	96.17%	99.28%	0.12668	0.0348
6, and 7	Adam	99.62%	99.9250%	0.0144	0.0032
7	Adam	99.84%	99.75%	0.0054	0.0100
7	SGD	89.17%	94.23%	0.3321	0.2030

Table 4.7: Training and validation accuracy loss of fine-tuned EfficientNet in split case 80:20: The performance of the model is studied according to the type of optimizer and the number of trainable blocks. The model achieves better performance when trained using Adam optimizer; the training and validation accuracy vary between 99.62% and 99.91%, and between 99.75% and 99.97%, respectively. The training and validation loss also vary between 0.0034 and 0.0144, and 0.001 and 0.01, respectively.

Since the training and validation accuracies and loss vary slightly with the number of trainable parameters, and according to the results obtained when testing the model with the test images that we have used in FSDNet, we have chosen to train our model from block 7 in addition to the adding layers at the top, which reduces the number of trainable parameters and thus the training time. To summarize the hyper-parameters of the fine-tuned EfficientNet b0, the model is used in split case 80:20 for training versus validation. It is trained for 12 epochs with batch size 16 using Adam optimizer with learning rate 0.0001. The curves of training and validation accuracy and loss are demonstrated in Figure 4.23. Experimental analysis and

performance evaluations have been carried out in terms of precision, recall, and F1-score which are widely preferred for the evaluation of machine learning techniques (Figure 4.24), as well as the confusion matrix (Figure 4.25).

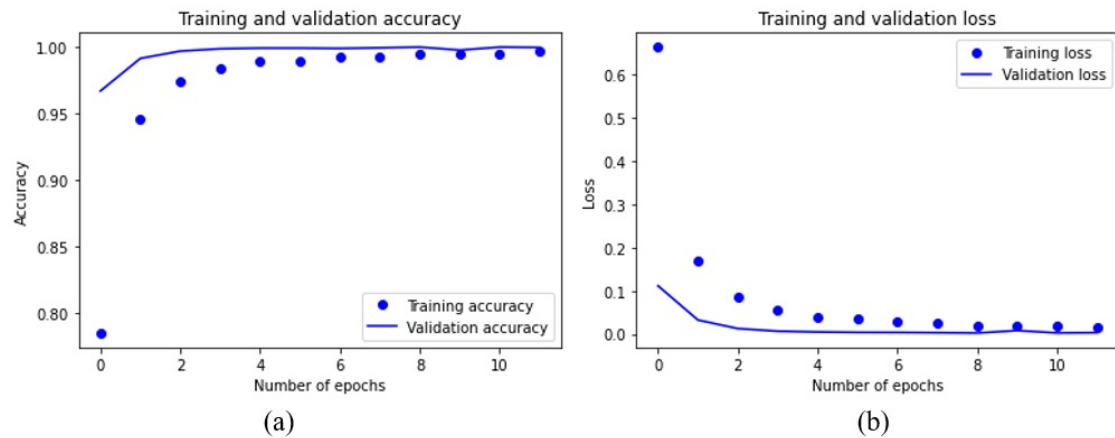


Figure 4.23: (a) Training and validation accuracy, and (b) training and validation loss curves of fine-tuned EfficientNet b0. The training and validation accuracy are both increasing, and the training and validation loss are decreasing together. There is no gap between them.

	Precision	Recal	F1-score	Support
Acne	1.00	1.00	1.00	407
Actinic	1.00	1.00	1.00	386
Angiodema	1.00	1.00	1.00	390
Blepharitis	1.00	1.00	1.00	414
Eczema	1.00	0.99	1.00	395
Melasma	1.00	1.00	1.00	415
Normal	1.00	1.00	1.00	402
No-Face	1.00	1.00	1.00	418
Rosacea	0.99	1.00	1.00	380
Vitiligo	1.00	1.00	1.00	393
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

Figure 4.24: Classification report of fine-tuned EfficientNet b0: The metrics precision, recall, and F1-score have been calculated for each class to evaluate the model. The column support represents the number of test samples available for validation. Macro avg stands for macro average that calculates the average without taking into account the proportion of each class from the total number of samples, and weighted average calculates the average with consideration of the proportion. The classifier has very high metrics varying between 0.99 and 1.

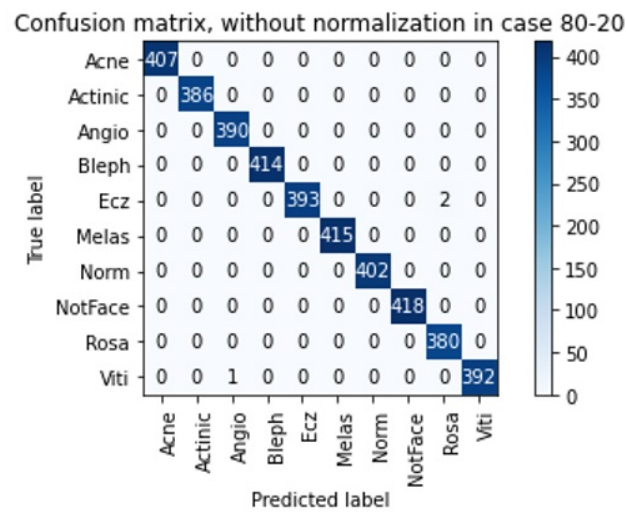


Figure 4.25: Confusion matrices of fine-tuned EfficientNet b0: The vertical axis of the confusion matrix represents the true label belonging to the ten classes (Acne, Actinic keratosis, Angioedema, Blepharitis, Eczema, Melasma, Rosacea, Vitiligo, Normal skin, and no face), and the horizontal axis is the predicted label. The diagonal of the matrix in dark color shows the number of correctly predicted samples and all the other numbers present the wrong predicted samples in each class. The model has predicted all the images correctly except for three images. One vitiligo image has been predicted as angioedema and two rosacea images have been classified as eczema.

EfficientNet b0 has predicted 3997 correct labels and 3 incorrect labels. To make sure that our model generalizes well, we have tested it with unseen images. For this purpose, we have provided the model with the same test images that have been used in FSDNet and pre-trained VGG-16. The results and the accuracy of prediction of the images in each class are presented in the tables below.

Class: Acne

Acne images	Predicted class with fine-tuned EfficientNet b0
Image 1	Acne: 65.0025%
Image 2	Acne: 99.7379%
Image 3	Acne: 96.3461%
Image 4	Acne: 71.1650%
Image 5	Melasma:59.0859%
Image 6	Normal: 72.7821%
Image 7	Acne: 86.9358%
Image 8	Acne: 77.6664%
Image 9	Acne:97.3565%
Image 10	Acne: 99.9390%

Class: Actinic keratosis

Actinic images	Predicted class with fine-tuned EfficientNet b0
Image 1	Actinic: 96.9876%
Image 2	Rosacea: 70.5007%
Image 3	Rosacea: 98.6781%
Image 4	Actinic: 99.9941%
Image 5	Actinic:74.1771%
Image 6	Actinic: 95.6463%
Image 7	Rosacea: 99.4783%
Image 8	Acne: 44.4715%
Image 9	Actinic:99.7310%
Image 10	Actinic: 97.9344%

Class: Angioedema

Angioedema images	Predicted class with fine-tuned EfficientNet b0
Image 1	Angioedema: 96.6407%
Image 2	Angioedema: 94.2552%
Image 3	Angioedema: 88.1840%
Image 4	Angioedema: 98.0572%
Image 5	Angioedema:97.2835%
Image 6	Angioedema: 75.4522%
Image 7	Angioedema: 99.8112%
Image 8	Angioedema: 98.8096%
Image 9	Angioedema:76.8689%
Image 10	Angioedema: 75.5810%

Class: Blepharitis

Blepharitis images	Predicted class with fine-tuned EfficientNet b0
Image 1	No-Face: 86.9879%
Image 2	Blepharitis: 90.9901%
Image 3	No-Face: 58.8601%
Image 4	Blepharitis: 87.9741%
Image 5	Blepharitis:93.0375%
Image 6	Eczema: 69.8308%
Image 7	Blepharitis: 99.9926%
Image 8	Blepharitis: 99.9973%
Image 9	Blepharitis:99.9997%
Image 10	Blepharitis: 99.9992%

Class: Eczema

Eczema images	Predicted class with fine-tuned EfficientNet b0
Image 1	Eczema: 96.6222%
Image 2	Eczema: 61.5240%
Image 3	Eczema: 79.5920%
Image 4	Rosacea: 54.6626%
Image 5	Acne:42.9557%
Image 6	Eczema: 99.5448%
Image 7	Eczema: 65.1269%
Image 8	Acne: 41.7624%
Image 9	Rosacea:98.0146%
Image 10	Eczema: 73.1151%

Class: Melasma

Melasma images	Predicted class with fine-tuned EfficientNet b0
Image 1	Melasma: 99.2175%
Image 2	Melasma: 89.1373%
Image 3	Melasma: 96.6173%
Image 4	Melasma: 89.3710%
Image 5	Melasma:97.6351%
Image 6	Rosacea: 89.4115%
Image 7	Melasma: 86.6251%
Image 8	Melasma: 80.5855%
Image 9	Acne:27.6550%
Image 10	Melasma: 99.0375%

Class: Rosacea

Rosacea images	Predicted class with fine-tuned EfficientNet b0
Image 1	Normal: 70.7044%
Image 2	Rosacea: 83.6251%
Image 3	Rosacea: 80.0116%
Image 4	Rosacea: 99.9880%
Image 5	Rosacea:95.5712%
Image 6	Rosacea: 77.1564%
Image 7	Rosacea: 87.1939%
Image 8	No-Face: 93.7725%
Image 9	Rosacea:97.5833%
Image 10	Normal: 68.2801%

Class: Vitiligo

Vitiligo images	Predicted class with fine-tuned EfficientNet b0
Image 1	Vitiligo: 60.2665%
Image 2	Vitiligo: 99.9791%
Image 3	Vitiligo:56.1974%
Image 4	Vitiligo: 64.4331%
Image 5	Vitiligo:88.8534%
Image 6	Angioedema: 99.9977%
Image 7	Angioedema: 58.0148%
Image 8	Vitiligo: 96.1195%
Image 9	Vitiligo:85.8112%
Image 10	Vitiligo: 69.8017%

Class: Normal

Normal images	Predicted class with fine-tuned EfficientNet b0
Image 1	Normal: 87.1288%
Image 2	Normal: 99.9964%
Image 3	No-Face:99.9989%
Image 4	Normal: 88.1720%
Image 5	Normal:85.7319%
Image 6	Normal: 99.5307%
Image 7	Normal: 99.8434%
Image 8	Normal: 98.9554%
Image 9	Normal:61.9296%
Image 10	Normal: 81.9936%

Class: No-Face

No-Face images	Predicted class with fine-tuned EfficientNet b0
Image 1	No-Face : 99.9991%
Image 2	No-Face : 99.9870%
Image 3	No-Face:99.9920%
Image 4	No-Face : 99.9961%
Image 5	No-Face :99.9999%
Image 6	No-Face : 100.0000%
Image 7	No-Face : 99.9781%
Image 8	No-Face :99.9981%
Image 9	No-Face :99.9991%
Image 10	No-Face : 99.9957%

Fine-tuned EfficientNet b0 is considered a good model to identify facial skin diseases. It achieves in average a classification accuracy of 89.5%. We have tested the efficiency of the proposed model in different conditions such as brightness and face pose as done in FSDNet. We have used the same test images used in FSDNet

evaluation. The brightness is modified by factors 0.8, 0.6, and 0.5 to make the image darker. We have applied two rotations of 30° and -30° to change the pose of the face. The model has predicted correctly the classes of all the images but with a small decrease in the accuracy of identification. The results are illustrated in (Figure 4.26) and (Figure 4.27).









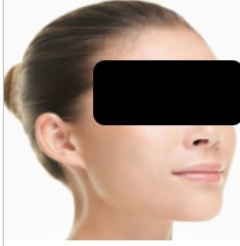
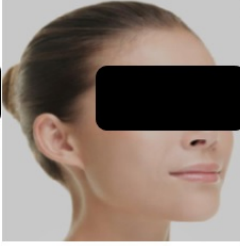
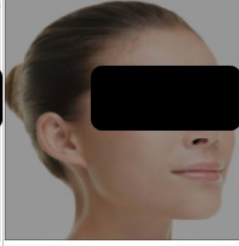
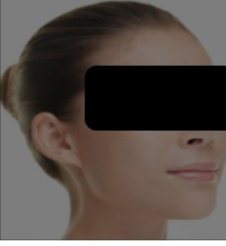
	Original image	Brightness modified by a factor 0.8	Brightness modified by a factor 0.6	Brightness modified by a factor 0.5
Class: Eczema				
Accuracy of fine-tuned EfficientNet b0	96.6222%	94.4356%	92.9532%	91.1572%
	Original image	Brightness modified by a factor 0.8	Brightness modified by a factor 0.6	Brightness modified by a factor 0.5
Class: Acne				
Accuracy of fine-tuned EfficientNet b0	71.1650%	70.8576%	70.2389%	69.8354%
	Original image	Brightness modified by a factor 0.8	Brightness modified by a factor 0.6	Brightness modified by a factor 0.5
Class: Normal skin				
Accuracy of fine-tuned EfficientNet b0	91.6582%	89.9231%	88.5251%	88.1561%

Figure 4.26: A study of the effect of brightness of the image on the performance of fine-tuned EfficientNet b0 model in 3 classes: eczema, acne, and normal skin: All the images are correctly identified but with a lower accuracy as the image gets darker.










	Original image	Rotation of 30°	Rotation of -30°
Class: Vitiligo			
Accuracy of fine-tuned EfficientNet b0	96.1195%	95.8508%	95.7748%
	Original image	Rotation of 30°	Rotation of -30°
Class: Angioedema			
Accuracy of fine-tuned EfficientNet b0	94.2552%	92.9912%	89.5512%
	Original image	Rotation of 30°	Rotation of -30°
Class: Melasma			
Accuracy of fine-tuned EfficientNet b0	97.6351%	96.6300%	89.0506%

Figure 4.27: Face pose effect on the performance of fine-tuned EfficientNet b0 evaluated on 3 classes: vitiligo, angioedema, and melasma. The classes of the images are predicted correctly, but the accuracy decreases with the rotation.

4.6 Facial skin disease identification using transfer learning with Inception V3

The third model for facial skin disease identification is built using transfer learning with Inception V3. We have added at the top of the original model a global average pooling layer, dropout of value 0.5, and a softmax classifier of dimension 10. We have used the same architecture used at the top of FSDNet to make the comparison between the models significant. The architecture of the fine-tuned version Inception V3 is presented in Figure 4.28.

Layer (type)	Output Shape	Param #
inception_v3 (Model)	(None, 5, 5, 2048)	21802784
global_average_pooling2d_1 ((None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 10)	20490
Total params: 21,823,274		
Trainable params: 21,788,842		
Non-trainable params: 34,432		

Figure 4.28: Fine-tuned Inception V3: We have conserved the general architecture of Inception V3, and we have added 3 layers at the top: global average pooling, dropout denoted as dropout_1, and 10-dimension softmax classifier denoted dense_1. The total trainable parameters in the network is 21788842.

The network takes in input an RGB image of dimension 229 x 229 x 3. Data is initially loaded into the model, and the classes are mapped (from 0 to 9) and converted to one-hot encoding. The model is trained and validated using FSDD dataset, in different conditions, to determine the best configuration leading to the best performance.

The network is trained in five data split cases 90:10, 80:20, 70:30, 60:40, and 50:50 for training versus validation using a batch size 16 for 30 epochs, and Adam optimizer with a learning rate 0.0001.

The accuracy and loss of the model almost stabilize around the epoch 12, and the curves of training and validation converge as seen in Figure 4.29. We have also changed the number of trainable parameters in the model.

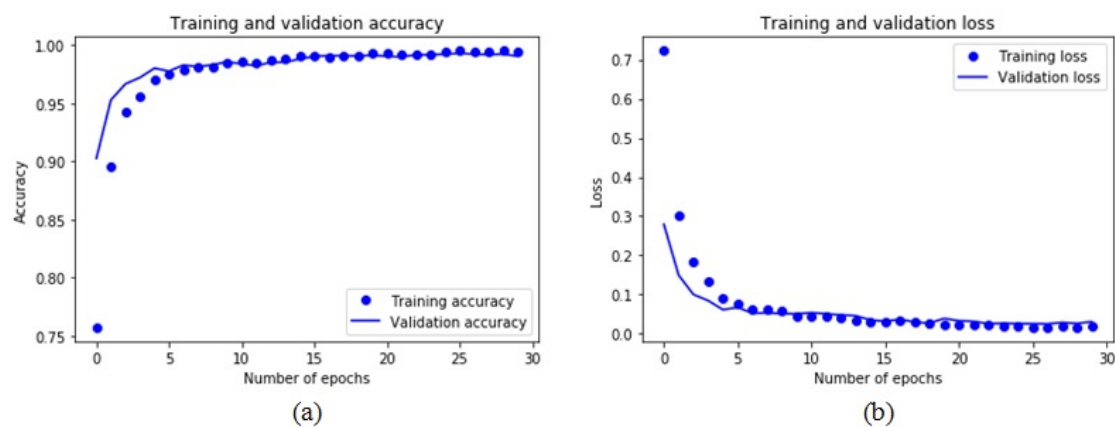


Figure 4.29: Fine-tuned Inception V3: We have conserved the general architecture of Inception V3, and we have added 3 layers at the top: global average pooling, dropout denoted as dropout_1, and 10-dimension softmax classifier denoted dense_1. The total trainable parameters in the network is 21788842.

Experimental results show that fine-tuned inception v3 in split case 80:20 for training versus validation trained for 15 epochs with a batch 16 using Adam optimizer with a learning rate of 0.0001 is the most accurate. Hence, we have adopted this model to make predictions. 6,094,026 parameters have been trained in the model including the new added layers of the classifier. It has provided an accuracy of 99%. The training and validation accuracy and loss are presented in Figure 4.30. Experimental analysis and performance evaluations have been carried out in terms of precision, recall, and F1-score (Figure 4.31), and we have also computed the confusion matrix (Figure 4.32).

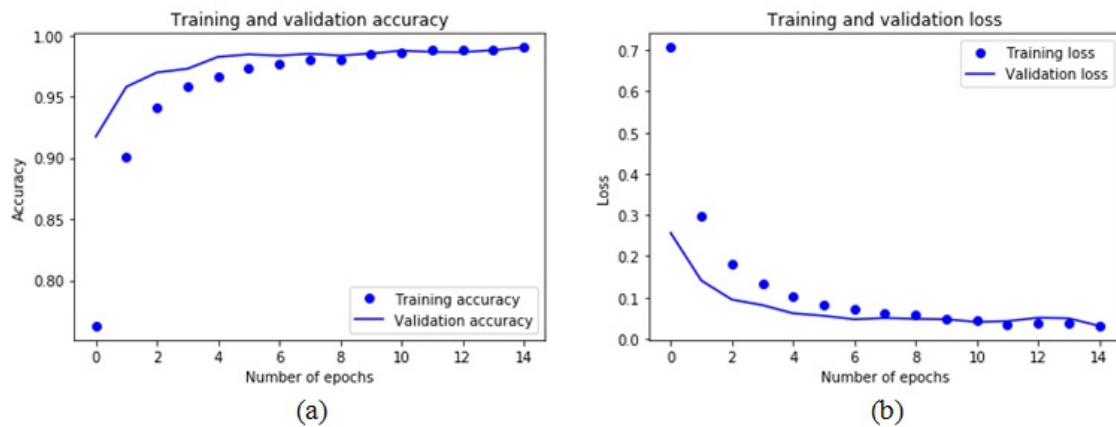


Figure 4.30: (a) Training and validation accuracy, and (b) training and validation loss curves of fine-tuned Inception V3: The training and validation accuracy are both increasing, and also the training and validation loss are decreasing together.

	Precision	Recal	F1-score	Support
Acne	0.99	1.00	1.00	407
Actinic	1.00	0.97	0.99	386
Angiodema	0.99	0.99	0.99	390
Blepharitis	1.00	1.00	1.00	414
Eczema	0.97	0.99	0.98	395
Melasma	0.99	0.98	0.98	415
Normal	1.00	0.99	0.99	402
No-Face	1.00	1.00	1.00	418
Rosacea	0.97	0.98	0.98	380
Vitiligo	1.00	0.99	0.99	393
macro avg	0.99	0.99	0.99	4000
weighted avg	0.99	0.99	0.99	4000

Figure 4.31: Classification report of fine-tuned Inception V3: The metrics precision, recall, and F1-score are computed for each class to assess the model. The column support represents the number of test samples available for validation while, macro avg stands for macro average that calculates the average without taking into account the proportion of each class from the total number of samples, and weighted average calculates the average with consideration of the proportion. The classifier presents high metrics varying between 0.97 and 1.

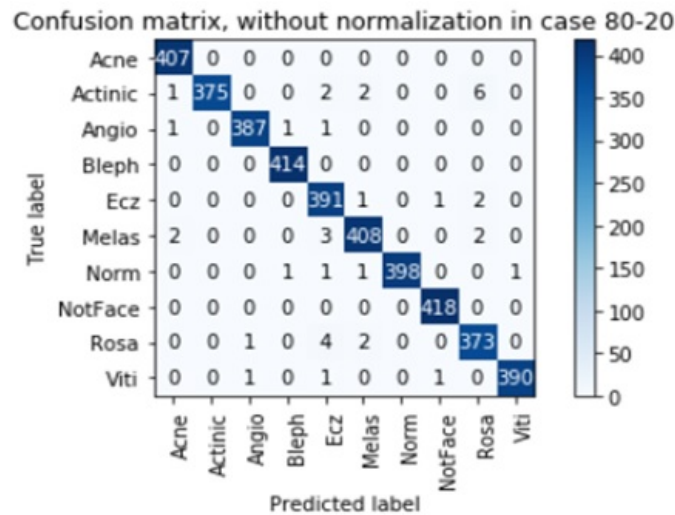


Figure 4.32: Confusion matrix of fine-tuned Inception V3: The vertical axis of the confusion matrix represents the true label belonging to the ten classes (Acne, Actinic keratosis, Angioedema, Blepharitis, Eczema, Melasma, Rosacea, Vitiligo, Normal skin, and no face), and the horizontal axis is the predicted label. The diagonal of the matrix in dark color shows the number of correctly predicted samples while all the other numbers present the wrong predicted samples in each class. Generally, the model has predicted correctly a great number of images. The number of incorrect labels in each class is very small.

The model could predict 3961 correct labels and 39 incorrect labels. After these evaluations, the next step consists of testing the proposed model with unseen images to make sure that it could identify correctly the class of each one. The same test images used in the previous models mentioned above are used. The results of each class are demonstrated in the tables below.

Class: Acne

Acne images	Predicted class with inception V3
Image 1	Eczema: 71.5601%
Image 2	Acne: 89.8751%
Image 3	Acne: 89.6239%
Image 4	Acne: 83.7845%
Image 5	Acne: 76.7306%
Image 6	Melasma: 72.7248%
Image 7	Normal: 56.0885%
Image 8	Acne: 78.1119%
Image 9	Acne: 77.8795%
Image 10	Acne: 70.0996%

Class: Actinic keratosis

Actinic images	Predicted class with inception V3
Image 1	Actinic: 94.8712%
Image 2	Melasma: 77.1518%
Image 3	Melasma: 61.3543%
Image 4	Actinic:99.9806%
Image 5	Actinic:71.5958%
Image 6	Actinic: 90.5395%
Image 7	Melasma: 83.0333%
Image 8	Acne: 33.6184%
Image 9	Actinic:99.9722%
Image 10	Actinic: 99.8913%

Class: Angioedema

Angioedema images	Predicted class with inception V3
Image 1	Angioedema: 88.2401%
Image 2	Angioedema: 89.6239%
Image 3	Angioedema: 88.1840%
Image 4	Angioedema: 76.7306%
Image 5	Angioedema:75.9901%
Image 6	Angioedema: 85.5522%
Image 7	Angioedema: 79.5112%
Image 8	Angioedema: 93.0375%
Image 9	Acne:76.8689%
Image 10	Angioedema: 70.5702%

Class: Blepharitis

Blepharitis images	Predicted class with inception V3
Image 1	Blepharitis: 80.4607%
Image 2	Blepharitis: 75.9901%
Image 3	No-Face: 68.7601%
Image 4	Blepharitis: 83.0333%
Image 5	Blepharitis:71.5958%
Image 6	Melasma: 61.3543%
Image 7	Blepharitis: 99.9926%
Image 8	Blepharitis: 99.9973%
Image 9	Blepharitis:99.9997%
Image 10	Acne: 73.6184%

Class: Eczema

Eczema images	Predicted class with inception V3
Image 1	Eczema: 85.5543%
Image 2	Eczema: 59.9673%
Image 3	Eczema: 72.3570%
Image 4	Melasma: 52.8105%
Image 5	Eczema:63.4372%
Image 6	Eczema: 87.5295%
Image 7	Rosacea: 53.6154%
Image 8	Normal: 79.5539%
Image 9	Acne:43.9354%
Image 10	Eczema: 72.9042%

Class: Melasma

Melasma images	Predicted class with inception V3
Image 1	Melasma: 95.4175%
Image 2	Melasma: 71.2955%
Image 3	Melasma: 60.1918%
Image 4	Melasma: 71.9109%
Image 5	Melasma:79.2687%
Image 6	Acne: 48.8724%
Image 7	Melasma: 78.8591%
Image 8	Melasma: 71.5855%
Image 9	Acne:45.3441%
Image 10	Melasma: 98.9854%

Class: Rosacea

Rosacea images	Predicted class with inception V3
Image 1	Acne: 36.5183%
Image 2	Rosacea: 78.7014%
Image 3	Rosacea: 79.9520%
Image 4	Rosacea: 85.1588%
Image 5	Rosacea:72.7248%
Image 6	Rosacea: 69.8019%
Image 7	Rosacea: 74.7943%
Image 8	Eczema: 95.2470%
Image 9	Rosacea:81.9585%
Image 10	Eczema: 88.4642%

Class: Vitiligo

Vitiligo images	Predicted class with inception V3
Image 1	Vitiligo: 70.3655%
Image 2	Vitiligo: 79.1751%
Image 3	Vitiligo:66.2674%
Image 4	Vitiligo: 75.3321%
Image 5	Vitiligo:83.5634%
Image 6	Normal: 79.6677%
Image 7	Angioedema: 55.2148%
Image 8	Vitiligo: 85.2195%
Image 9	Vitiligo:83.6112%
Image 10	Vitiligo: 75.6018%

Class: Normal

Normal images	Predicted class with inception V3
Image 1	Normal: 71.2217%
Image 2	Normal: 99.6220%
Image 3	No-Face:99.9936%
Image 4	Normal: 69.6572%
Image 5	Normal:99.7377%
Image 6	Normal: 88.0761%
Image 7	Normal: 97.3916%
Image 8	Normal: 85.8012%
Image 9	Normal:93.1284%
Image 10	Normal: 77.3763%

Class: No-Face

No-Face images	Predicted class with inception V3
Image 1	No-Face : 99.9595%
Image 2	No-Face : 97.8035%
Image 3	No-Face:99.8604%
Image 4	No-Face : 99.5975%
Image 5	No-Face :99.5367%
Image 6	No-Face : 99.9990%
Image 7	No-Face : 99.1813%
Image 8	No-Face :96.3868%
Image 9	No-Face :99.6456%
Image 10	No-Face : 99.8493%

After being trained and validated on our dataset, fine-tuned Inception v3 model could generalize on unseen images. Generally, it could predict the classes of the

different test images with an average accuracy of 82.5%. We have also evaluated the performance of the proposed model according to the brightness of the image, and the face pose using the same test images used in the previous mentioned models for the same tasks. We have applied the same modifications of brightness and face pose as previously. The results mentioned in (Figure 4.33) and (Figure 4.34) show that the model could predict correctly the class of each image but with a lower accuracy. Hence, we can conclude that the model can perform efficiently regardless of the brightness and the face pose in the image.









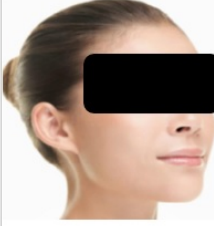
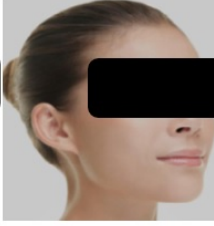
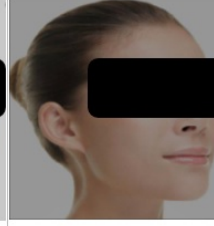
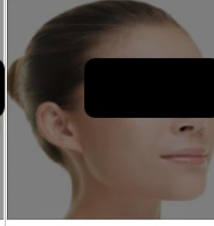
	Original image	Brightness modified by a factor 0.8	Brightness modified by a factor 0.6	Brightness modified by a factor 0.5
Class: Eczema				
Accuracy of fine-tuned Inception V3	85.5543%	81.9757%	80.4779%	80.1801%
	Original image	Brightness modified by a factor 0.8	Brightness modified by a factor 0.6	Brightness modified by a factor 0.5
Class: Acne				
Accuracy of fine-tuned Inception V3	83.7845%	76.9984%	75.5583%	74.9905%
	Original image	Brightness modified by a factor 0.8	Brightness modified by a factor 0.6	Brightness modified by a factor 0.5
Class: Normal skin				
Accuracy of fine-tuned Inception V3	76.3189%	75.4830%	74.1576%	72.4413%

Figure 4.33: Evaluation of the effect of brightness of the image on the performance of fine-tuned Inception V3 model in 3 classes: eczema, acne, and normal skin. All the images are correctly identified but with a lower accuracy as the image gets darker.










	Original image	Rotation of 30°	Rotation of -30°
Class: Vitiligo			
Accuracy of fine-tuned Inception V3	85.2195%	83.9108%	82.7610%
	Original image	Rotation of 30°	Rotation of -30°
Class: Angioedema			
Accuracy of fine-tuned Inception V3	89.6239%	88.5270%	85.2876%
	Original image	Rotation of 30°	Rotation of -30°
Class: Melasma			
Accuracy of fine-tuned Inception V3	79.2687%	77.1847%	75.6196%

Figure 4.34: Face pose effect on the performance of fine-tuned Inception V3: It is evaluated on 3 classes: vitiligo, angioedema, and melasma. The images are classified correctly in the correct classes. The accuracy decreases with the rotation.

4.7 Comparative study

In Table 4.8, we compare the test prediction results of each model used in our approach, in terms of the number of correct labels and accuracy of prediction.

	Acne	Actinic keratosis	Angioedema	Blepharitis	Eczema	Melasma	Rosacea	Vitiligo	Normal skin	No face
Total number of images	10	10	10	10	10	10	10	10	10	10
Number of correct predictions in FSDNET	9	8	10	9	8	10	9	10	10	10
Average accuracy of correct predictions with FSDNet	99.86%	94.6%	99.9%	100%	97.4%	97.3%	97.75%	93.73%	98.26%	100%
Number of correct predictions in pre-trained VGG-16	7	6	6	8	5	6	9	6	8	9
Average accuracy of correct predictions with pre-trained VGG-16	73.6%	85.42%	79.9%	95.6%	80.2%	90.2%	97.35%	72.31%	88.35%	99%
Number of correct predictions with fine-tuned EfficientNet b0	8	6	10	7	6	8	7	8	10	10
Average accuracy of correct predictions with fine-tuned EfficientNet b0	86.7%	94%	80%	96%	79%	92%	88.7%	88%	90%	99.9%
Number of correct predictions with fine-tuned Inception V3	7	6	9	7	6	8	7	8	9	10
Average accuracy of correct predictions with fine-tuned Inception V3	80.8%	92.7%	83%	87.2%	75%	80%	77.5%	70%	80%	98.9%

Table 4.8: Summary of results: The number of correct and incorrect predicted labels for each class are shown in the table below. The FSDNet could predict more correct labels and with higher accuracy than the pre-trained VGG-16 model.

	Total number of correct labels	Average accuracy of prediction
Pre-trained VGG-16	70	86.19%
FSDNet	92	98%
Fine-tuned EfficientNet b0	80	89.5%
Fine-tuned Inception V3	77	82.5%

Table 4.9: Summary of test images classification: We mention in the table the average accuracy of classification of pre-trained VGG-16, FSDNet, fine-tuned EfficientNet b0, and fine-tuned Inception V3, as well as the total number of correctly predicted images among 100 test images.

Table 4.8 and Table 4.9 show that the models that we have built with transfer learning adapted to facial skin disease classification; FSDNet, fine-tuned EfficientNet b0, and fine-tuned Inception V3, are more accurate than the pre-trained VGG-16 model. Transfer learning has boosted the performance of the network. Since the model has been trained previously and learned over large sets, transfer learning provides it with a higher starting accuracy, a faster convergence, and a higher asymptotic accuracy when used on a new task. The number of trainable parameters differs between the models. We have 40970 trainable parameters in pre-trained

VGG-16, 7084554 in FSDNet, 3168550 in fine-tuned EfficientNet b0, and 6094026 in fine-tuned Inception V3. Among all, FSDNet is the most accurate model. It has achieved the highest average accuracy of classification of 98%, and has been able to predict the highest number of test images, which is 92 correct labels among 100. It is followed by fine-tuned EfficientNet b0 with 80 correct labels predicted with an average accuracy of 89.5%. Then comes fine-tuned Inception V3 with 77 correct labels classified with an average accuracy of 82.5%, and finally pre-trained VGG-16 with 70 correct labels and an average accuracy of 86.19%. FSDNet is based on VGG-16 which is characterized by the stack of a small sized kernel 3 x 3. Using this stack is better than using a large sized kernel because numerous nonlinear layers augment the depth of the model, and thus it can learn more complex features at a low cost. Moreover, small size kernels help to maintain the finer level properties of the image. This explains the superiority and the advantage of FSDNet in facial skin diseases identification among the other models.

4.8 Conclusion

In this chapter, we have presented a complete end-to-end computer vision and a deep learning method to perform facial skin disease identification using RGB images. To do so, first we built our facial skin disease dataset (FSDD) including 20000 labelled images referring to the eight facial skin diseases to be identified, normal skin, and no-face categories. The ten classes are balanced. This dataset is used to train and validate our models regardless of face pose, illumination, and image resolutions. We have tested four models: a pre-trained model, and three fine-tuned models adapted to facial skin diseases classification. We have adopted the same architecture of the classifier in three of the proposed models to make the comparison meaningful. The first model was the pre-trained VGG-16. After training and validation, we have evaluated the model on our testing set and found that the network has achieved an 86.19% identification accuracy. The second model is a

model that we have built using transfer learning with VGG-16. We have removed the classifier layers of the main architecture of VGG-16 and replaced them by a global average pooling, a dropout of 0.5, and a softmax classifier of dimension 10 to fit the number of handled classes in our approach. The model, called Facial Skin Diseases Network (FSDNet), has also been trained and validated using FSDD. Our proposed model has achieved a classification accuracy of 98% on the testing set. The model has proven to be very robust and performs well in different conditions of brightness and face poses. The third model is based on EfficientNet b0. We have fine-tuned the main architecture of the model by adding the same layers used in FSDNet. Once trained and validated on FSDD, the suggested network has been tested and has provided a classification accuracy of 89.5%. The last model is fine-tuned Inception V3. The same architecture adopted in the previous networks has been added at to the top of the original architecture of Inception V3. The model has been is trained, and validated on our dataset. It has identified the classes of the images of the test set with an accuracy of 82.5%. The three suggested models could identify the classes of the different images regardless of the face pose, and the brightness of the images. As seen, FSDNet has achieved the best results among all and this is due to the topology of the network based on the use of a small sized kernel that could detect and learn very fine features in the image.

General conclusion and perspectives

The use of artificial intelligence in the medical field, especially in dermatology, has been expanding due to advances in machine learning and deep learning algorithms which have revolutionized image recognition and classification which form the basis for the application of AI in dermatology. AI based methods require big amounts of data, which are considered the oil of these approaches. The progress in digital cameras has led to an important augmentation in the number of digital data available which helps to build dermatology image datasets. Deep learning application in dermatology has exceeded the use to identify skin cancer. Many automated techniques have been developed to detect facial skin lesions such as acne, eczema, and psoriasis. All these developments can assist dermatologists and improve the sensitivity and accuracy of screening skin disorders. They also allow remote analysis of skin, early diagnosis and treatment of skin lesions, and minimize waiting times for appointments.

Within this context, we have proposed an AI model to classify facial skin diseases. Compared to the state of the art, three main contributions are highlighted in this work: (1) the identification process doesn't require the extraction of the ROI from face images since the system is trained regardless face pose, illumination, image resolution, etc. (2) the number of detected pathologies is greater compared to the one reported in the literature.(3) Due to the absence of any standard public dataset for the same, we created a database composed of RGB images of the different classes

identified in our work.

We have introduced a deep learning based computer aided diagnosis system that could identify more diverse diseases than the ones handled in the literature. Our system identifies 8 facial skin diseases such as acne, actinic keratosis, angioedema, blepharitis, eczema, melasma, rosacea, and vitiligo. It can also determine if the skin is “normal” and can indicate “no face” if the image is not a facial image. For this purpose, we have built three convolutional neural network (CNN) models to predict the class of the images using a transfer learning method. CNN is a commonly used shift invariant method of extracting learnable features. CNNs have played a major role in the development and popularity of deep learning and neural networks, and are mainly used in image classification. As a first step, we have used the pre-trained VGG-16 model without changing its architecture to classify facial skin diseases. We have only replaced the softmax classifier of dimension 1000 by a new one of dimension 10 to fit the number of classes handled in our method. The aim was to see this pre-trained model on another dataset for a specific task and observe how it will perform on a new task that is facial skin diseases identification and compare it with the models that we will build using transfer learning. The model was trained and validated using a dataset that we have created and we called Facial Skin Diseases Dataset (FSDD). Actually, there is not any public dataset handling these diseases, so we had to build our own dataset composed of 20000 images referring to the ten classes that are 8 facial skin diseases, normal skin, and no face. The classes are balanced; each class has 2000 labelled images for different persons, of different genders, group ages, and races. They are also of different resolutions, illumination, and face poses. We have trained only the new softmax classifier that we have added. We trained the model for 30 epochs with batch size of 16, compiled with categorical crossentropy loss function, and Adam optimizer with a learning rate of 0.0001 with different split cases. The model performs the best when it is trained with 90% of the data and validated on 10%. Pre-trained VGG-16 achieves an accuracy of 97% and

can predict the classes of unseen images with an average accuracy of 86.19%. The first model is a VGG-16 based model adapted to facial skin diseases identification. We have kept the convolutional base of VGG-16, and replaced the top composed of a flatten layer, 2 fully connected layers, and a softmax classifier of dimension 1000 with a global average pooling layer, a dropout of 0.5, and a 10-dimension softmax classifier corresponding to the number of predicted classes. The model is trained and validated using our dataset FSDD. We have tried many split cases of data into training and validation to see which case achieves a better performance. After many trials to set the best hyper-parameters, and the best optimizer to be used, FSDNet has been trained from block 5, for 10 epochs with batch size 16, using Adam optimizer with a learning rate 0.0001. Experimental results of performance evaluation metrics have shown that FSDNet in split case 80:20 for training versus validation has the best performance, achieving an accuracy of 98.57%. It could predict the classes of test images that are not included in FSDD with a high accuracy of 98%. We have built the second model using transfer learning with EfficientNet b0. We have added to the pre-trained efficientnet b0 model the same layers added to FSDNet at the top of the network to make the comparison and the evaluation of the classifier meaningful. The model is trained and validated on FSDD data from block 6, for 12 epochs with batch size 16 using Adam with learning rate 0.0001. Also the performance has been evaluated in different split cases, and the best performance is obtained with split case 80:20 for training versus validation, with an accuracy of 99.92%. The fine-tuned EfficientNet model adapted to facial skin diseases identification has been tested with the same test images used in FSDNet. The built model predicts the classes with an accuracy of 89.5%. The third model we proposed is created using transfer learning with Inception V3. The same architecture adopted in the previous models is used at the top of Inception V3. We have trained the new added layers and some previous layers as well, for 15 epochs with a batch 16 using Adam optimizer with a learning rate of 0.0001. The best metrics are obtained in

the split case 80:20 for training versus validation, with an accuracy of 99%. Despite this high accuracy, when tested on new unseen images, the model has achieved an average classification accuracy of 82.5%.

As perspectives, we can implement FSDNet, seeing that it is the most accurate model adapted to facial skin diseases identification, into a real time acquisition system that acquires face images and predicts directly the corresponding class. On the other hand, the model can be improved to identify more diseases and to measure the severity of the disease. Other pre-trained architectures can also be explored using transfer learning. Also, we can work on increasing the size of FSDD by adding more classes and more images. We can use Generative adversarial networks, GAN, to generate new synthetic images referring to the different classes identified in our approach and study their effect on the performance of the models.

Bibliography

- [1] YOUSSEF H., ALHAJJ M., SHARMA S. Anatomy, Skin (Integument), Epidermis. 2020 Jul 27. StatPearls [Internet]. Jan 2021. Treasure Island (FL): StatPearls Publishing; PMID: 29262154.
- [2] ARDA O., GÖKSÜGÜR N., TÜZÜN Y. Basic histological structure and functions of facial skin. Clin Dermatol. [Internet]. Jan-Feb 2014, vol. 32, n°1, p. 3-13. doi: 10.1016/j.clindermatol.2013.05.021. PMID: 24314373.
- [3] CHANG C.Y., LIAO. H.Y. Automatic Facial Spots and Acnes Detection System. Journal of Cosmetics, Dermatological Sciences and Applications [Internet]. January 2013, vol. 3, p. 28-35 <http://dx.doi.org/10.4236/jcdsa.2013.31A006>.
- [4] CHANTHARAPHAICHIT T., UYYANONVARA B., SINTHANAYOTHIN C., NISHIHARA A. (2015). Automatic acne detection with featured Bayesian classifier for medical treatment. Semantic Scholar [Internet]. 2015.
- [5] AMINI M., VASEFI F., VALDEBRAN M., HUANG K, ZHANG H., KEMP W., MACKINNON N. Automated facial acne assessment from smartphone images. Proc. SPIE 10497, Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues XVI, 104970N. February 2018. San Francisco; doi: 10.1117/12.2292506.
- [6] BAJAJ L., KUMAR H., HASIJA Y. Automated System for Prediction of Skin

- Disease using Image Processing and Machine Learning. *International Journal of Computer Applications* [Internet]. February 2018, vol. 180, n° 19, p. 9-12.
- [7] SHEN X., ZHANG J., YAN C., ZHOU H. An Automatic Diagnosis Method of Facial Acne Vulgaris Based on Convolutional Neural Network. *Scientific Reports* [Internet]. 2018, vol. 8, 5839, DOI: 10.1038/s41598-018-24204-6.
- [8] HAMEED N., SHABUT A.M., GHOSH M.K., HOSSAIN M.A. Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques. *Expert Systems With Applications* [Internet]. 2020, vol. 141, 112961. <https://doi.org/10.1016/j.eswa.2019.112961>.
- [9] AL-MASNI M.A., KIMA D.H., KIM T.S. Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification. *Computer Methods and Programs in Biomedicine* [Internet]. 2020, vol. 190, 105351, doi: 10.1016/j.cmpb.2020.105351. PMID: 32028084.
- [10] GOCERI E. Deep learning based classification of facial dermatological disorders. *Computers in Biology and Medicine*[Internet]. 2021, vol. 128, 104118, <https://doi.org/10.1016/j.compbimed.2020.104118>.
- [11] WU Z., ZHAO S., PENG Y., et al. Studies on Different CNN Algorithms for Face Skin Disease Classification based on Clinical Images. *IEEE access SPECIAL SECTION ON DATA-ENABLED INTELLIGENCE FOR DIGITAL HEALTH*. June 2019, vol. 7, p. 66505-66511, DOI: 10.1109/ACCESS.2019.2918221.
- [12] WU H., YIN H., CHEN H., et al. A deep learning, image based approach for automated diagnosis for inflammatory skin diseases. *Annals of Translational Medicine* [Internet]. March 2020 , vol. 8, n° 9, <http://dx.doi.org/10.21037/atm.2020.04.39>.
- [13] SHANTHI T., SABEENIAN R.S., ANAND R. Automatic diagnosis of skin

- diseases using convolution neural network. *Microprocessors and Microsystems* [Internet]. 2020, vol. 76, 103074. DOI:10.1016/j.micpro.2020.103074.
- [14] IQBAL I., YOUNUS M., WALAYAT K, et al. Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society* [Internet]. March 2021, vol. 88, 101843. DOI: 10.1016/j.compmedimag.2020.101843. PMID: 33445062.
- [15] BHATE K., Williams H.C. Epidemiology of acne vulgaris. *The British journal of dermatology* [Internet]. March 2013, vol. 168, n° 3, p. 474-485. DOI:10.1111/bjd.12149.
- [16] Acne vulgaris. *Nature Reviews Disease Primers* [Internet]. 2015, vol. 1, 15033, <https://doi.org/10.1038/nrdp.2015.33>
- [17] RIGEL D.S., COCKERELL CJ, et al. Actinic keratosis, basal cell carcinoma, and squamous cell carcinoma. In BOLOGNIA J.L., et al., *Dermatology*. Second edition. Mosby Elsevier, Spain, 2008, p. 1645-1658.
- [18] BERNSTEIN J.A., CREMONESI P., HOFFMANN T.K., HOLLINGSWORTH J., (2017). Angioedema in the emergency department a practical guide to differential diagnosis and management. *International Journal of Emergency Medicine* [Internet]. 2017 December, vol. 10, n° 1. DOI: 10.1186/s12245-017-0141-z.
- [19] WONG K. Blepharitis. *DermNet NZ* .2009, <https://dermnetnz.org/topics/blepharitis2009>
- [20] OAKLEY A. Dermatitis. *DermNet NZ*, 1997, <https://dermnetnz.org/topics/allergic-contact-dermatitis>
- [21] STANWAY A., JARRETT P. Atopic dermatitis. *DermNet NZ*, February 2021. <https://dermnetnz.org/topics/atopic-dermatitis>.

- [22] OAKLEY A., GOMEZ J. Seborrheic dermatitis. DermNet NZ. October 2017.
<https://dermnetnz.org/topics/seborrhoeic-dermatitis>.
- [23] OAKLEY A. Contact dermatitis. DermNet NZ. 2012.
<https://dermnetnz.org/topics/contact-dermatitis>.
- [24] OAKLEY A., DOOLAN B.J., GUPTA M. Melasma. DermNet NZ, October 2020. <https://dermnetnz.org/topics/melasma>.
- [25] OAKLEY A. Rosacea. DermNet NZ. 2014.
<https://dermnetnz.org/topics/rosacea>.
- [26] OAKLEY A. Vitiligo. DermNet NZ. 2015.
<https://dermnetnz.org/topics/vitiligo>.
- [27] GOODFELLOW I., BENGIO Y., COURVILLE A. Deep Learning [Internet]. MIT Press, 2016. [http : // www.deeplearningbook.org](http://www.deeplearningbook.org).
- [28] RUSSELL S.J, NORVIG P. (2010) Artificial Intelligence: A Modern Approach. Third Edition, Prentice Hall, ISBN 9780136042594.
- [29] ROMAN V. Unsupervised Machine Learning: Clustering Analysis. Medium [Internet]. October 2019. <https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e>.
- [30] DENG L., YU D. Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing [Internet]. 2016, vol. 7, n° 3 – 4, p. 1–199. DOI:10.1561/20000000039.
- [31] OSBORN G. Mnemonic for hyperbolic formula. The Mathematical Gazette. July 1902, vol. 2, n° 34, [ref 20-06-2015], p. 189. DOI:10.2307/3602492.
- [32] HAHNLOSER R.H., SARPESHKAR R., MAHOWALD M.A., et al. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature [Internet]. June 2000, vol. 405, n° 6789, p. 947-951. DOI: 10.1038/35016072.

- [33] NAIR V., HINTON G.E. (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning. 2018. <http://citeseerx.ist.psu.edu/viewdoc/download>. DOI:10.1.1.165.6419rep=rep1type=pd.
- [34] SCHMIDHUBER J. Deep Learning in Neural Networks: An Overview. Neural Networks [Internet]. Jan 2015, vol. 61, p. 85-117, arXiv:1404.7828.
- [35] BALDI P. Autoencoders, Unsupervised Learning, and Deep Architectures. In: Proceedings of ICML Workshop. 2012, vol. 27, p. 37-50, Washington, USA.
- [36] HINTON G. Deep belief networks. Scholarpedia [Internet]. 2009, vol. 4, n° 5, p. 5947. DOI:10.4249/scholarpedia.5947.
- [37] AMIDI A., AMIDI S. VIP Cheatsheet: Recurrent Neural Networks. Cs 230 – Deep learning [Internet]. November 2018. <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>.
- [38] KARPATHY A. Convolutional Networks. CS231n [Internet]. <http://cs231n.github.io/convolutional-networks>.
- [39] IOFFE S., SZEGEDY C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv [Internet]. 2015. <http://arxiv.org/abs/1502.03167v3>.
- [40] HINTON G.E., SRIVASTAVA N., KRIZHEVSKY A., et al. Improving neural networks by preventing co-adaptation of feature detectors. arXiv [Internet]. 2018. <https://arxiv.org/pdf/1207.0580.pdf>.
- [41] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P. Gradient-based learning applied to document recognition. In Proceedings of the IEEE. , Nov. 1998, vol. 86, no. 11, pp. 2278-2324. DOI: 10.1109/5.726791.

- [42] KRIZHEVSKY A., SUTSKEVER I., HINTON G. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* [Internet]. Jan 2012, vol. 25, p. 1106–1114. DOI: 10.1145/3065386.
- [43] SZEGEDY C., LIU W., JIA Y. et al. Going Deeper with Convolutions. *arXiv* [Internet]. 2014. .arXiv:1409.4842v1.
- [44] SIMONYAN K., ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *Proceedings of ICLR*. 2015, San Diego, CA, USA. arXiv:1409.1556v6.
- [45] HE K., ZHANG X., REN S., SUN J. Deep Residual Learning for Image Recognition. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [46] YAMASHITA R., NISHIO M., DO R.K.G., TOGASHI K. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* [Internet]. 2018, vol.9, p.611–629. <https://doi.org/10.1007/s13244-018-0639-9>.
- [47] RUDER S. An overview of gradient descent optimization algorithms. *ArXiv* [Internet]. 2016, arXiv:1609.04747v2.
- [48] KINGMA D.P., Ba J. Adam: a method for stochastic optimization. In: *Proceedings of the 3rd International Conference for Learning Representations (San Diego 2015)*. 2017, arXiv:1412.6980v9.
- [49] HUANG G., LIU Z., WEINBERGER K.Q. Densely Connected Convolutional Networks. In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. 2017, p. 2261-2269. arXiv:1608.06993v5.
- [50] TAN M., LE Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: *Proceedings of International Conference on Machine Learning*, 2019, arXiv:1905.11946v5.

- [51] SZEGEDY C., VANHOUCKE V., IOFFE S., et al. (2016). Rethinking the inception architecture for computer vision. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition, 2016, p. 2818-2826. DOI: 10.1109/CVPR.2016.308.
- [52] SZEGEDY C., IOFFE S., VANHOUCKE V., ALEMI A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI. 2017. arXiv:1602.07261v2.
- [53] LAVALLE S. M., BRANICKY M. S., LINDEMANN S. R. On the relationship between classical grid search and probabilistic roadmaps. The International Journal of Robotics Research. 2004, vol. 23, n°7-8, p. 673-692. DOI: 10.1177/0278364904045481.
- [54] SANDLER M., HOWARD G.A., ZHU M., et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018, p. 4510-4520. arXiv:1801.04381v4.
- [55] RAMACHANDRAN P., ZOPH B., LE Q.V. Searching for Activation Functions. ArXiv [Internet]. 2017. arXiv:1710.05941v2.
- [56] HU J., SHEN L., ALBANIE S., et al. Squeeze-and-Excitation Networks. ArXiv [Internet]. 2019. arXiv:1709.01507v4.
- [57] RUSSAKOVSKY O., DENG J., SU H., et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision [Internet]. 2015, vol. 115, p. 211-252. arXiv:1409.0575v3.
- [58] Dermweb photo atlas. www.dermweb.com/photoatlas.
- [59] LIN M., CHEN Q., SHUICHENG Y. Network In Network. CoRR [Interent]. 2014. 10 p. arXiv:1312.4400v3.