TRABALHO DE CONCLUSÃO DE CURSO

APPLICATION OF CNNS IN THE IDENTIFICATION OF PULMONARY AND DERMATOLOGICAL DISEASES: A PERFORMANCE ANALYSIS

Lucas Santos Lessa

Brasília, julho de 2024

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

TRABALHO DE CONCLUSÃO DE CURSO

APPLICATION OF CNNS IN THE IDENTIFICATION OF PULMONARY AND DERMATOLOGICAL DISEASES: A PERFORMANCE ANALYSIS

Lucas Santos Lessa

Trabalho de conclusão de curso submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Engenheiro Eletricista

Banca Examinadora

Prof. Daniel Guerreiro e Silva, Ph.D, FT/UnB
Orientador

Prof. Eduardo Peixoto Fernandes da Silva, Ph.D, FT/UnB
Examinador Interno

Prof. Osmar Luiz Ferreira, Msc., FT/UnB
Examinador externo

FICHA CATALOGRÁFICA

LESSA, LUCAS SANTOS
APPLICATION OF CNNS IN THE IDENTIFICATION OF PULMONARY AND DERMATOLOGICAL DISEASES: A PERFORMANCE ANALYSIS [Distrito Federal] 2024.
xvi, 54 p., 210 x 297 mm (ENE/FT/UnB, Engenheiro, Engenharia Elétrica, 2024).
Trabalho de conclusão de curso - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. CNN	2. COVID
3. Skin lesions	4. Medical images
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

LESSA, S.L. (2024). *APPLICATION OF CNNS IN THE IDENTIFICATION OF PULMONARY AND DERMATOLOGICAL DISEASES: A PERFORMANCE ANALYSIS*. Trabalho de conclusão de curso, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 54 p.

CESSÃO DE DIREITOS

AUTOR: Lucas Santos Lessa
TÍTULO: APPLICATION OF CNNS IN THE IDENTIFICATION OF PULMONARY AND DERMATOLOGICAL DISEASES: A PERFORMANCE ANALYSIS.
GRAU: Engenheiro Eletricista ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Projeto Final de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Projeto Final de Graduação pode ser reproduzida sem autorização por escrito do autor.

Lucas Santos Lessa
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

Dedicatória

Dedico este trabalho aos meu pais, especialmente à minha mãe, Elaine Carla, sem a qual esse percurso não teria sido possível.

Dedicatory

I dedicate this work to my parents, especially my mother, Elaine Carla, without whom this journey would not have been possible.

Lucas Santos Lessa

Agradecimentos

Primeiramente eu gostaria de agradecer à minha Mãe, Elaine Carla, por sempre ter me apoiado, me incentivado e ser a minha inspiração de coragem. Obrigado por todas as vezes em que a senhora rasgou as folhas do meu caderno da escola, obrigando-me a fazer melhor, seu apoio e exemplos foram primordiais. A coragem para empreender na jornada acadêmica surgiu em mim apartir do testemunho diário de sua força para enfrentar os desafios da vida, da sua coragem para fazer uma graduação, simultaneamente, com o desafio de ser mãe de duas crianças pequenas. Sem sombra de dúvida, minha mãe é a razão e a motivação para a conclusão desse curso. Esse trabalho é uma pequena demonstração do meu respeito, reconhecimento e reverência por essa grande mulher que é a minha mãe.

Ao meu Pai, Jacques Fernando, obrigado, pois, com o seu jeito de ser e com suas palavras, por vezes duras e desafiadoras, me ensinou o valor da constância e da persistência, me levando a densenvolver uma couraça necessária para enfrentar a vida com honra e resiliência. Foram suas criações e "gatos", próprios do seu espírito inventivo, que me apresentaram, precocemente, o universo da elétrica.

Ao meu Padastro e Padrinho, Joaquim Teles, por me mostrar o universo acadêmico e ter me apoiado nesse processo, sempre me incentivando a ser ousado, a arriscar e a viver a energia da juventude.

Aos meus 6 irmãos, Jorge, Juliana, Jean, Janaína, Junior e Livia pelos conselhos de vida, apoio e compreensão. Especialmente ao Jean, por me instigar na excelência dos estudos e por mostrar o caminho do empreendedorismo. E, ao meu irmão mais velho e também Padrinho, Jorge, por contribuir para que meu sonho do intercâmbio fosse possível.

Ao Caio Flôres pela parceria e apoio mútuo ao longo desse percurso de graduação. Na academia encontrei inúmeras pessoas incríveis, algumas se tornaram colegas, outras amigos, mas o Caio, além de tudo isso, é um irmão. Ao encerrar esse percurso sinto imensa gratidão à vida, pelo presente que ela me dá de poder concluir o curso com êxito e levar a amizade sincera de alguém que vivenciou cada disciplina comigo, cada dúvida, cada frustação mas, também, cada vitória. E ao meus outros amigos: César Pissolati, pelo companherismo e espontaneidade que, em muitos momentos, trouxe alegria onde a angústia da jornada era intensa; César Carlyle, pela disposição e generosidade de sacrificar seu descanso para socorrer os amigos; Anne Matos; Mateus Vieira; Pedro Ferreira pelas orientações profissionais e pela amizade. Vocês contribuiram para o sucesso da minha jornada.

Ao Prof. Amauri, também conhecido como Patrão, pelo padrão de excelência como docente, engenheiro e pessoa. Obrigado por ter me mostrado a amplitude de possibilidades que

a engenharia oferece, agora sei que eu posso ser algo mais do que um motorista de aplicativo, sem a sua participação o meu intercâmbio para os EUA não teria sido possível.

Ao IEEE-CIS pela oportunidade de aprendizado, tenho enorme respeito e admiração pelo capitulo. Agradeço também a todos os Professores, bem como amigos, colegas e conhecidos que me ajudaram a arcar com os custos inerentes ao meu intercâmbio, especialmente ao Wallace Flôres por tanto que fez nesse sentido.

Ao meu orientador, Prof. Daniel Guerreiro, obrigado pela excelente orientação, pela paciência e pelo rigor no que se refere a construção deste trabalho. A todos os membros da banca, obrigado pela disponibilidade e contribuição a este trabalho.

À Universidade de Brasília - UnB por tudo que ela me proporcionou: pelo crescimento acadêmico; pelo intercâmbio humano e pelo PIBIC que culminou nesse trabalho de conclusão de curso.

E enfim, mas, não menos importante, a Deus.

Lucas Santos Lessa

Acknowledgments

First, I would like to thank my mother, Elaine Carla, for always supporting me, encouraging me and being my inspiration for courage. Thank you for all the times you tore up my schoolwork, forcing me to do better. Your support and example were always essential. The courage to undertake the academic journey undoubtedly arose in me from your daily testimony of courage to face life's challenges, from the courage to pursue a degree at the same time as the challenge of being a mother of two small children. Without a shadow of a doubt, she is the reason and motivation for completing this course. This work is a small demonstration of my respect, recognition and reverence for this great woman who is my mother.

To my father, Jacques Fernando, thank you, because with your way of being and with your words, sometimes harsh and challenging, you taught me the value of constancy, persistence, leading me to develop the armor necessary to face life with honor and resilience. It was your creations and "cats" typical of your inventive spirit that introduced me to the world of electricity at an early age.

To my stepfather and godfather, Joaquim Teles, for showing me the academic world and supporting me in this process, always encouraging me to be bold, take risks and live the energy of youth.

To my 6 siblings, for their life advice, support and understanding. Especially to Jean, for instigating excellence in studies and showing me the path of entrepreneurship. And to my brother Jorge, for helping to make my dream of an exchange program possible.

To Caio Flôres, for your partnership and mutual support throughout this undergraduate course. At the academy, I met countless incredible people, some of whom became colleagues and others friends, but you, above all, are a brother. As I conclude this journey, I am immensely grateful to life for the gift it gave me of being able to successfully complete the course and to have the sincere friendship of someone who experienced each subject with me, each doubt, each frustration but also each victory. And to my other friends: César Pissolati, for his companionship and irreverence that in many moments brought joy when the anguish of the journey was intense; César Carlyle, for his willingness and generosity in sacrificing his rest to help his friends; Anne Matos; Mateus Vieira; Pedro Ferreira for their professional guidance and friendship. You contributed to the success of my journey.

To Prof. Amauri, also known as Patrão, for his standard of excellence as a teacher, as an engineer and as a person. For having shown me the breadth of possibilities that engineering offers, I now know that I can be something more than a ride-sharing driver. Without his participation, my exchange to the USA would not have been possible.

To IEEE-CIS for the learning opportunity; I have enormous respect and admiration for the chapter. I also thank all the professors, as well as friends, colleagues and acquaintances who helped me cover the costs inherent to my exchange. Especially to Wallace Flôres for all he did in this regard.

To my advisor, Prof. Daniel Guerreiro, for his excellent guidance, patience and rigor in the construction of this work. To all the members of the committee, thank you for your availability and contribution to this work.

To the University of Brasília - UnB for everything it has provided me: for my academic growth; for the human exchange and for PIBIC, which culminated in this final project.

And finally, to God.

Lucas Santos Lessa

RESUMO

O uso de técnicas de aprendizado de máquina para auxiliar no diagnóstico de doenças por meio de imagens médicas tem impulsionado um grande volume de pesquisas na área de visão computacional. Neste trabalho, propomos a utilização de redes neurais convolucionais (CNNs) para identificar doenças pulmonares e lesões de pele, com o objetivo de otimizar as etapas de diagnóstico e tratamento. Buscamos implementar a arquitetura mais eficaz para sistemas de saúde como o SUS, reduzindo o tempo necessário para o diagnóstico.

Realizou-se dois experimentos utilizando o conjunto de dados HAM10000, que inclui as classes Nevo Melanocítico (NV), Melanoma (mel), Lesões Benignas (bkl), Carcinoma Basocelular (bcc), Ceratoses Actínicas e Carcinoma Intraepitelial/Doença de Bowen (akiec), Lesões Vasculares (vasc) e Dermatofibroma (df), e o conjunto de dados COVID, que contém as classes Opacidade Pulmonar, COVID-19, Pneumonia Viral e Normal. Este estudo avalia seis redes convolucionais reconhecidas por seu alto desempenho em outras aplicações, buscando identificar a arquitetura mais adequada com base em diversas métricas de avaliação.

Para cada modelo de CNN treinado, apresentamos as métricas de acurácia, sensibilidade, especificidade, precisão e F1-score. Na classificação de lesões de pele com o conjunto de dados HAM10000, a arquitetura que obteve melhor desempenho foi a EfficientNetB4, com uma acurácia de 84% e F1-scores de 91.4%, 65.7%, 75.0%, 80.4%, 68.9%, 77.3% e 75.9%, respectivamente, para as classes Nevo Melanocítico (nv), Melanoma (mel), Lesões benignas (bkl), Carcinoma basocelular (bcc), Ceratoses actínicas e carcinoma intraepitelial/Doença de Bowen (akiec), Lesões vasculares (vasc) e Dermatofibroma (df). Na classificação de doenças pulmonares, a EfficientNetB7 se destacou com uma acurácia de 97% e F1-scores de 93.7%, 98.6% e 99.3% para as classes Opacidade Pulmonar, COVID-19 e Pneumonia Viral, respectivamente.

ABSTRACT

The use of machine learning techniques to aid in the diagnosis of diseases through medical images has driven a large volume of research in the area of computer vision. In this work, we propose the use of convolutional neural networks (CNNs) to identify lung diseases and skin lesions, with the aim of optimizing the stages of diagnosis and treatment. We seek to implement the most effective architecture for health systems such as SUS, reducing the time required for diagnosis.

Two experiments were carried out using the HAM10000 dataset, which includes the classes Melanocytic Nevus (NV), Melanoma (mel), Benign Lesions (bkl), Basal Cell Carcinoma (bcc), Actinic Keratoses and Intraepithelial Carcinoma/Bowen's Disease (akiec), Vascular Lesions (vasc) and Dermatofibroma (df), and the COVID dataset, which contains the classes Lung Opacity, COVID-19, Viral Pneumonia and Normal. This study evaluates six convolutional networks recognized for their high performance in other applications, seeking to identify the most suitable architecture based on several evaluation metrics.

For each trained CNN model, we present the metrics of accuracy, sensitivity, specificity, precision and F1-score. In the classification of skin lesions with the HAM10000 dataset, the architecture that obtained the best performance was EfficientNetB4, with an accuracy of 84% and F1-scores of 91.4%, 65.7%, 75.0%, 80.4%, 68.9%, 77.3% and 75.9%, respectively, for the classes Melanocytic Nevus (nv), Melanoma (mel), Benign Lesions (bkl), Basal Cell Carcinoma (bcc), Actinic Keratoses and Intraepithelial Carcinoma/Bowen's Disease (akiec), Vascular Lesions (vasc) and Dermatofibroma (df). In the classification of lung diseases, EfficientNetB7 stood out with an accuracy of 97% and F1-scores of 93.7%, 98.6% and 99.3% for the classes Lung Opacity, COVID-19 and Viral Pneumonia, respectively.

Contents

List of Figures

List of Tables

ACRONYM

CNN	Convolutional Neural Network
DL	Deep Learning
MLP	Multilayer Perceptron
ANN	Artificial Neural Networks
NN	Neural Network
AI	Artificial Inteligence

1 INTRODUCTION

When individuals develop illnesses, the diagnostic journey often involves appointments with various specialists and numerous medical examinations. Identifying the ailment and determining an effective treatment can be time-consuming. Physicians, as specialists, undergo extensive years of study to analyze diseases through medical examinations, like medical images. However, the inherent limitations of human capability can pose challenges, like the time to analysis, human errors, disease evolution,genetic variation, skilled labor, and et cetera.

In a country like Brazil, which presents a healthcare system with big challenges related to quality and universalization, an AI system for disease detection or patient triage could enhance efficiency. Later in this text, we will explore into the details, but it's worth noting that some diseases covered by this project, like Pneumonia, Skin lesions, and COVID-19 are responsible for thousands of deaths in Brazil: 2616 deaths were caused by Skin Cancer (SAUDE, 2020), Pneumonia 44.523 deaths from January to August 2022 (ALMEIDA et al., 2024), and the impact of COVID-19 has been profound. Based on this, there is a great motivation to develop an automatic tool to detect these illnesses (DUANMU et al., 2022).

The inception of Computer Aided Detection/Diagnosis (CAD) began in 1960 (FUJITA, 2020). CAD leverages machine learning to analyze imaging data, offering insights into the patient's condition (CHAN; HADJIISKI; SAMALA, 2020). With the development of Artificial Intelligence (AI) and, more specifically, Deep Learning (DL), it becomes feasible to extract specific features from images that conventional analyses might overlook or require excessive time and specialization (GOEL et al., 2021; SUN; HUANG; GUO, 2021; MARTINEZ-MURCIA et al., 2021).

The first ailment from the first dataset, Lung Opacity, refers to areas that normally have dark-appearing lungs that appear cloudy (HERRING, 2019). This opacity can be observed by areas that appear white on a chest radiograph when it should be darker as said by Türk e Kökver (2023). The identification of opacity in a chest X-ray image can signify various conditions, including the presence of fluid in air spaces, thickening of air space walls, lung tissue thickening, inflammation, pulmonary edema, vascular damage and bleeding, cancerous growth, and fibrosis (TÜRK; KÖKVER, 2023; ARENAS-JIMÉNEZ; PLASENCIA-MARTÍNEZ; GARCÍA-GARRIGÓS, 2021; KIM et al., 2022; ZHU et al., 2022).

Pneumonia, the second ailment, is an infection of one or both of the lungs. It can have several types: bacteria, viruses, or fungi (HOPKINS, 2024a). As one can read on (HOPKINS, 2024b), diagnosing pneumonia relies significantly on tests, which include a chest X-ray for internal imaging, blood tests to check for infection, arterial blood gas testing to measure blood oxygen, sputum culture for lung infection, pulse oximetry for painless blood oxygen measurement, a detailed chest CT scan, bronchoscopy for direct lung airway examination, and a pleural fluid culture to identify pneumonia-causing bacteria. These tests collectively contribute to a comprehensive diagnosis.

The third ailment, COVID-19, gained widespread recognition in 2020 due to its global viral outbreak, stemming from the SARS-CoV-2 virus. According to Pascarella et al. (2020), COVID-19 diagnosis primarily relies on the detection of the virus's genetic material using RT-PCR tests. These tests utilize nasal swabs, tracheal aspirates, or bronchoalveolar lavage specimens. A multifaceted approach, combining clinical, laboratory, and radiological methods, enhances the accuracy of COVID-19 diagnosis. Conventional chest radiography is not used to solely define the diagnosis, because it has lower diagnostic sensitivity (BERNHEIM et al., 2020).

Diagnosing diseases often involves a spectrum of methods, ranging from cost-effective to expensive or highly invasive examinations. One particularly accessible and widely used diagnostic tool is the Chest X-ray. The X-ray is a medical technique that is capable to visualize intern organs and has become ubiquitous globally due to its relative affordability, as noted by (HAYGOOD; BRIGGS, 1992).

AI algorithms can be employed to analyze these X-ray images, offering a valuable tool for physicians in the diagnostic process. While not replacing the expertise of healthcare professionals, AI-assisted analysis can serve as a guide, optimizing the medical procedure. By providing a preliminary assessment or highlighting regions of interest, these algorithms could help streamline the diagnostic workflow. Additionally, such AI-driven screening methods could offer patients a more informed and statistical perspective, aiding in shared decision-making regarding further diagnostic steps and potential treatment plans. This intersection of medical imaging and AI presents an exciting avenue for enhancing efficiency and precision in healthcare practices.

The second dataset utilized in this study encompasses a broad range of skin lesions, each representing a distinct condition. As highlighted by (SRINIVASU et al., 2021), there is a growing demand for research focused on developing procedures to identify the effects of various skin diseases through imaging technology. New methods like lasers and photonics-based medical technology have enabled a faster diagnosis of skin diseases, but the cost is still limited and expensive. This dataset consists of seven skin diseases: Melanocytic nevi, Benign keratosis-like lesions, Dermatofibroma, Vascular lesions, Actinic keratoses, Intraepithelial carcinoma, Basal cell carcinoma, and Melanoma. Actinic Keratoses and Intraepithelial Carcinoma/Bowen's Disease (AKIEC) are precancerous growths resulting from prolonged sun exposure, characterized by rough and scaly patches on the skin. Basal Cell Carcinoma (BCC), the most prevalent type of skin cancer, is distinguished by the presence of pearly or waxy bumps with visible blood vessels. Benign Keratosis-like Lesions (BKL) are non-cancerous skin growths that display a diverse appearance, ranging from elevated to flat lesions.

Furthermore, the dataset comprises Dermatofibroma (DF), a benign skin nodule characterized by firm, raised nodules that can have varying colors. Melanoma (MEL), the most severe form of skin cancer, is identified by irregular moles displaying asymmetrical shapes and uneven colors. Melanocytic Nevi (NV) are common moles, presenting as small, circular, typically brown spots on the skin. Vascular Lesions (VASC) encompass skin conditions related to blood vessels, often appearing as red or purple discolorations, such as hemangiomas. This diverse dataset provides

a comprehensive representation of various skin lesions, facilitating the development of robust diagnostic models.

Skin lesions are diagnosed through dermatological examinations that use visual inspection, clinical expertise, and sometimes dermoscopy. Dermatologists evaluate the lesion's appearance, including its color, shape, size, and border irregularities. In more complicated cases, dermatopathology may be used through a laboratory test.

The efficacy of a medical classification model may vary across countries, attributed to differences in training data sources and feature distributions. While the training dataset could originate from a different country, the crucial factor is ensuring that validation and test datasets incorporate real-world data relevant to the intended application. Merging datasets becomes a viable strategy to augment the volume of training data.

1.1 OBJECTIVES

The core objective of this study is to develop a pragmatic prototype system for efficiently detecting Viral Pneumonia, COVID-19, Lung Opacity, and various skin lesions. The envisioned outcome is creating a cost-effective diagnostic tool that leverages radiological images for respiratory conditions and skin images. The primary goal is to establish a prototype system that aids in the initial diagnosis, facilitating a more streamlined approach to patient triage within the Sistema Único de Saúde (SUS) in Brazil. This triage system holds the potential to optimize SUS operations, reduce diagnostic processing time, and serve as a valuable decision-making support tool. The anticipated advantages include mitigating hospital and health center congestion, alleviating administrative burdens, and contributing to the overall enhancement of the healthcare system.

Two datasets were selected for this objective. One of the datasets selected for this study is the Human Against Machine 10000 (HAM10000), which stands out as one of the largest and most diverse collections of publicly available dermoscopic images. Composed of 10.015 dermoscopic images, HAM10000 is widely used for training neural networks aimed at diagnosing pigmented skin lesions. This dataset covers a significant variety of diagnostic categories, including melanocytic nevus, melanoma, and benign and malignant lesions, among others, which makes it an essential tool for the development and validation of machine-learning models aimed at dermatology.

The second dataset chosen is the COVID Dataset. This dataset includes 3.616 confirmed cases of COVID-19, 10.192 normal lung X-ray images, 6.012 images showing lung opacity, and 1.345 cases of viral pneumonia, all from a variety of reliable sources. The volume of data contained in this dataset provides a solid foundation for training and evaluating neural network models aimed at detecting lung infections, allowing for an in-depth and robust study of the different manifestations of these conditions in imaging exams.

The secondary aim involves a comparative analysis of six distinct models applied to each se-

lected dataset. This analysis seeks to unveil performance variations among the models, providing insights into their strengths, weaknesses, and overall effectiveness within the specific contexts of the datasets under consideration.

2 BACKGROUND

A particular focus will be directed towards comprehending the theoretical frameworks underpinning Artificial Neural Networks (ANN), encompassing the intricate details of models, metrics, and the inherent specifications that collectively define this expansive field.

2.1 NEURAL NETWORK

As highlighted by Wu e Feng (2017), ANN have been a prominent focus in Artificial Intelligence (AI) since the 1980s. However, it's important to note that ANN constitutes merely a branch within Artificial Inteligence (AI) (GARDNER, 1998). The foundational theory underpinning ANN is a inspiration from the network processes observed in the brain (SUN; LIANG; CUI, 2021). Analogous to the interconnected neurons forming networks in the brain, ANN features layers containing numerous artificial neurons within each layer.

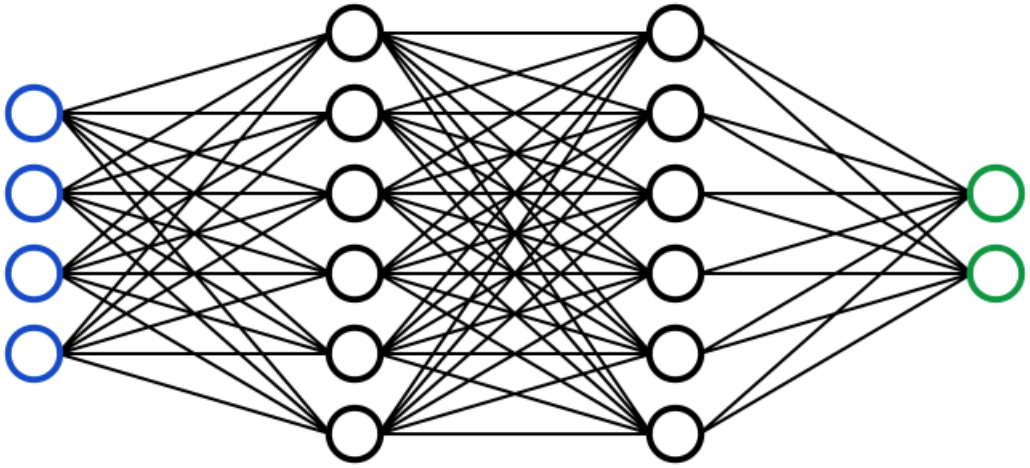


Figure 2.1: This figure is a representation of a Neural Network (ZHOU, 2020).The first layer of neurons, in blue, is the input layer through where the data flows. Between two layers all neurons are connected.

Fig. 2.1 presents an example of an ANN, the Multilayer Perceptron (MLP). Every neuron, symbolized by a small circle, establishes connections with all neurons in the subsequent layer. The initial layers, depicted as blue circles, are responsible for receiving raw data.

As affirmed by Block (1962), the Multilayer Perceptron (MLP) stands as a preeminent and extensively employed type of neural network, structured with an input layer, a hidden layer, and an output layer. The designation "hidden" underscores its intermediary function, abstaining from direct engagement with both inputs and outputs. Additionally, Kleyko et al. (2023) underscores

the significance of the perceptron, introduced by Rosenblatt (ROSENBLATT, 1957), serving as a fundamental artificial neuron or node within the MLP. Each node adeptly processes inputs, incorporating assigned weights and biases to generate an output, as detailed by (DELASHMIT; MISSILES, 2005):

$$y_i = \sum_{k=1}^{N} w_{j,k} \cdot x_k + B. \tag{2.1}$$

Here, N represents the total number of neurons connected to the given neuron, and j denotes the instance of each neuron. The terms $w_{j,k}$ represent the weights connecting the k^{th} input unit to the j^{th} hidden unit, x_k represents the input value of the preceding neuron, and B is the bias.

This function above represents a linear transformation, but with only this the layer it would learn strictly linear relationships. So, a non-linear transformation is necessary like as an activation function enabling the network to learn complex patterns (CHOLLET, 2022).

There are many types of activation functions, and each one has one output range. A list of common activation functions is shown in Table 2.1.

Name	**Formula**	**Range**
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$(0, 1)$
Hyperbolic Tangent (tanh)	$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$(-1, 1)$
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x)$	$[0, \infty)$
Leaky Rectified Linear Unit (Leaky ReLU)	$f(x) = \max(\alpha x, x)$	$(-\infty, \infty)$
Softmax	$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$	$(0, 1)$

Table 2.1: Activation Functions

In this project, we will focus on supervised learning it is a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns (GOODFELLOW; BENGIO; COURVILLE, 2016). In other words, supervised learning involves training the model with known answers.

2.1.1 Cost function

Training a Neural Network is to adjust the weights and bias to ensure the correct prediction or classification. Initially, the weights and biases are randomly selected and during the process of training, the parameters are iteratively modified each epoch.

An epoch is the process in which data passes through the neural network, from the input layer to the output layer, ending with the updating of the parameters. To make this possible it is necessary to have a cost function, that observes the true value and the predicted value and computes a score of error or loss. Figure 2.2 shows two layers Neural Network with the Loss function in a block diagram.

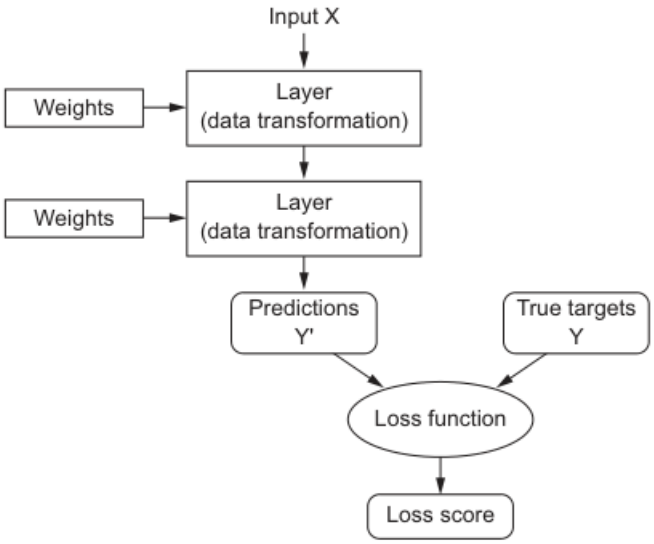


Figure 2.2: This is block diagram about the Loss function inside a two layers Neural Network (CHOLLET, 2022). The Cost function calculates a score, that has to be minimized, and that is the job of another function the Optimizer that will find the best parameters to minimize the loss function.

It is expected that the loss will be high in the initial epochs and should decrease after each subsequent epoch. The lesser the value of loss, the closer the output is to the true value.

2.1.2 Backpropagation

The loss function feeds into the *backpropagation* algorithm, a fundamental component of deep learning. This algorithm, is part of the the optimizer and enables the operation of utilizes gradient descent to adjust the weights and biases, minimizing the cost function (GOODFELLOW; BENGIO; COURVILLE, 2016). Backpropagation computes these gradients by applying the chain rule (GOODFELLOW; BENGIO; COURVILLE, 2016, Section 6.5.2).

Figure 2.3 shows the process of training a Neural Network using a block diagram. The input data passes through two layers and the final layer contains the model prediction that is computed using the loss function through the true samples with this a loss score is obtained which is the input of the optimizer to update the weights of the neural network.

As outlined by Leung e Haykin (1991), the backpropagation algorithm employs a systematic approach, aiming for the approximation of global minimization. It meticulously traverses the complex terrain of optimization, allowing the neural network to iteratively fine-tune its parameters and refine its ability to interpret and respond to the intricacies embedded in the input data.

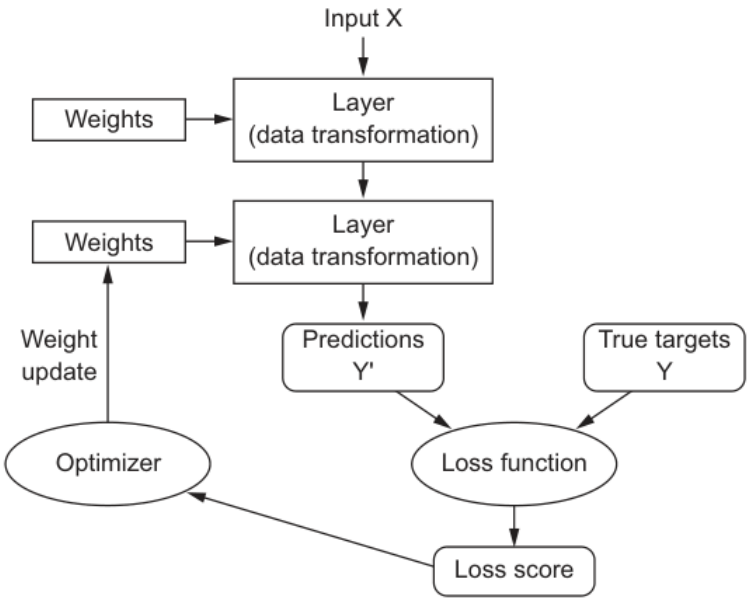


Figure 2.3: This is a block diagram about the training process of a Neural Network (CHOLLET, 2022). The Optimizer employs the implementation of the backpropagation algorithm to comput the gradient of the loss function.

2.1.3 Convolutional Neural Network

Convolutional Neural Network (CNN), a subclass of neural networks (NN), exhibits best performance in processing images, more specifically grid-shaped data. Over the years, CNN has yielded groundbreaking results, particularly in pattern recognition (ALBAWI; MOHAMMED; ALZAWI, 2017). This success is attributed to its adept management of pixel dependencies, this has allowed the emergence of many applications in the area of computer vision (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Initially, the utilization of CNN incurred substantial expenses, primarily due to the costs associated with training high-resolution images. However, with the advancements in GPU technology, it became more viable to work with CNNs.

The underlying concept of CNN lies in its hierarchical feature extraction process. A key innovation of CNN is its approach of selecting small regions from an image and passing them through the layers, eliminating the need for an excessive number of interconnected neurons for each, but only to pixels in their receptive fields (GERON, 2019; ALBAWI; MOHAMMED; ALZAWI, 2017).

A typical structure of a Convolutional Neural Network for classification comprises three essential layers, as outlined in (ALBAWI; MOHAMMED; ALZAWI, 2017): the Convolutional layer, the Pooling layer, and the Fully-connected layer. The convolution operation entails the application of a filter (or kernel) to the original image, presented as a three-dimensional array (height, width, and channels). This filter traverses the entire image, generating a new array with

updated values. This efficiency is a distinctive feature of CNN, as it processes only a portion of the data, optimizing computational resources. Fig 2.4 illustrates this process.

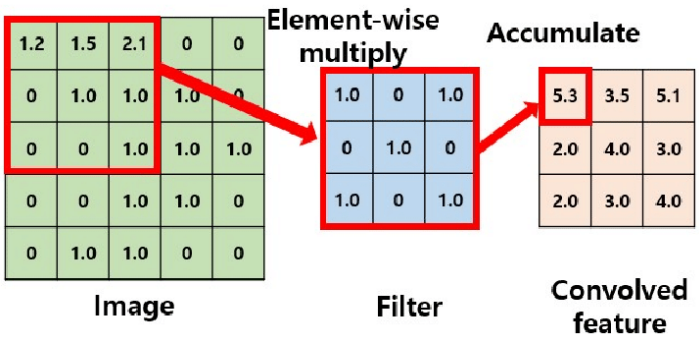


Figure 2.4: This is a representation of a convolution operation in a CNN (KIM et al., 2021). Each pixel inside the red block is multiplied by the corresponding value in the kernel, and all the values are added together. This sum is the value that remains in the convolved feature.

The pooling layer modifies the output of the convolutional layers before passing it to the fully connected layer. The objective is to replace the region of pixels with a summary statistic of nearby outputs (GOODFELLOW; BENGIO; COURVILLE, 2016). Max pooling, for example, returns the maximum output within a rectangular region (ZHOU; CHELLAPPA, 1988). Pooling reduces the impact of small translations in the input sample (GOODFELLOW; BENGIO; COURVILLE, 2016). Fig 2.5 illustrates the pooling operation.

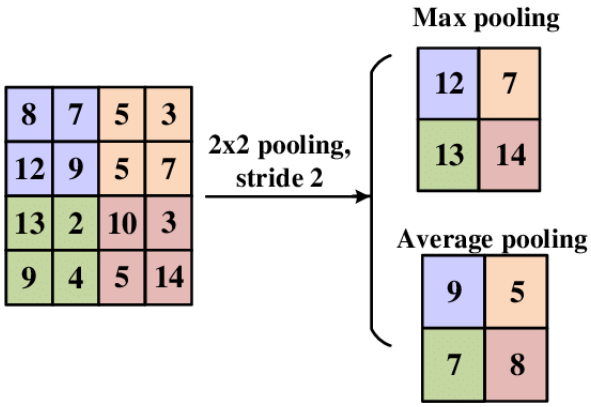


Figure 2.5: This represents a Pooling operation on the data. The table on the left represents a gray-scale image. The two tables on the right demonstrate different types of pooling layer, the table above utilizes the max value and the table below shows the average value., (YINGGE; ALI; LEE, 2020)

Finally, the fully connected layer establishes connections from all units in the preceding layer to produce a prediction/decision. When dealing with multiple classes, which is the case of this project, employing the softmax function becomes crucial, as it provides the probability distribution across different classes. Conversely, when addressing a binary classification task , a single output neuron with, the sigmoid function proves to be effective. Figure 2.6 shows an example of a CNN, VGG-16 (SIMONYAN; ZISSERMAN, 2014).

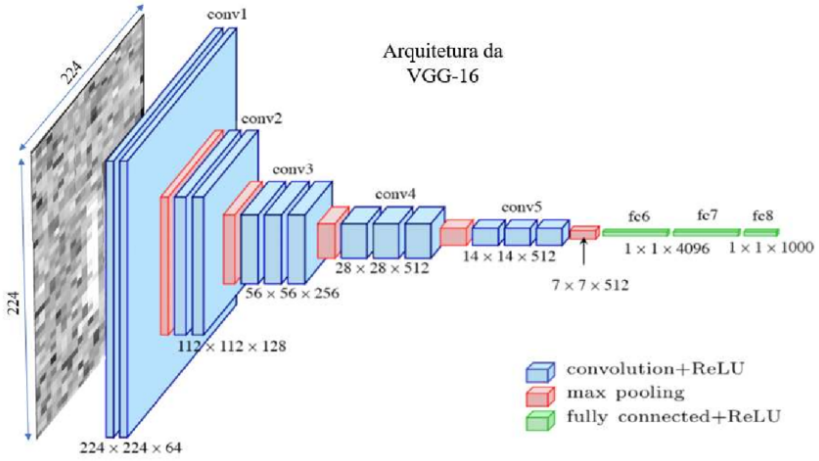


Figure 2.6: Illustration of the VGG-16 architecture (FALQUETO; PAES; PASSARO, 2023). The numbers below are the shape of each layer in the CNN.

2.1.4 Transfer Learning

Raghu et al. (2019) state that Transfer Learning, as its name suggests, involves the transference of knowledge within neural networks through the weights and biases of an initial Neural Network (NN) architecture. This involves utilizing an existing neural network architecture and fine-tuning its parameters on a target dataset using a pre-trained model. This practice has become widely prevalent in medical image applications. For instance, a sizable architecture trained on a diverse dataset like IMAGENET (DENG et al., 2009) can be repurposed for tasks such as analyzing Chest X-rays (RAGHU et al., 2019). As enunciated by (ZHUANG et al., 2021), transfer learning has gained prominence and holds significant promise in machine learning, owing to its broad range of application prospects.

In this study, transfer learning is employed to enhance classification outcomes. The selected architectures for this purpose include ResNet50 and ResNet101 (HE et al., 2016), EfficientNetB0, EfficientNetB4, and EfficientNetB7 (TAN; LE, 2019a), and VGG16 (SIMONYAN; ZISSERMAN, 2014). Subsequent sections will provide detailed explanations about these architectures and their utilization in the research.

2.2 MODELS

This section aims to elucidate the models selected for training. In the overload of available models, our decision-making process was anchored in the evaluation of their robustness. It's important to acknowledge that, due to constraints in computing power and data availability, opting for the most sophisticated CNN architecture may not guarantee optimal performance. As previously mentioned, our objective is to meticulously assess and compare the efficacy of various models with these specific medical datasets.

2.2.1 VGG

VGG, or the Visual Geometry Group, is known for its iconic and effective architecture in the field of convolutional neural networks (CNNs) (SIMONYAN; ZISSERMAN, 2014). VGG's approach focuses on depth and uses a uniform layout across its layers. VGG's architectures integrate max-pooling layers and rectification (ReLU) non-linearities in hidden layers. Notably, VGG does not use Local Response Normalisation (LRN) to increase memory consumption without commensurate performance gains. Fig 2.6 represents the VGG16 architecture model.

VGG's unique approach is its consistent use of small receptive fields (3×3) throughout the entire network. VGG's strategy of stacking multiple 3×3 convolution layers without spatial pooling effectively achieves larger effective receptive fields while maintaining computational efficiency. This departure from conventional architectural choices highlights VGG's commitment to simplicity, yet its impact has been profound, influencing subsequent developments in deep learning architectures (SIMONYAN; ZISSERMAN, 2014).

2.2.2 ResNet

The ResNet was designed to tackle the challenges of training deeper networks. It incorporates shortcut connections, also known as skip connections or residual connections, which enable the smooth flow of information through the network (HE et al., 2016). By introducing these residual connections, certain layers can be bypassed, which helps to mitigate the vanishing gradient problem and alleviate the degradation issue that arises when training deeper networks. The vanishing gradient is when the calculation of the gradient, in the backpropagation algorithm, has minimal variation causing a saturation in the parameters (weights and bias), caused by the depth of the neural network (HOCHREITER; SCHMIDHUBER, 1997; AMANATULLAH, 2023). The core concept behind ResNet is to create shortcuts across pairs of 3×3 filters, resulting in a unique architecture that fundamentally transforms the approach to building deep neural networks.

The ResNet family has a range of different models, including 18-layer, 34-layer, 50-layer, 101-layer, and 152-layer configurations. The 50-layer ResNet introduced bottleneck blocks, while the 101-layer and 152-layer ResNets increased depth, showcasing the flexibility and scalability of the ResNet architecture. Figure 2.7 shows the process of ResNet50. Bottleneck layers compresses the input data, reducing computational cost by lowering dimensionality (REHMAN, 2024). The advantages of depth, such as better accuracy and generalization, make ResNet a robust model family, and the 152-layer ResNet achieved first place in the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), (HE et al., 2016).

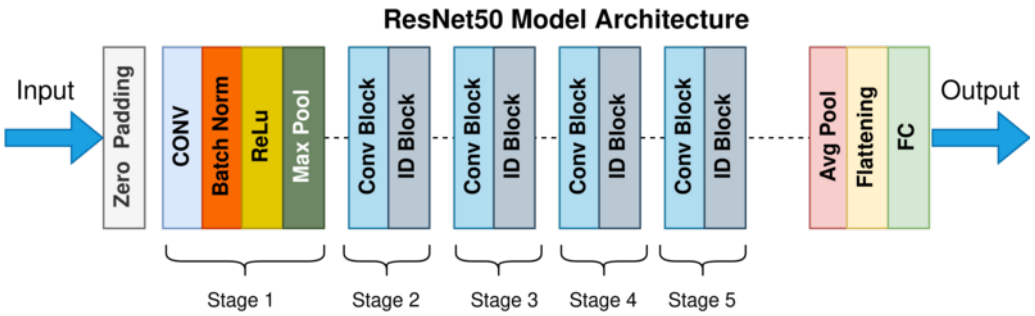


Figure 2.7: This is a simple representation of ResNet50 architecture (MUKHERJEE, 2022). Flattering is a operation to convert an array of n-dimension to one dimensional.

2.2.3 EfficientNet

The EfficientNet is a family of neural network architectures that represents a significant milestone in achieving optimal trade-offs between model accuracy and computational efficiency. It was introduced through the lens of neural architecture search, and it is designed to offer superior performance compared to its predecessors, particularly in terms of accuracy and computational speed (TAN; LE, 2019b). The key principle behind EfficientNet is the strategic scaling of baseline networks through a systematic approach, resulting in a family of models denoted by different scales, such as EfficientNet-B0 to EfficientNet-B7.

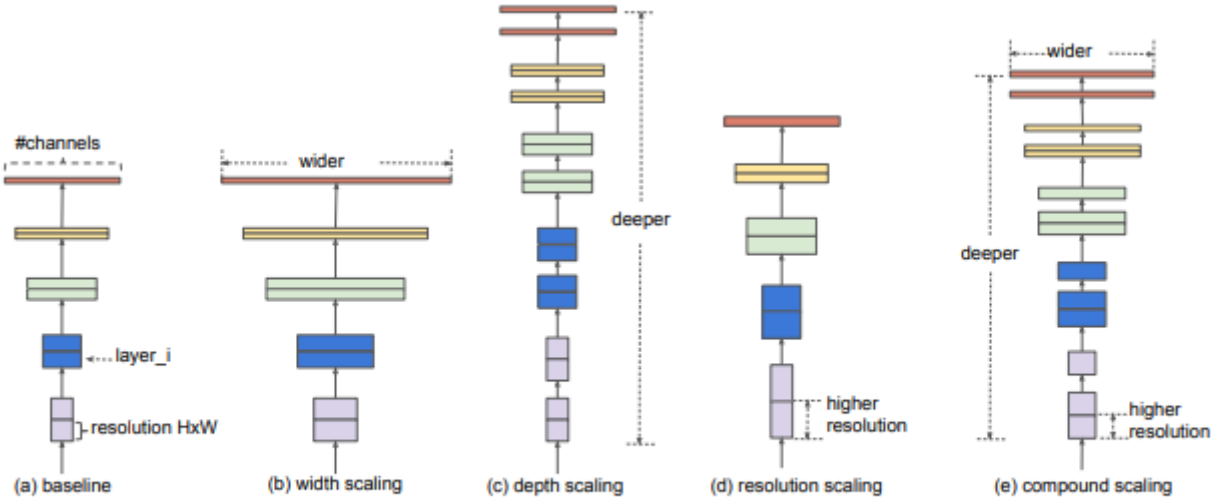


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Figure 2.8: This shows the differences of Efficient net (TAN; LE, 2019a)

Figure 2.8 illustrates the relationship between EfficientNet and other CNN models. EfficientNet employs compound scaling, which simultaneously scales the network wider and deeper.

Each member of the EfficientNet family is a result of a careful balance achieved by scaling up the baseline architecture, leading to remarkable improvements in accuracy while maintaining

efficiency. Notably, the EfficientNet-B7, the flagship model of this family, has achieved state-of-the-art performance on benchmark datasets like ImageNet, attaining an exceptional 84.3% top-1 accuracy (TAN; LE, 2019b), as shown in Fig 2.9

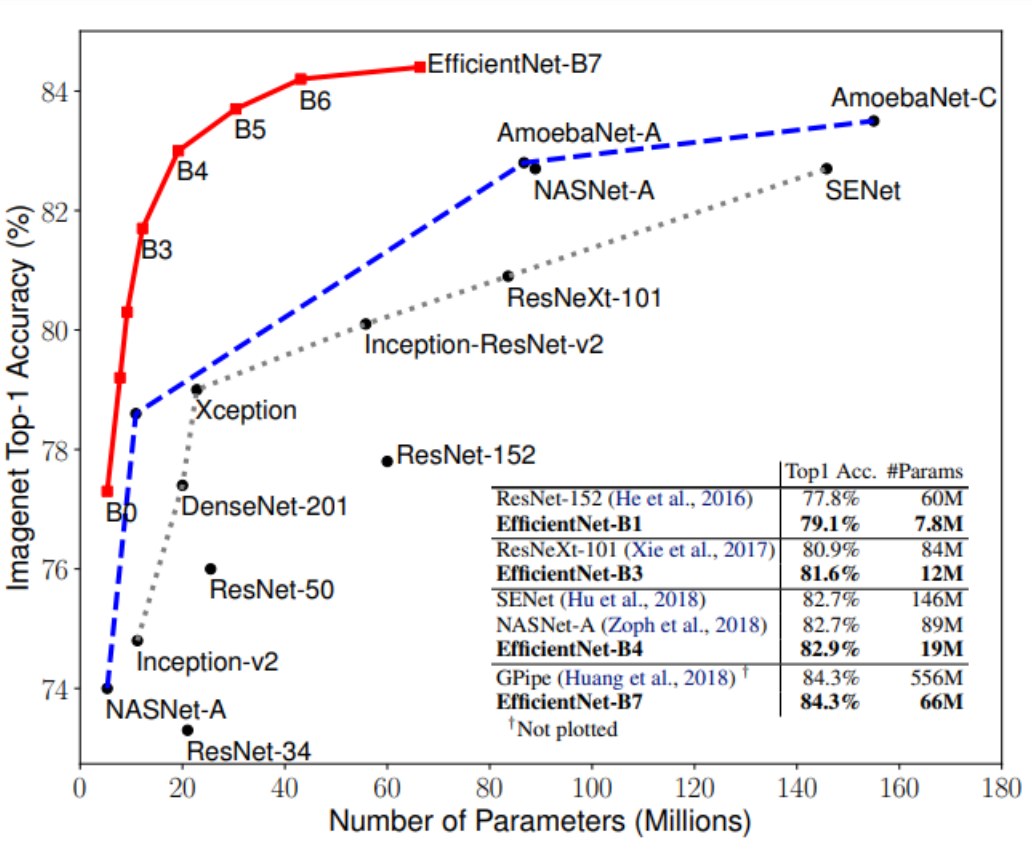


Figure 2.9: Comparison of models trained with Imagenet dataset, including EfficientNets and ResNet50, based on accuracy and number of parameters. Figure 1 from Tan e Le (2019b)

2.3 METRICS

This section aims to introduce the metrics that will be employed to evaluate all the models discussed herein. These metrics are crucial for enhancing not only the theoretical aspects of the models but also their practical applicability in real-world scenarios. The metrics under consideration include Accuracy, Precision, Recall, Sensitivity, Specificity, and F1-Score.

Accuracy is the most basic metric which is seen in DL, its formula is:

$$Accuracy = \frac{\text{Correct predictions}}{\text{All predictions}}. \tag{2.2}$$

It is possible to define the accuracy in terms of true and false positives (TP and FP) and true and false negatives (TN and FN), where true positives are the number of correct positive predictions, false positives are the number of incorrect positive predictions, true negatives are the number of

correct negative predictions, and false negatives are the number of incorrect negative predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.3}$$

Precision measures how often the model correctly predicts the positive class:

$$Precision = \frac{TP}{TP + FP}. \tag{2.4}$$

Precision is the rate of true positive results divided by the number of all positive results, including those not correctly classified. If the precision is set to 1, then all records may be considered relevant (ALPAYDIN, 2020).

Recall, also known as sensitivity or true positive rate, is the number of true positive results divided by the number of all relevant samples (the sum of true positives and false negatives) (ALPAYDIN, 2020).

$$Recall = \frac{TP}{TP + FN}. \tag{2.5}$$

Specificity is the ability of a classification model to correctly identify true negative instances or each available category (ALPAYDIN, 2020). It is calculated using the formula:

$$Specificity = \frac{TN}{TN + FP}. \tag{2.6}$$

The F1-Score is a metric that provides a balance between precision and recall and is particularly useful when there is an uneven class distribution. The formula for F1-Score is:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{2.7}$$

All these metrics are relative to each class worked. Each class has its own TP, TN, FP, and FN.

3 MATERIAL AND METHODS

This chapter involved a comprehensive search for medical datasets that met the specific requirements and were suitable for the intended purpose, and an evaluation of various neural networks to ensure that they met the criteria for this project.

The evaluation process included a review of the dataset's source, format, size, and content. The datasets were also evaluated based on their relevance to the project's objectives and the data quality.

3.1 DATASET

As previously said, this project works with two datasets: The HAM10000, and the COVID dataset.

3.1.1 The HAM10000

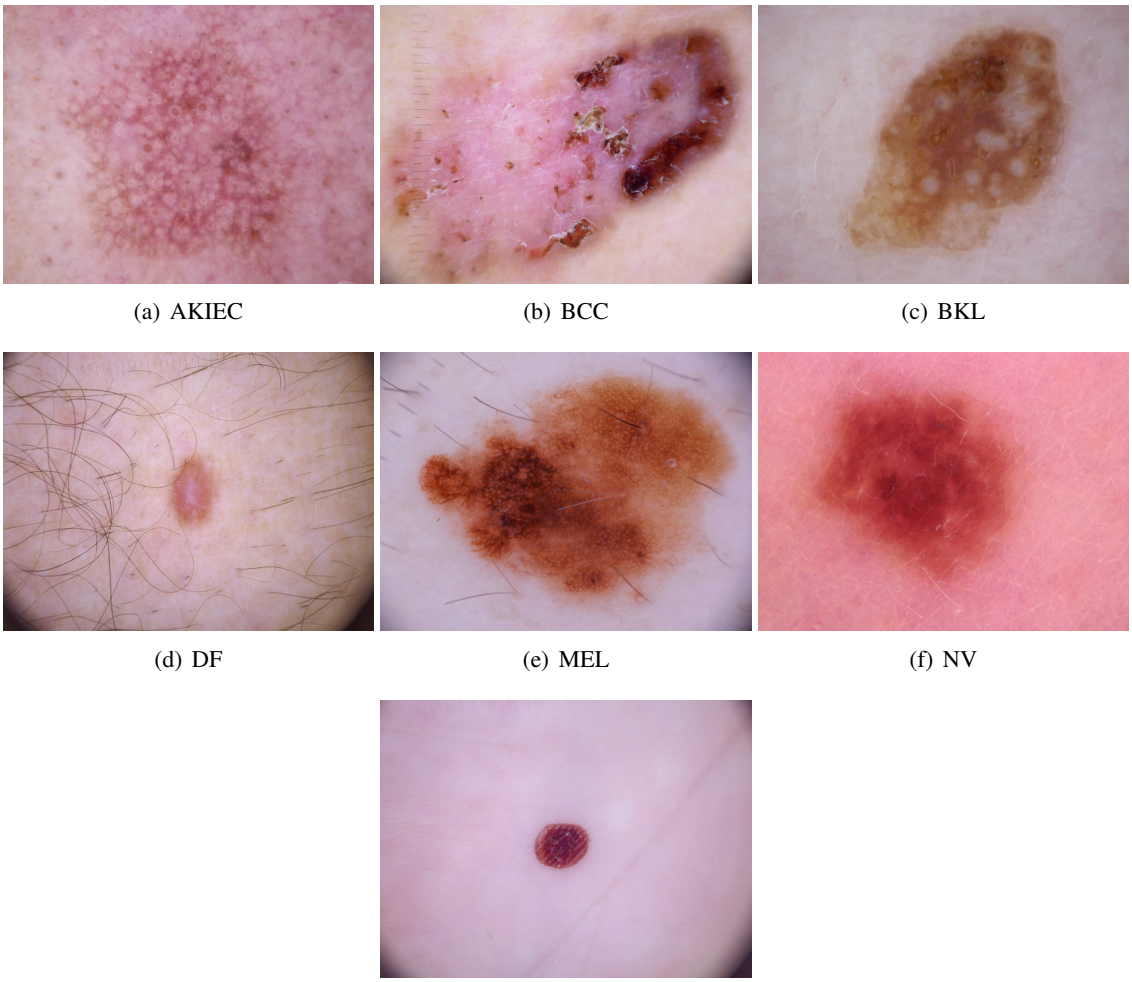
The HAM10000 dataset is a highly informative collection of dermatoscopic images gathered from a diverse range of populations (TSCHANDL, 2018). It comprises 10015 images for training and 1511 for testing each image has a resolution of 460 x 600 pixels. This dataset has 7 classes of skin lesions associated with different diseases/causes: Actinic Keratoses and Intraepithelial Carcinoma / Bowen's Disease (AKIEC), Basal Cell Carcinoma (BCC), Benign Keratosis-like Lesions (BKL), Dermatofibroma (DF), Melanoma (MEL), Melanocytic Nevi (NV) and Vascular Lesions (VASC) (TSCHANDL, 2018).

The dataset is imbalanced. The distribution is shown in Table 3.1.

Table 3.1: Class Distribution Percentage

Class	Percentage (%)	Absolute value
NV	76	7613
MEL	12	1316
BKL	13	1284
BCC	6	607
AKIEC	3.7	159
VASC	1.7	370
DF	1.6	177

All data of the HAM10000 dataset are deposited at the Harvard Dataverse in the following Link. An example of the images is shown in Figure 3.1.



(a) AKIEC (b) BCC (c) BKL

(d) DF (e) MEL (f) NV

(g) VASC

Figure 3.1: Images from HAM10000 Dataset, showing (a) Actinic Keratoses and Intraepithelial Carcinoma / Bowen's Disease (AKIEC), (b) Basal Cell Carcinoma (BCC), (c) Benign Keratosis-like Lesions (BKL), (d) Dermatofibroma (DF), (e) Melanoma (MEL), (f) Melanocytic Nevi (NV), and (g) Vascular Lesions (VASC).

3.1.2 COVID dataset

The COVID dataset is a compilation from several reputable sources, as said by Nascimento et al. (2023). This dataset spans a wide spectrum of well-known diseases, categorizing images into four distinct classes: Viral Pneumonia, COVID-19, Lung Opacity, and Normal. It is noteworthy that, similar to the HAM10000 dataset, the COVID dataset exhibits an imbalance classes.

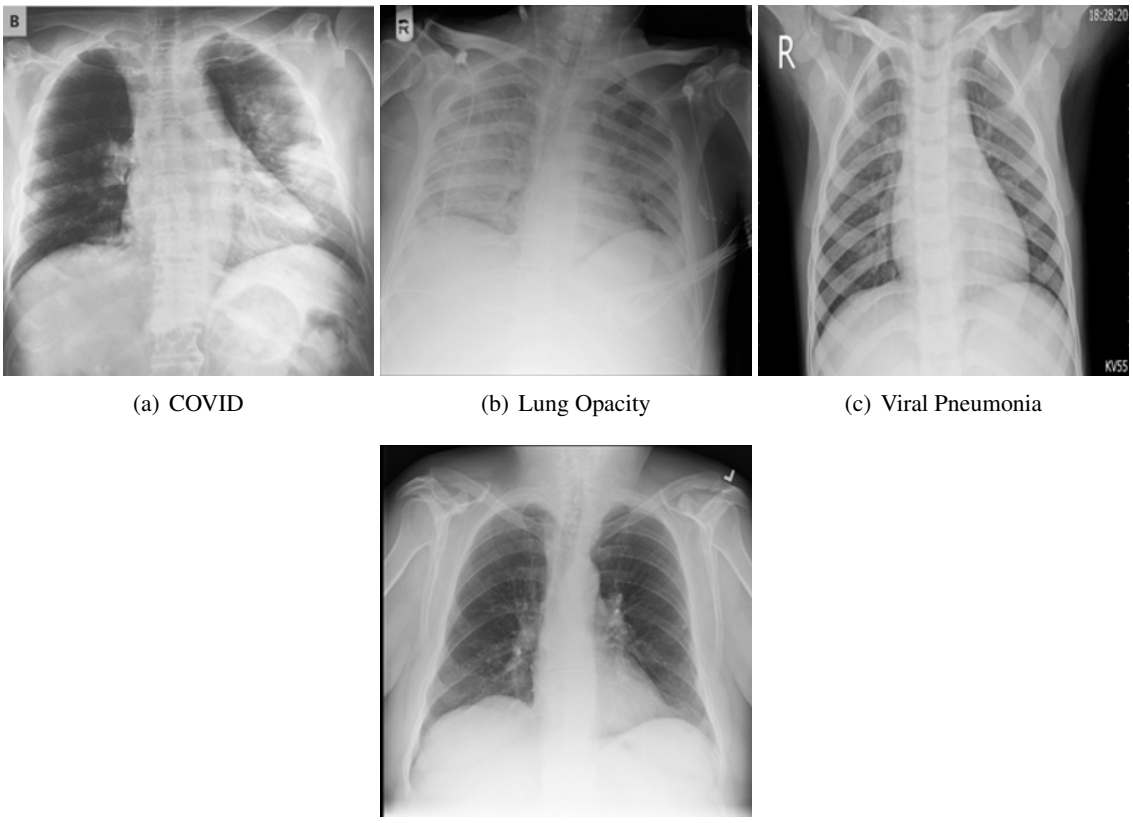
The images within this dataset are encoded in RGB format. Each image maintains a resolution of 299 x 299 pixels. Importantly, the dataset has not been pre-divided for training and testing purposes, as this segmentation is part of the ongoing project.

The dataset is imbalanced. The distribution is shown in Table 3.2.

Table 3.2: Class Distribution Percentage

Class	Percentage (%)	Absolute value
Normal	6	10192
Lung Opacity	13	6012
COVID-19	76	3616
Viral Pneumonia	12	1345

An example of the images is shown in Figure 3.2.



(a) COVID (b) Lung Opacity (c) Viral Pneumonia

(d) Normal

Figure 3.2: COVID Dataset Images, in which (a) COVID, (b) Lung Opacity, (c) Viral Pneumonia, and (d) Normal.

3.2 PREPROCESSING

3.2.1 HAM10000

The dataset was already divided into train and test sets, and a CSV file with the name, path, and label was already created. The validation set had to be split, thus it was chosen 10% of the train set to be the validation data. Table 3.3 shows the quantity for each set.

Table 3.3: Number of images per class in the training, validation, and test datasets

Class	Training	Validation	Test
NV	6034	671	908
MEL	1001	112	171
BKL	989	110	217
BCC	462	52	93
AKIEC	294	33	43
VASC	127	15	35
DF	103	12	44

3.2.2 COVID

The dataset comprises individual folders, each containing images of a specific disease. Thus, the processing must be capable of manipulating this form of data. Prior to data partitioning, a dataset variable (pandas.DataFrame) is created to include the name, path, and label. This facilitates easier data manipulation and manual verification, if necessary.

The split percentage chosen for this dataset was 80% of the images for train, 10% for test, and 10% for validation. Table 3.4 shows the number of images for each class. All this proportion split has to be verified because all the analysis, training, and evaluation depend on it.

Table 3.4: Number of images per class in the training, validation, and test datasets

Class	Training	Validation	Test
Normal	8153	1020	1019
Lung Opacity	4809	602	601
COVID-19	2892	362	362
Viral Pneumonia	1076	135	134

3.3 ANALYSIS

Before exploring into data processing, a comprehensive understanding of relevant aspects is crucial. When employing a Neural Network, standardizing the data format is paramount. This involves examining key metrics such as maximum and minimum pixel values, standard deviation, the average size of images, and the prevalence of images conforming to this average shape. Notably, this average size serves as the input dimension for the Neural Network. This section provides insights into the datasets and their inherent details, as discussed earlier in Section 3.1.

The pseudocode provided in Algorithm 1 and Algorithm 2 illustrates the logic underlying the analysis process, including the data and image statistics.

Algorithm 1 Data Analysis and Dataset Statistics

procedure ANALYZEDATASET($dataset$)
    Input: $dataset$ - DataFrame with images and labels
    for $image$ in $dataset$ **do**
        Step 1: Analyze Image
            Compute standard deviation, max, min pixel values, and average of channels
            Print the analysis results
        Step 2: Check Image Existence

        if $image$ not in $dataset$ **then**
                Print message: "Image not found in the dataset"
        end if
    end for
    Step 3: Count Labels and Print Statistics

        Count the number of occurrences for each label in the dataset
        Print the total number of images for each label
        Calculate and print the total and average shape of the images
end procedure

Algorithm 2 Main Routine

Main:
    ANALYZEDATASET($train\_dataset$) ▷ For Train Dataset
    ANALYZEDATASET($test\_dataset$) ▷ For Test Dataset
    ANALYZEDATASET($validation\_dataset$) ▷ For Validation Dataset

This algorithm outputs the smallest and largest pixel values, total images processed, and channel statistics (average and standard deviation) for the R, G, and B channels. It provides valuable insights into pixel values and their distribution, which is useful for image processing. Table 3.5 shows the values for each dataset.

Table 3.5: Channel Values for COVID and HAM10000 Datasets. The μ is the mean and σ the standard deviation

Channel	**Values - COVID**	**Values - HAM10000**
R - $\mu \pm \sigma$	193 ± 24	193 ± 22
G - $\mu \pm \sigma$	141 ± 31	139 ± 30
B - $\mu \pm \sigma$	147 ± 35	145 ± 33
Other Outputs		**Value**
Smallest pixel value		0
Largest pixel value		255

These values span the range observed across the training, validation, and test sets. The average and the standard deviation show that the image content is significant in all channels of the images, and doesn't have a duplicate channel. All code related to this work is available in the GitHub repository *LucasLessa1/medical_images_models* [1].

[1] <https://github.com/LucasLessa1/medical_images_models>

3.4 MODEL PREPARATION

In this section, we will prepare the data for input into a neural network model. This involves several considerations, including the specification of the loss function to be used, the selection of appropriate weights, and the formatting of the input data. The model is saved in *.pth*, pytorch format.

3.4.1 Loss

In multi-class scenarios, the commonly employed loss function is the Cross-entropy loss (ALPAYDIN, 2020):

$$E\left(\Theta \mid \chi\right) = -\sum_{n}\sum_{i} r_i^n \log y_i^n, \tag{3.1}$$

where Θ denotes the set of parameters/weights associated with the neural network, χ represents the current batch of examples, y_i^n is the probality estimate given by the NN for the i-th class of the n-th batch instance, while r_i^n represents the respective ground truth.

To handle imbalanced data, a good option is to use cost-sensitive learning, which adjusts the weight of each instance based on the cost of misclassifying its class. This maintains the total weight of the training set, ensuring that the impact of each instance reflects its importance (TING, 1998).

An algorithm will be utilized to determine suitable weights for loss calculation. This algorithm is based on the logic outlined in Algorithm 3 as specified in (REZAEI-DASTJERDEHEI; MIJANI; FATEMIZADEH, 2020).

Algorithm 3 Calculate Weights

procedure CALCULATEWEIGHTS($train\_df, label\_dict, dict\_train\_qntd$)
  Input: $train\_df$ - DataFrame with training data
    $label\_dict$ - Dictionary mapping labels to their indices
    $dict\_train\_qntd$ - Dictionary containing label occurrences in the training set
  Step 1: Initialize Variables
    $num\_classes \leftarrow$ Number of unique classes in $label\_dict$
    $total\_samples \leftarrow$ Total number of samples in $train\_df$
    $weights \leftarrow []$
  Step 2: Calculate Weights for Each Class
    $weights \leftarrow$ Empty list
    Function: CalculateWeight($total\_samples, label\_count, num\_classes$)
      $weight \leftarrow \frac{total\_samples}{label\_count \times num\_classes}$
      Return $weight$
      Append $weight$ to $weights$
  Step 3: Return Computed Weights
    return $weights$
end procedure

The calculation for each weight is formulated in Eq. 3.2 as follows (REZAEI-DASTJERDEHEI; MIJANI; FATEMIZADEH, 2020).

$$\text{Weight} = \frac{\text{Total number of samples in the dataset}}{\text{Quantity of samples per label} \times \text{Number of labels}} \tag{3.2}$$

3.4.2 Model input

Both datasets in this project are represented in RGB, necessitating that the CNN model's input consist of three channels. Ensuring consistency in data format during preprocessing stages is crucial. As emphasized throughout this study, minimizing extraneous data improves the performance of the neural network model. To achieve this, Algorithm 1 was implemented to validate data consistency, and the results are shown on Section 3.3.

3.5 MODEL

This project uses the Transfer Learning method, as described in 2.1.4. It involves taking a pre-trained model, which in this case was trained on the ImageNet dataset (DENG et al., 2009), and fine-tuning it using our data. Specifically, we import a pre-trained CNN architecture and then fine-tune the model using our data, doing so separately for each dataset.

For each model employed, the initial layer is adjusted to accommodate the precise number of input channels. Simultaneously, the concluding layer undergoes a linear transformation, incorporates the Rectified Linear Unit (ReLU) activation function, a dropout layer, and is ultimately tailored to output, through a softmax layer, the specific number of classes relevant to the given task.

Overfitting occurs when a model performs exceptionally well on the training data but exhibits poor performance when evaluated on new, unseen samples. To mitigate overfitting and enhance the model's robustness, a dropout layer is implemented. This layer selectively deactivates a fraction of neurons during training, discouraging the network from relying excessively on specific features. The goal is to foster a more resilient and generalized model, promoting improved overall performance.

This project employed the VGG-16, ResNet50, ResNet101, EfficientNetB0, EfficientNetB4, and EfficientNetB7 architectures, which were described in Chapter 2. Size and parameters information of these models are presented in Table 3.6

Table 3.6: Neural Network Model Comparison

Model	Total Layers	Total Params	Trainable Params	Input Size (MB)	Estimated Total Size (MB)
VGG16	43	136,361,799	136,361,799	3.09	1696.87
ResNet50	178	24,560,711	24,560,711	3.09	1666.21
ResNet101	348	43,552,839	43,552,839	3.09	2543.63
EfficientB0	280	4,016,515	4,016,515	3.09	1132.10
EfficientB4	548	5,288,548	5,288,548	3.09	786.99
EfficientB7	931	63,804,887	63,804,887	3.09	6928.38

An important aspect is the Hyperparameters, which are architecture-level parameters (CHOLLET, 2022), in other words, they are external configurations for the learning and training. In this work we didn't change their values for each simulation. The Hyperparameters used are on Table 3.7

Table 3.7: Optimization Hyperparameters

Hyperparameters	**Value**
Optimizer	Adam
Weight Decay	1×10^{-5}
Learning Rate	1×10^{-4}
Lr Scheduler	
Function	Step LR
Patience	7
Gamma	0.5

To determine the optimal learning rate, various tests were conducted. The most effective value identified was 1×10^{-4}. This learning rate was validated through fine-tuning using the datasets employed in this study. Regarding the optimizer, Adam was selected based on the findings of Yaqub et al. (2020), which demonstrate that Adam performs well in classification problems.

4 RESULTS

Due to the large size of the datasets, the initial training was constrained to 10 epochs across the five different models: VGG16, ResNet50, ResNet101, EfficientNetB0, EfficientNetB4 and EfficientNetB7. For the second phase, the top three models—selected based on the evaluation metrics discussed in Section 2.3, were trained for 20 epochs. Finally, the best-performing model from the second phase was trained for an extended period of 50 epochs. This process was repeated for each dataset, ensuring a consistent evaluation approach.

4.1 DATASET HAM10000

4.1.1 10 epochs

Tables 4.1-4.6 present the detailed results in 10 epochs for each class and for each model using HAM10000 dataset.

Table 4.1: Performance Metrics for VGG16

Class	Precision	Recall	F1-Score	Specificity
nv	0.960	0.503	0.660	0.968
mel	0.237	0.731	0.358	0.700
bkl	0.311	0.304	0.308	0.887
bcc	0.310	0.290	0.300	0.958
akiec	0.149	0.302	0.200	0.950
vasc	0.489	0.629	0.550	0.984
df	0.182	0.318	0.231	0.957
Mean	0.377	0.440	0.373	0.915
Accuracy	0.479			

Table 4.1 highlights that the Precision for class “nv” is the highest among all classes. However, the Precision for the remaining classes is considerably low, with an average of 37%. Both Recall and F1-score are also notably low. In contrast, specificity shows a favorable result, with an average of 91.5%.

Table 4.2: Performance Metrics for ResNet50

Class	Precision	Recall	F1-Score	Specificity
nv	0.917	0.783	0.845	0.894
mel	0.462	0.708	0.559	0.895
bkl	0.599	0.737	0.661	0.917
bcc	0.600	0.613	0.606	0.973
akiec	0.478	0.256	0.333	0.992
vasc	0.808	0.600	0.689	0.997
df	0.365	0.523	0.430	0.973
Mean	0.604	0.603	0.589	0.949
Accuracy	0.731			

The precision achieved by ResNet50 (Table 4.2) is twice that of VGG-16 (Table 4.1), with classes “nv” and “vasc” demonstrating particularly strong results. The Recall and F1-score both remain around 60%. An accuracy of 73% and a specificity mean of 94.9% indicate a relatively good performance.

Table 4.3: Performance Metrics for ResNet101

Class	Precision	Recall	F1-Score	Specificity
nv	0.922	0.689	0.789	0.912
mel	0.372	0.719	0.490	0.845
bkl	0.551	0.668	0.604	0.909
bcc	0.523	0.602	0.560	0.964
akiec	0.533	0.372	0.438	0.990
vasc	0.435	0.771	0.557	0.976
df	0.436	0.386	0.410	0.985
Mean	0.539	0.601	0.550	0.940
Accuracy	0.668			

Compared to the results from ResNet50 (Table 4.2), the results obtained with ResNet101 (Table 4.3) were relatively lower. The accuracy decreases by 7% compared to ResNet50, which has a smaller architecture.

Table 4.4: Performance Metrics for EfficientNetB0

Class	Precision	Recall	F1-Score	Specificity
nv	0.899	0.937	0.918	0.841
mel	0.667	0.596	0.630	0.962
bkl	0.767	0.714	0.740	0.964
bcc	0.844	0.699	0.765	0.992
akiec	0.540	0.628	0.581	0.984
vasc	0.906	0.829	0.866	0.998
df	0.740	0.841	0.787	0.991
Mean	0.766	0.749	0.755	0.962
Accuracy	0.838			

Table 4.4 shows that the Accuracy of 83.8% and the mean Specificity of 96.2% is the highest

among the models presented here. However, the classes “mel” and “akiec” exhibited the lowest Precision and Recall, indicating that the model struggled to correctly predict positive samples for these classes.

Table 4.5: Performance Metrics for EfficientNetB4

Class	**Precision**	**Recall**	**F1-Score**	**Specificity**
nv	0.907	0.921	0.914	0.857
mel	0.632	0.684	0.657	0.949
bkl	0.801	0.705	0.750	0.971
bcc	0.813	0.796	0.804	0.988
akiec	0.660	0.721	0.689	0.989
vasc	0.725	0.829	0.773	0.993
df	0.857	0.682	0.759	0.997
Mean	0.771	0.762	0.764	0.963
Accuracy	0.841			

EfficientNetB4 metrics (Table 4.5) had the highest accuracy from all models trained with 10 epochs, an accuracy of 84.1%. The mean of the Precision, Recall and F1-score stays equal or higher than trained with EfficientNetB0.

Table 4.6: Performance Metrics for EfficientNetB7

Class	**Precision**	**Recall**	**F1-Score**	**Specificity**
nv	0.877	0.942	0.908	0.801
mel	0.739	0.515	0.607	0.977
bkl	0.815	0.668	0.734	0.974
bcc	0.778	0.828	0.802	0.984
akiec	0.569	0.860	0.685	0.981
vasc	0.568	0.714	0.633	0.987
df	0.903	0.636	0.747	0.998
Mean	0.750	0.738	0.731	0.958
Accuracy	0.831			

The largest architecture, EfficientNetB7, did not exhibit the best performance (Table 4.6). Its accuracy is lower than that of EfficientNetB0 (Table 4.4), yet it remains superior when compared to VGG-16 (Table 4.1), ResNet50 (Table 4.2), and ResNet101 (Table 4.3). Table 4.7 presents a summary analysis of all training with 10 epochs.

Table 4.7: Performance from each model in 10 epochs.

Models	Accuracy	Precision	Recall	F1-Score	Specificity
VGG16	0.479	0.310	0.318	0.308	0.957
ResNet50	0.731	0.599	0.613	0.606	0.973
ResNet101	0.668	0.523	0.668	0.557	0.964
EfficientNetB0	0.838	0.767	0.714	**0.765**	0.984
EfficientNetB4	**0.841**	**0.801**	**0.721**	0.759	**0.988**
EfficientNetB7	0.831	0.778	0.714	0.734	0.981
Best	EfficientNetB4	EfficientNetB4	EfficientNetB4	EfficientNetB0	EfficientNetB4
Second Best	EfficientNetB0	EfficientNetB7	EfficientNetB0	EfficientNetB4	EfficientNetB0
Third Best	EfficientNetB7	EfficientNetB0	EfficientNetB7	EfficientNetB7	EfficientNetB7

Table 4.7 presents the top three models based on all metrics evaluated in this study. EfficientNetB4 ranked first overall, outperforming EfficientNetB0 in every metric except for the F1-score. EfficientNetB7 consistently maintained the third position across all metrics, except for Precision, where it ranked second.

Based on the aforementioned results, it was chosen the top 3 models for performing a new training session with 20 epochs, which are the EfficientNet family models.

4.1.2 20 epochs

As a reminder, the best models trained with 10 epochs were the EfficientNet family (Table4.7). In this subsection, they will be evaluated using training with 20 epochs. Tables 4.8 - 4.10 present the results in 20 epochs for EfficientNet family class.

Table 4.8: Performance Metrics for EfficientNetB0

Class	Precision	Recall	F1-Score	Specificity
nv	0.897	0.953	0.924	0.836
mel	0.718	0.462	0.562	0.977
bkl	0.768	0.719	0.743	0.964
bcc	0.805	0.710	0.754	0.989
akiec	0.652	0.698	0.674	0.989
vasc	0.560	0.800	0.659	0.985
df	0.661	0.841	0.740	0.987
Mean	0.723	0.740	0.722	0.961
Accuracy	0.835			

The results from EfficientNetB0 over 20 epochs, as shown in Table 4.8 are inferior to those obtained over 10 epochs (Table 4.4). The Accuracy decreased by 0.3%, with the most significant reduction observed in Precision, which declined by 4.3%.

Table 4.9: Performance Metrics for EfficientNetB4

Class	Precision	Recall	F1-Score	Specificity
nv	0.894	0.947	0.920	0.831
mel	0.727	0.544	0.622	0.974
bkl	0.803	0.733	0.766	0.970
bcc	0.841	0.796	0.818	0.990
akiec	0.551	0.628	0.587	0.985
vasc	0.780	0.914	0.842	0.994
df	0.711	0.727	0.719	0.991
Mean	0.758	0.756	0.753	0.962
Accuracy	0.845			

Different to the EfficientNetB0 (Table 4.8), results over 20 epochs in EfficientNetB4, Table 4.9, are superior to those obtained over 10 epochs (Table 4.5). The mean for all metrics increased, and Accuracy increased by 0.4%.

Table 4.10: Performance Metrics for EfficientNetB7

Class	Precision	Recall	F1-Score	Specificity
nv	0.889	0.950	0.919	0.821
mel	0.750	0.474	0.581	0.980
bkl	0.717	0.724	0.720	0.952
bcc	0.892	0.796	0.841	0.994
akiec	0.725	0.674	0.699	0.993
vasc	0.640	0.914	0.753	0.988
df	0.800	0.727	0.762	0.995
Mean	0.773	0.751	0.753	0.960
Accuracy	0.839			

The accuracy achieved by EfficientNetB7 after 20 epochs, as presented in Table 4.10, exhibits an improvement of 0.8% compared to the results obtained after 10 epochs (Table 4.6). Furthermore, all other evaluated metrics demonstrated an increase as well.

Table 4.11 compares the metrics of the EfficientNet models.

Table 4.11: Performance from EfficientNet model using 20 epochs.

Models	Accuracy	Precision	Recall	F1-Score	Specificity
EfficientNetB0	0.835	0.718	0.719	0.740	0.985
EfficientNetB4	**0.845**	**0.780**	**0.733**	**0.766**	0.985
EfficientNetB7	0.839	0.750	0.727	0.753	**0.988**
Best	EfficientNetB4	EfficientNetB4	EfficientNetB4	EfficientNetB4	EfficientNetB7
Second Best	EfficientNetB7	EfficientNetB7	EfficientNetB7	EfficientNetB7	EfficientNetB4
Third Best	EfficientNetB0	EfficientNetB0	EfficientNetB0	EfficientNetB0	EfficientNetB0

One can see that the EfficientNetB4 demonstrates improved performance, through the lens of all metrics, than the other variants, indicating that training for more epochs is advantageous, particularly for larger models like the B7.

4.1.3 50 epochs

Table 4.12 presents the results in 50 epochs for each class:

Table 4.12: Performance Metrics for EfficientNetB4

Class	Precision	Recall	F1-Score	Specificity
nv	0.906	0.930	0.917	0.854
mel	0.692	0.591	0.637	0.966
bkl	0.787	0.733	0.759	0.967
bcc	0.835	0.817	0.826	0.989
akiec	0.549	0.651	0.596	0.984
vasc	0.667	0.914	0.771	0.989
df	0.732	0.682	0.706	0.993
Mean	0.738	0.760	0.745	0.963
Accuracy	0.835			

The accuracy achieved by EfficientNetB4 after 50 epochs, as reported in Table 4.12, exhibits a decrease of 0.4% compared to the results obtained after 20 epochs (Table 4.10). Additionally, the F1-score demonstrates an increment of 0.4% relative to EfficientNetB7 trained for 20 epochs

4.1.4 Analysis of all training with HAM dataset

The following table indicate the best model for every training with the HAM dataset. The numbers after the model name specifies the number of training epochs.

Table 4.13: Performance Metrics - Part 1 (Accuracy to F1-Score)

Models	Accuracy	Precision	Recall	F1-Score
Best Model	EfficientNetB4 - 20	EfficientNetB7 - 20	EfficientNetB4 - 10	EfficientNetB4 - 10
Second Best Model	EfficientNetB4 - 10	EfficientNetB4 - 10	EfficientNetB4 - 50	EfficientNetB0 - 10
Third Best Model	EfficientNetB4 - 50	EfficientNetB0 - 10	EfficientNetB4 - 20	EfficientNetB4 - 20
Fourth Best Model	EfficientNetB7 - 20	EfficientNetB4 - 20	EfficientNetB7 - 20	EfficientNetB7 - 20
Fifth Best Model	EfficientNetB0 - 10	EfficientNetB7 - 10	EfficientNetB0 - 10	EfficientNetB4 - 50
Sixth Best Model	EfficientNetB0 - 20	EfficientNetB4 - 50	EfficientNetB0 - 20	EfficientNetB7 - 10
Seventh Best Model	EfficientNetB7 - 10	EfficientNetB0 - 20	EfficientNetB7 - 10	EfficientNetB0 - 20
Eighth Best Model	ResNet50 - 10	ResNet50 - 10	ResNet50 - 10	ResNet50 - 10
Ninth Best Model	ResNet101 - 10	ResNet101 - 10	ResNet101 - 10	ResNet101 - 10
Tenth Best Model	VGG16 - 10	VGG16 - 10	VGG16 - 10	VGG16 - 10

Table 4.14: Performance Metrics - Part 2 (Specificity to Sum of False Negatives)

Models	Specificity	False Negatives	Sum False Negatives
Best Model	EfficientNetB4 - 10	EfficientNetB4 - 20	EfficientNetB4 - 20
Second Best Model	EfficientNetB4 - 50	EfficientNetB4 - 10	EfficientNetB4 - 10
Third Best Model	EfficientNetB4 - 20	EfficientNetB4 - 10	EfficientNetB4 - 10
Fourth Best Model	EfficientNetB0 - 10	EfficientNetB7 - 20	EfficientNetB7 - 20
Fifth Best Model	EfficientNetB0 - 20	EfficientNetB0 - 10	EfficientNetB0 - 10
Sixth Best Model	EfficientNetB7 - 20	EfficientNetB0 - 20	EfficientNetB0 - 20
Seventh Best Model	EfficientNetB7 - 10	EfficientNetB7 - 10	EfficientNetB7 - 10
Eighth Best Model	ResNet50 - 10	ResNet50 - 10	ResNet50 - 10
Ninth Best Model	ResNet101 - 10	ResNet101 - 10	ResNet101 - 10
Tenth Best Model	VGG16 - 10	VGG16 - 10	VGG16 - 10

Upon reviewing Tables 4.13 and Table 4.14, and considering the validation losses of 0.262 after 10 epochs and 0.435 after 20 epochs, it is concluded that the optimal model is EfficientNetB4 trained over 10 epochs.

4.1.5 The Best model - HAM

Based on the overall ranking of the models, depicted in the aforementioned tables, it is possible to recognize as the best model for the HAM dataset the EfficientNetB4 when trained with 10 epochs, evaluated based on F1-Score. Table 4.15 shows again the results the model obtained in this scenario. Moreover, Fig. 4.1 presents the learning curve over the epochs.

Table 4.15: Performance Metrics for EfficientNetB4

Class	Precision	Recall	F1-Score	Specificity
nv	0.907	0.921	0.914	0.857
mel	0.632	0.684	0.657	0.949
bkl	0.801	0.705	0.750	0.971
bcc	0.813	0.796	0.804	0.988
akiec	0.660	0.721	0.689	0.989
vasc	0.725	0.829	0.773	0.993
df	0.857	0.682	0.759	0.997
Mean	0.771	0.762	0.764	0.963
Accuracy	0.841			

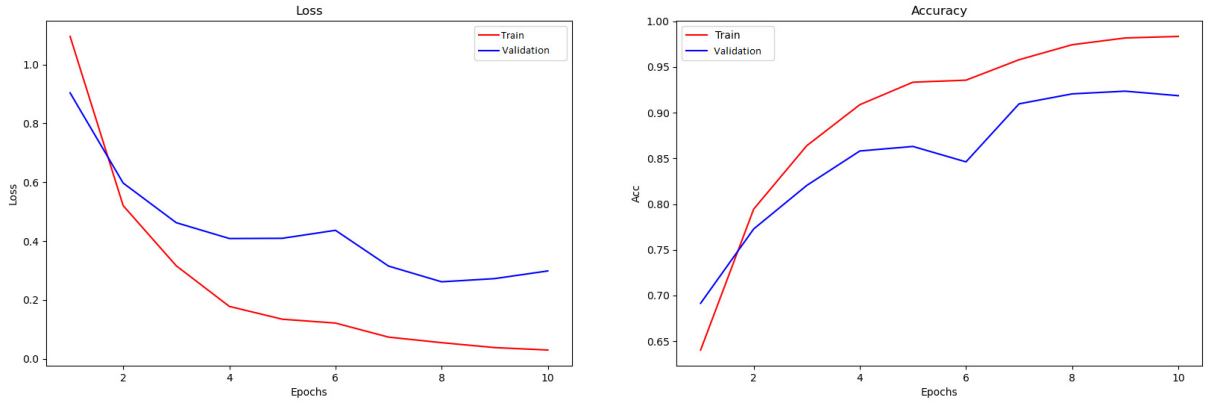


Figure 4.1: Curve of Accuracy and Loss from the EfficientNetB4 trained with 10 epochs. This model achieved an accuracy of 84%.

The performance metrics for the EfficientNetB4 model show variations across different classes. The "nv" class, representing Melanocytic Nevi, has the highest Precision (0.907), F1-Score (0.914), and Recall (0.921), indicating a high accuracy rate and effective detection of true positives.

Conversely, the "mel" class, representing Melanoma, exhibits the lowest Precision (0.632), F1-Score (0.657), and Recall (0.684), suggesting that the model has difficulty on accurately identifying this class.

The "bkl" class, representing Benign Keratosis, has a good Precision (0.801) but moderate Recall (0.705), implying a balanced performance. The "bcc" class, representing Basal Cell Carcinoma, demonstrates strong metrics across the board, with high Precision (0.813), F1-Score (0.804), and Specificity (0.988).

Actinic Keratosis ("akiec") has a high Specificity (0.989), indicating a low rate of false positives, but lower values for Precision (0.660), Recall (0.721), and F1-Score (0.689), suggesting that this class could be more refined.

Vascular Lesion ("vasc") and Dermatofibroma ("df") exhibit moderate Precision, with the latter showing high Specificity (0.997), indicating a reliable identification rate, though "df" has lower Recall (0.682).

In summary, while the EfficientNetB4 model shows strong performance across certain classes, some areas require improvement, particularly in the detection of challenging classes like Melanoma ("mel"). These insights can guide further model refinement and training to achieve a more balanced and robust performance.

To view the loss curves for the other models, please refer to the Appendix.

4.2 THE COVID DATASET

4.2.1 10 epochs

Tables 4.16-4.21 present the detailed results in 10 epochs for each class and for each model using COVID dataset.

Table 4.16: Performance Metrics for VGG16

Class	Precision	Recall	F1-Score	Specificity
Normal	0.952	0.899	0.925	0.958
Lung Opacity	0.854	0.925	0.789	0.937
COVID	0.957	0.975	0.797	0.991
Viral Pneumonia	0.970	0.970	0.970	0.998
Mean	0.933	0.942	0.870	0.971
Accuracy	0.924			

Table 4.16 presents the results for VGG-16, with an accuracy of 92.4% and all other metrics exceeding 90%, except for the F1-Score mean, which is 87%, still a commendable result. The VGG-16 model (Table 4.1) performed worse when trained with the HAM10000 dataset compared to its performance with the COVID dataset, as shown in Table 4.16.

Table 4.17: Performance Metrics for ResNet50

Class	Precision	Recall	F1-Score	Specificity
Normal	0.940	0.969	0.954	0.943
Lung Opacity	0.950	0.894	0.921	0.982
COVID	0.975	0.978	0.977	0.995
Viral Pneumonia	0.971	1.000	0.985	0.998
Mean	0.959	0.960	0.959	0.979
Accuracy	0.951			

The results obtained with ResNet50, as presented in Table 4.17, are superior to those of the VGG-16 model 4.16. All other metrics exceed 95.9%. The accuracy is 2.7% higher, and the F1-Score, which was 87% for VGG-16, is 95.9% for ResNet50.

Table 4.18: Performance Metrics for ResNet101

Class	Precision	Recall	F1-Score	Specificity
Normal	0.939	0.965	0.952	0.942
Lung Opacity	0.955	0.890	0.922	0.983
COVID	0.981	0.992	0.986	0.996
Viral Pneumonia	0.937	1.000	0.968	0.995
Mean	0.953	0.962	0.957	0.979
Accuracy	0.950			

The accuracy achieved with ResNet101, as shown in Table 4.18, is 0.1% lower than that

achieved with ResNet50 (Table 4.17). Specificity remains consistent with that of ResNet50, while Precision is slightly lower, and consequently, the F1-Score is also marginally reduced.

Table 4.19: Performance Metrics for EfficientNetB0

Class	Precision	Recall	F1-Score	Specificity
Normal	0.929	0.960	0.944	0.932
Lung Opacity	0.946	0.872	0.907	0.980
COVID	0.973	0.989	0.981	0.994
Viral Pneumonia	0.943	0.993	0.967	0.996
Mean	0.948	0.953	0.950	0.976
Accuracy	0.942			

The results from ResNet50 (Table 4.17) are better than compared to the results from EfficientNetB0 presented in Table 4.19. The accuracy in EfficientNetB0 is 0.9% lower than ResNet50.

Table 4.20: Performance Metrics for EfficientNetB4

Class	Precision	Recall	F1-Score	Specificity
Normal	0.954	0.956	0.955	0.957
Lung Opacity	0.942	0.922	0.932	0.978
COVID	0.984	0.992	0.988	0.997
Viral Pneumonia	0.944	1.000	0.971	0.996
Mean	0.956	0.967	0.961	0.982
Accuracy	0.955			

Training with EfficientNetB4 yielded superior results compared to ResNet50, as demonstrated in Table 4.20 and Table 4.17. Only the Precision metric is slightly lower than that of ResNet50 by 0.3%; however, all other metrics are higher.

Table 4.21: Performance Metrics for EfficientNetB7

Class	Precision	Recall	F1-Score	Specificity
Normal	0.954	0.967	0.960	0.956
Lung Opacity	0.949	0.922	0.935	0.980
COVID	0.989	0.994	0.992	0.998
Viral Pneumonia	0.985	0.993	0.989	0.999
Mean	0.969	0.969	0.969	0.983
Accuracy	0.960			

The model that achieved the highest accuracy is EfficientNetB7, as shown in Table 4.21, with an accuracy of 96%. Additionally, all other metrics are higher as well.

Table 4.22 presents a summary analysis of all models.

Table 4.22: Performance Comparison of Different Models

Models	Accuracy	Precision	Recall	F1-Score	Specificity
VGG16	0.92391	0.93326	0.94233	0.87008	0.97106
ResNet50	0.95085	0.95917	0.96000	0.95926	0.97924
ResNet101	0.95038	0.95304	0.96164	0.95675	0.97916
EfficientNetB0	0.94187	0.94768	0.95328	0.94987	0.97552
EfficientNetB4	0.95510	0.95584	0.96734	0.96135	0.98181
EfficientNetB7	**0.96030**	**0.96909**	**0.96886**	**0.96891**	**0.98329**
Higher	EfficientNetB7	EfficientNetB7	EfficientNetB7	EfficientNetB7	EfficientNetB7
Second Higher	EfficientNetB4	ResNet50	EfficientNetB4	EfficientNetB4	EfficientNetB4
Third Higher	ResNet50	EfficientNetB4	ResNet101	ResNet50	ResNet50

Based on the aforementioned results, it was choosen the top 3 models: EfficientNetB7, EfficientNetB4, and ResNet50.

4.2.2 20 epochs

As a reminder, the best models trained with 10 epochs were the ResNet50, EfficientNetB4, and EfficientNetB7 (Table 4.22). In this subsection, they will be evaluated using training with 20 epochs.

Table 4.23-4.25 presented the results with 20 epochs of the 3 models chosen.

Table 4.23: Performance Metrics for ResNet50

Class	Precision	Recall	F1-Score	Specificity
Normal	0.940	0.971	0.955	0.943
Lung Opacity	0.957	0.892	0.923	0.984
COVID	0.986	0.997	0.992	0.997
Viral Pneumonia	0.964	0.993	0.978	0.997
Mean	0.962	0.963	0.962	0.980
Accuracy	0.954			

The performance of the ResNet model trained for 20 epochs, as depicted in Table 4.23, exhibits improvements compared to the model trained for 10 epochs, as shown in Table 4.17. Specifically, the accuracy, recall, and precision metrics show an enhancement of 0.3%.

Table 4.24: Performance Metrics for EfficientNetB4

Class	Precision	Recall	F1-Score	Specificity
Normal	0.946	0.977	0.961	0.948
Lung Opacity	0.970	0.905	0.936	0.989
COVID	0.981	0.994	0.988	0.996
Viral Pneumonia	0.985	0.993	0.989	0.999
Mean	0.970	0.967	0.969	0.983
Accuracy	0.961			

The performance of EfficientNetB4 trained for 20 epochs, as presented in Table 4.24, surpasses the results obtained with 10 epochs (Table 4.19). The accuracy increased from 95.5% to 96.1%, an improvement of 0.6%, and all other metrics are higher, with the exception of Recall, which remains unchanged at 0.967.

Table 4.25: Performance Metrics for EfficientNetB7

Class	Precision	Recall	F1-Score	Specificity
Normal	0.944	0.984	0.964	0.946
Lung Opacity	0.975	0.902	0.937	0.991
COVID	0.984	0.989	0.986	0.997
Viral Pneumonia	0.993	0.993	0.993	0.999
Mean	0.974	0.967	0.970	0.983
Accuracy	0.962			

The performance of EfficientNetB7 trained for 20 epochs, as shown in Table 4.25, is notably impressive. An accuracy of 96.2%, an F1-Score of 97%, and a specificity of 98.3% demonstrate that the model has learned effectively.

Table 4.26 presents a summary analysis of the results in 20 epochs for the models trained in this section.

Table 4.26: Performance Metrics for top 3 Models

Models	Accuracy	Precision	Recall	F1-Score	Specificity
ResNet50	0.95416	0.96184	0.96305	0.96203	0.98034
EfficientNetB4	0.96078	0.97042	**0.96740**	0.96855	0.98295
EfficientNetB7	**0.96219**	**0.97383**	0.96690	**0.96991**	**0.98326**
Higher	EfficientNetB7	EfficientNetB7	EfficientNetB4	EfficientNetB7	EfficientNetB7
Second Higher	EfficientNetB4	EfficientNetB4	EfficientNetB7	EfficientNetB4	EfficientNetB4
Third Higher	ResNet50	ResNet50	ResNet50	ResNet50	ResNet50

One can see that the EfficientNetB7 demonstrates improved performance, through the lens of almost all metrics, than the other variants, indicating that training for more epochs is a good option.

4.2.3 50 epochs

Table 4.27 presents the results in 50 epochs for each class, for EfficientNetB7.

Table 4.27: Performance Metrics for EfficientNetB7

Class	Precision	Recall	F1-Score	Specificity
Normal	0.943	0.974	0.958	0.945
Lung Opacity	0.959	0.899	0.928	0.985
COVID	0.975	0.989	0.982	0.995
Viral Pneumonia	1.000	1.000	1.000	1.000
Mean	0.967	0.981	0.970	0.990
Accuracy	0.957			

Comparing the performance metrics between 20 epochs (Table 4.25) and 50 epochs (Table 4.27), it is observed that only the accuracy and recall are higher with 20 epochs.

4.2.4 Analysis of all training with COVID dataset

The following table indicates the best model for every training with the COVID dataset. The numbers after the model name specifies the number of training epochs.

Table 4.28: Performance Metrics - Part 1 (Accuracy to F1-Score)

Models	Accuracy	Precision	Recall	F1-Score
Best Model	EfficientNetB7 - 20	EfficientNetB7 - 20	EfficientNetB7 - 10	EfficientNetB7 - 20
Second Best Model	EfficientNetB4 - 20	EfficientNetB4 - 20	EfficientNetB4 - 20	EfficientNetB7 - 10
Third Best Model	EfficientNetB7 - 10	EfficientNetB7 - 50	EfficientNetB4 - 10	EfficientNetB4 - 20
Fourth Best Model	EfficientNetB7 - 50	EfficientNetB7 - 10	EfficientNetB7 - 20	EfficientNetB7 - 50
Fifth Best Model	EfficientNetB4 - 10	ResNet50 - 20	EfficientNetB7 - 50	ResNet50 - 20
Sixth Best Model	ResNet50 - 20	ResNet50 - 10	ResNet50 - 20	EfficientNetB4 - 10
Seventh Best Model	ResNet50 - 10	EfficientNetB4 - 10	ResNet101 - 10	ResNet50 - 10
Eighth Best Model	ResNet101 - 10	ResNet101 - 10	ResNet50 - 10	ResNet101 - 10
Ninth Best Model	EfficientNetB0 - 10	EfficientNetB0 - 10	EfficientNetB0 - 10	EfficientNetB0 - 10
Tenth Best Model	VGG16 - 10	VGG16 - 10	VGG16 - 10	VGG16 - 10

Table 4.29: Performance Metrics - Part 2 (Specificity to Sum of False Negatives)

Models	Specificity	False Negatives	Sum False Negatives
Best Model	EfficientNetB7 - 20	EfficientNetB7 - 20	EfficientNetB7 - 20
Second Best Model	EfficientNetB4 - 20	EfficientNetB4 - 20	EfficientNetB4 - 20
Third Best Model	EfficientNetB7 - 50	EfficientNetB7 - 50	EfficientNetB7 - 10
Fourth Best Model	ResNet50 - 20	ResNet50 - 20	EfficientNetB7 - 50
Fifth Best Model	EfficientNetB4 - 10	EfficientNetB4 - 10	EfficientNetB4 - 10
Sixth Best Model	EfficientNetB0 - 10	EfficientNetB0 - 10	ResNet50 - 20
Seventh Best Model	EfficientNetB7 - 10	EfficientNetB7 - 10	ResNet50 - 10
Eighth Best Model	ResNet50 - 10	ResNet50 - 10	ResNet101 - 10
Ninth Best Model	ResNet101 - 10	ResNet101 - 10	EfficientNetB0 - 10
Tenth Best Model	VGG16 - 10	VGG16 - 10	VGG16 - 10

Upon reviewing Tables 4.28 and Table 4.29 it is concluded that the optimal model is EfficientNetB7 trained over 20 epochs.

4.2.5 The Best model - COVID

Based on the overall ranking of the models, depicted in the aforementioned tables, it is possible to recognize as the best model for the COVID dataset the EfficientNetB7 when trained with 20 epochs, evaluated based on F1-Score. Table 4.30 shows again the results the model obtained in this scenario. Moreover, Fig. 4.2 presents the learning curve over the epochs.

Table 4.30: Performance Metrics for EfficientNetB7

Class	Precision	Recall	F1-Score	Specificity
Normal	0.944	0.984	0.964	0.946
Lung Opacity	0.975	0.902	0.937	0.991
COVID	0.984	0.989	0.986	0.997
Viral Pneumonia	0.993	0.993	0.993	0.999
Mean	0.974	0.967	0.970	0.983
Accuracy	0.962			

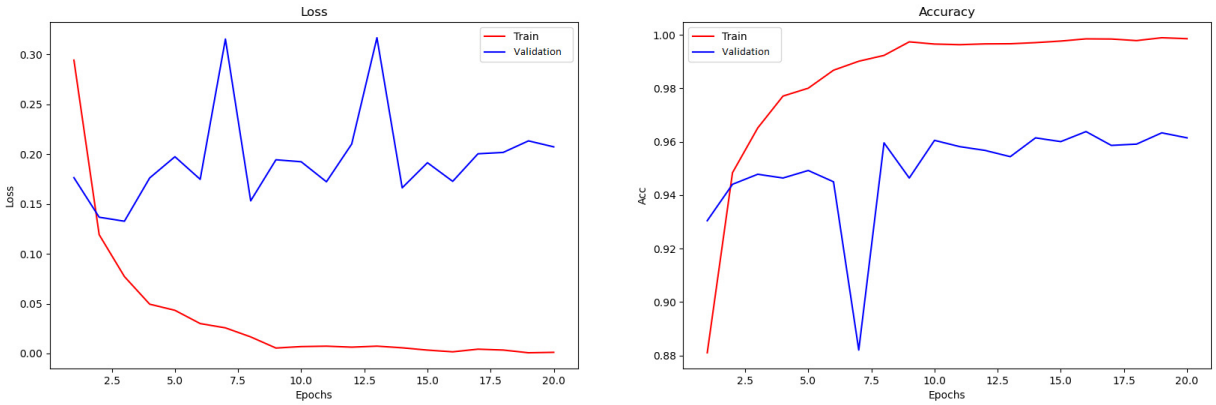


Figure 4.2: Curve of Accuracy and Loss from EfficientNetB7 trianed with 20 epochs. This model achieved an accuracy of 96.2%.

The EfficientNetB7 model demonstrates high performance across all evaluated metrics. Beginning with precision, which assesses the accuracy of positive predictions, notable high values are observed across all classes. Particularly noteworthy is the model's perfect precision for the "Viral Pneumonia" class, indicating that all positive predictions for this class are accurate. Furthermore, the model exhibits excellent recall scores, effectively capturing a substantial proportion of true positives for each class. Notably, the "COVID" class stands out with nearly perfect recall, underscoring the model's capability to identify almost all instances of COVID-19.

Turning to the F1-Score, the model consistently achieves high scores across all classes, indicative of balanced performance in terms of false positives and false negatives. This contributes significantly to its overall effectiveness in classification tasks. Moreover, the model demonstrates impressive accuracy, accurately classifying most instances across all classes.

Specificity, which measures the capacity to correctly identify true negatives, is exceptionally

high across all classes, highlighting the model's ability to minimize false positives. This is particularly evident in the "Viral Pneumonia" class, where the model achieves perfect specificity.

In summary, the EfficientNetB7 model exhibits robust performance across diverse metrics, underscoring its efficacy in accurately classifying chest X-ray images into various categories of pulmonary diseases, including COVID-19, Viral Pneumonia, Lung Opacity, and Normal cases.

4.3 DISCUSSION

4.3.1 The HAM10000

In the (TSCHANDL, 2018), the researchers fine-tuned an InceptionV3 architecture with weights pre-trained on ImageNet data, achieving an accuracy of 98.68%. This was accomplished using a learning rate of 0.003, a gamma value of 0.1, and a batch size of 64. In contrast, the model presented in this work achieved an accuracy of 84%, which is 14% lower than the top-performing model.

Table 4.31: InceptionV3 and EfficientB4 Architecture Details

Model	Total Layers	Total Params
InceptionV3	298	27,161,264
EfficientB4	548	19,341,616

When comparing the parameters of EfficientNetB4 and InceptionV3 (Tables 4.31 and 3.6), it is observed that, while InceptionV3 has a greater number of total parameters, EfficientNetB4 is more compact in terms of estimated total size and the number of layers.

Other results like (GAMAGE et al., 2024) achieved 98,37% accuracy using a Xception model.

4.3.2 The COVID dataset

In (ISLAM et al., 2023), a multi-class classification was performed by fine-tuning ResNet50, VGG19, and Xception models. The researchers achieved an accuracy of 75% for ResNet50, 92% for VGG19, and 93% for Xception. In comparison, the best COVID model trained in this work achieved an accuracy of 96.2%.

The 93% accuracy using Xception reported by (ISLAM et al., 2023) was attained using Google's Colab Pro edition hardware, which consists of 26.3 GB of system RAM and 16.1 GB of GPU memory, for this work it was used a 6 GB of GPU. They employed the same method to handle class imbalance by using weighted categorical loss and trained the models for 200 epochs.

This model has the potential to significantly improve the efficiency of doctors as they work through the intricate steps of diagnosis in pulmonary diseases. It helps doctors to dedicate their

time to other activities that are more urgent or perhaps require more technical knowledge. Moreover, the implementation of this model can lead to a substantial reduction in the costs associated with diagnosis. This includes not only the professional costs, such as the time but also the operational costs related to the use of medical equipment, laboratory tests, and other resources.

4.4 SUS-AI-COVID

The model trained with the COVID-19 dataset showed promising performance, justifying the investment in this diagnostic tool, aimed at detecting COVID-19, viral pneumonia and lung opacity. For implementation, it will be necessary to use the winning model to perform inferences on standard X-ray images. The cost of implementing and deploying the model is relatively low, and it can be easily integrated into the software used in the SUS via the internet (COOP, 2017). This means that the tool can be accessible anywhere in Brazil with internet access.

The X-ray images must be converted to .png format and resized to 299 x 299 pixels, maintaining the standard used during model training. Implementing the model in a cloud infrastructure can allow inferences to be performed efficiently and affordably (ENOH, 2024). The SUS software must be configured to receive these inferences and, instead of simply displaying the predicted class, it must present the probability or confidence level with which the model classifies each condition. To maximize the effectiveness of the tool, it would be advisable to create a specific course to train healthcare professionals in understanding and using the model. This will not only allow the physician to have the final say in the decision, but will also provide the possibility of complementing the diagnosis with other traditional tests. The main objective is to provide additional support to the physician, assisting in the clinical decision-making process.

Another fundamental aspect is the storage of the new data generated. This data, together with the final diagnoses, should be used for continuous re-training of the model, ensuring that it is constantly improving and adapting to new information (NUBANK, 2022). In addition, it is essential to consider the privacy and security issues of the data, ensuring that it is stored only with the patient's consent.

5 CONCLUSION

This work aimed to present various models capable of classifying lung diseases and skin lesions. Models such as these have the potential to reduce queues in hospitals and assist in diagnostic decision-making. The first analysis was conducted using the HAM10000 dataset, which comprises images of skin lesions. To evaluate the optimal model, the performance of six Convolutional Neural Network (CNN) models was assessed over 10 epochs. Following this initial evaluation, three of the six models were selected for further training over 20 epochs. Ultimately, the best-performing model was subjected to 50 epochs of training. For the HAM10000 dataset, the most effective model was EfficientNetB4, trained with 20 epochs.

The second analysis utilized the COVID dataset, encompassing three lung diseases: COVID-19, Viral Pneumonia, and Pulmonary Opacity. This dataset consists of chest X-rays. The same methodology was applied, with the six models initially trained over 10 epochs. Subsequently, three models were selected for 20 epochs of training, culminating in the best model being trained over 50 epochs. For the COVID dataset, the most effective model was EfficientNetB7, trained with 20 epochs.

In the realm of computer vision, optimal model performance necessitates a substantial amount of data to prevent overfitting issues. Hence, high-quality data is imperative for achieving satisfactory results. As demonstrated in this work, utilizing the largest model does not necessarily guarantee superior outcomes.

5.1 FUTURE WORK

In this work, we proposed an analysis to classify different types of skin lesions and pulmonary diseases using images as input. Through this work, we identified several areas for improvement.

One area for further exploration is hyperparameters. Hyperparameter tuning, using methods such as grid search or random search, could help identify the optimal set of hyperparameters for the model. Additionally, we could explore different types of optimizers (KANDEL; CASTELLI; POPOVIČ, 2020), including:

- Adam
- RMSprop
- Batch Gradient Descent
- Stochastic Gradient Descent
- Mini-Batch Gradient Descent

- Nesterov Momentum Gradient Descent

Another potential improvement may lie in the loss function. The Focal Loss function, for instance, is particularly suited for imbalanced classification.

Data augmentation is another technique worth exploring. This technique increases the size of the dataset by applying transformations to the original images, such as rotation, translation, scaling, flipping, and zooming. While data augmentation does not always improve the model, other approaches, such as adding white noise or random-erasing parts of an image, have been shown to enhance CNN models (MIKOŁAJCZYK; GROCHOWSKI, 2018).

Reinforcement Learning (RL) is another promising technique. RL involves an agent learning to achieve a goal by interacting with the environment. In this context, the goal would be to classify the image correctly. One possibility is to use the Actor-Critic method, where the Actor is a neural network and the Critic is another neural network that outputs the value function for the Actor. RL could make the model more robust and adaptive to different data distributions (SUTTON; BARTO, 2018).

A APPENDIX

For all the following images, consider that the blue curve represents the ”Validation“ loss, not the ”Evaluation“ as said written in the image.

A.1 HAM10000 - TRAINING CURVE

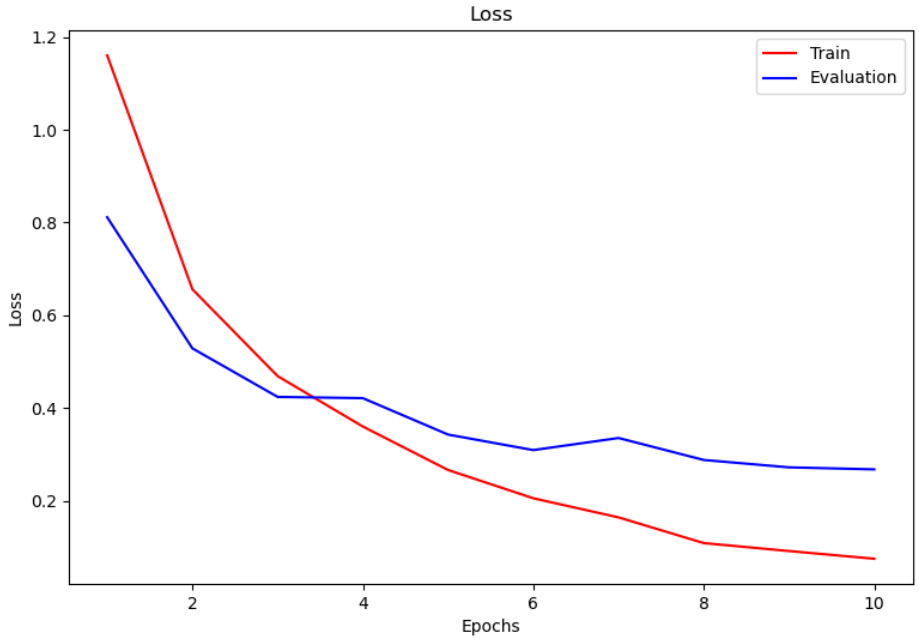


Figure A.1: Loss function values over 10 epochs, during training and validation of model EfficientNetB0.

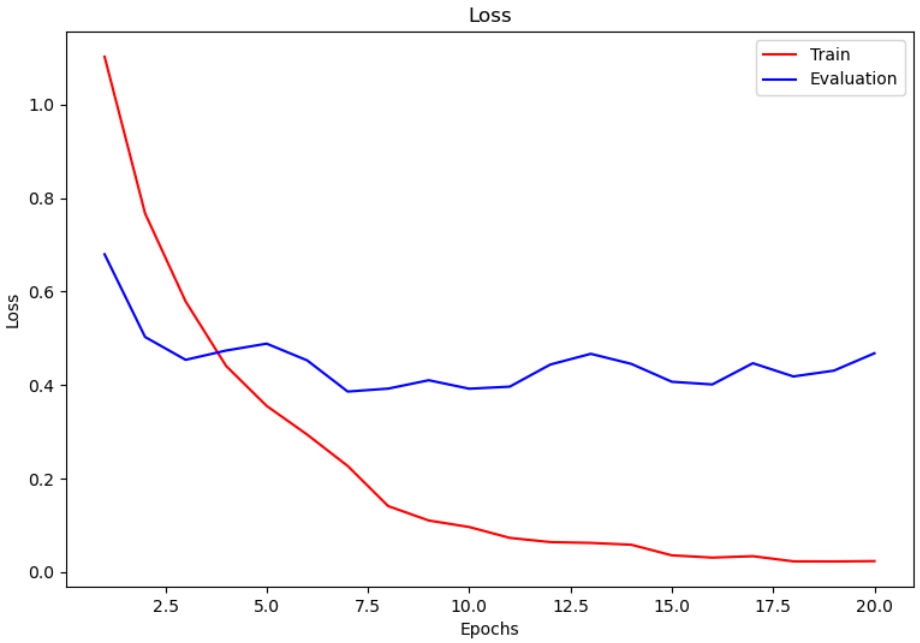


Figure A.2: Loss function values over 20 epochs, during training and validation of model EfficientNetB0.

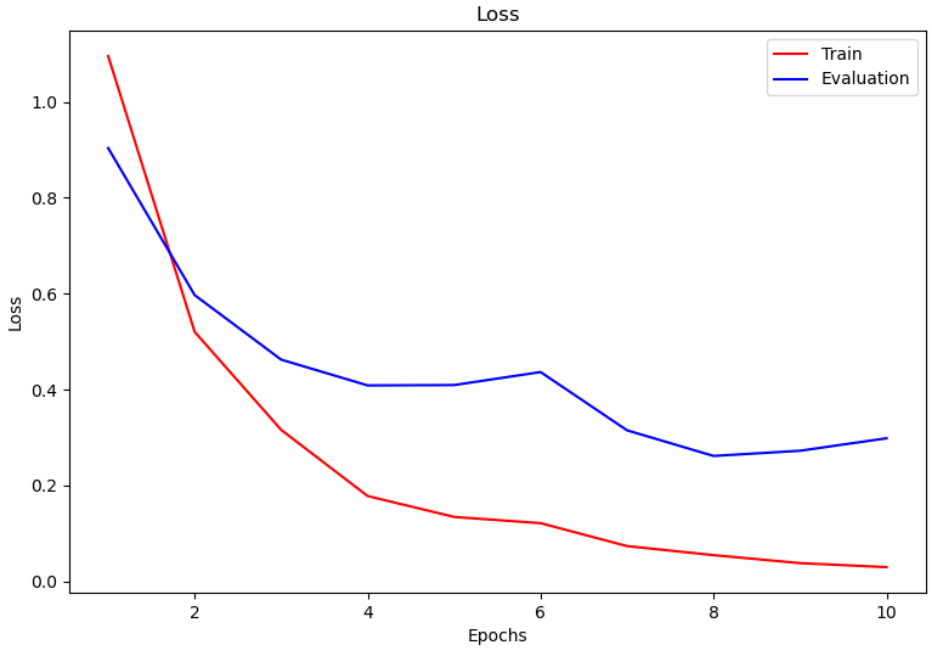


Figure A.3: Loss function values over 10 epochs, during training and validation of model EfficientNetB4.

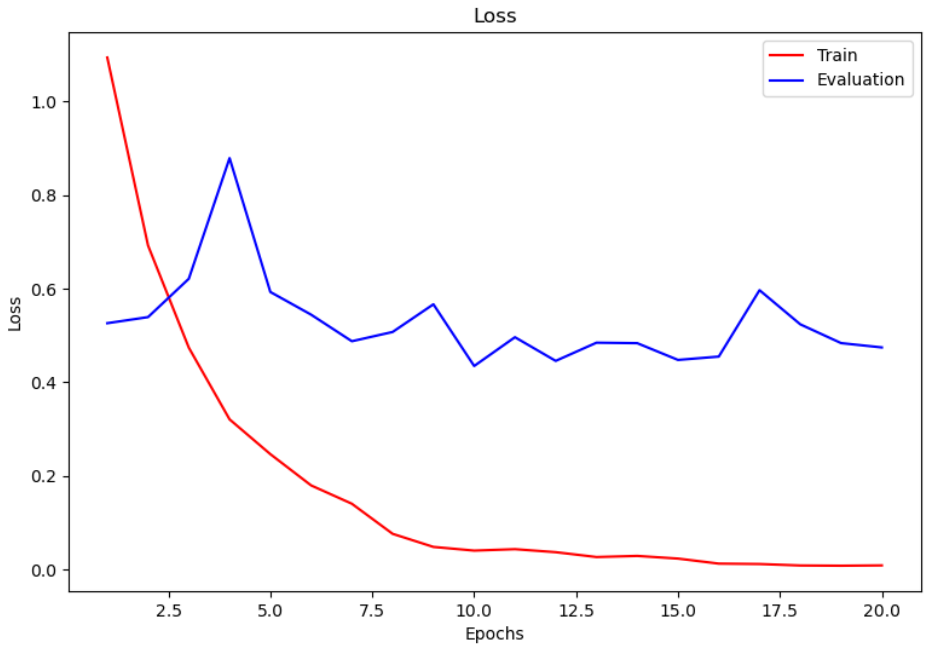


Figure A.4: Loss function values over 20 epochs, during training and validation of model EfficientNetB4.

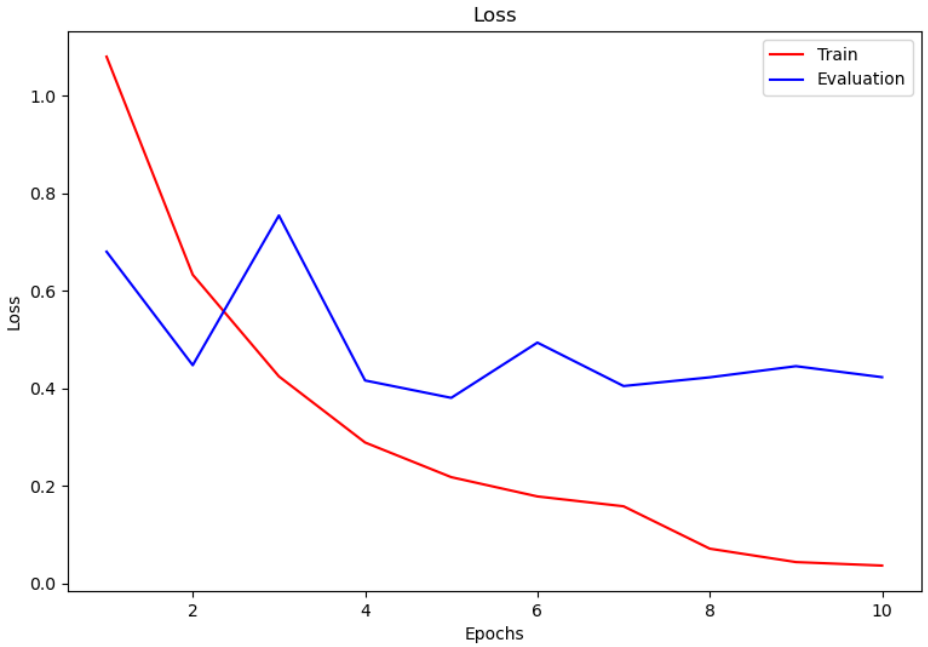


Figure A.5: Loss function values over 10 epochs, during training and validation of model EfficientNetB7.

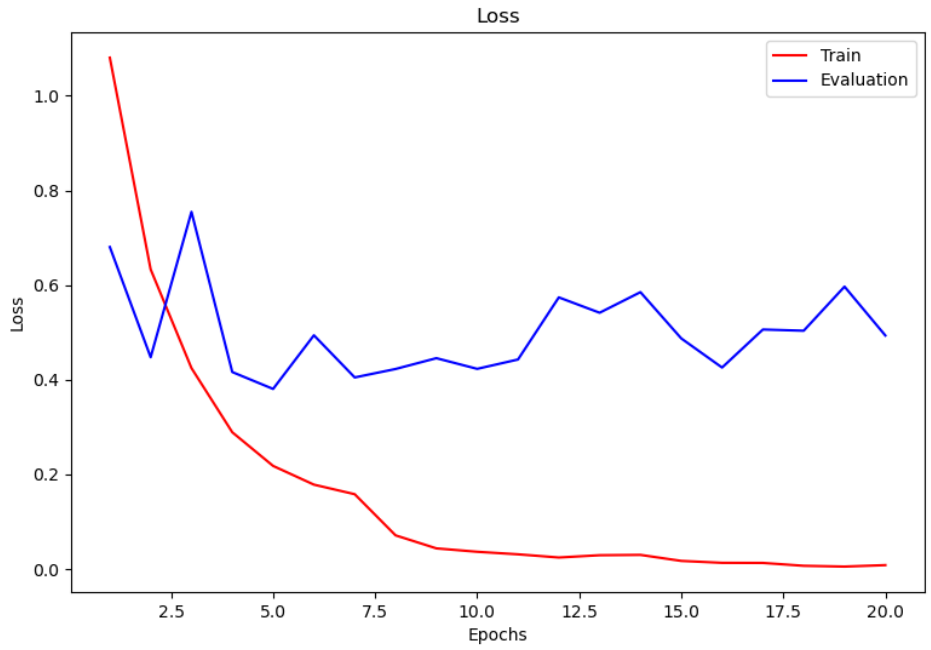


Figure A.6: Loss function values over 20 epochs, during training and validation of model EfficientNetB7.

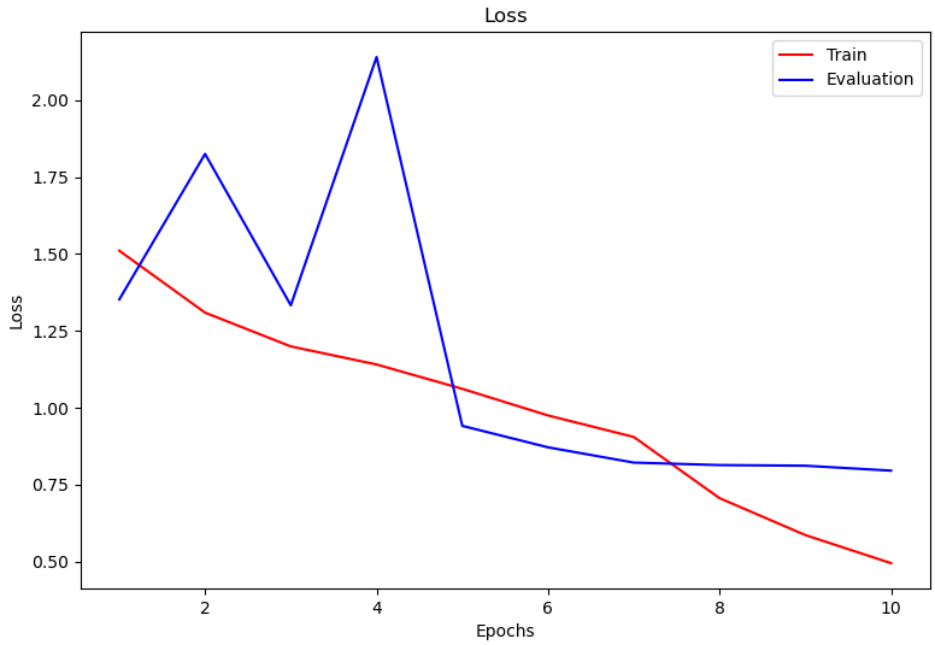


Figure A.7: Loss function values over 10 epochs, during training and validation of model ResNet101.

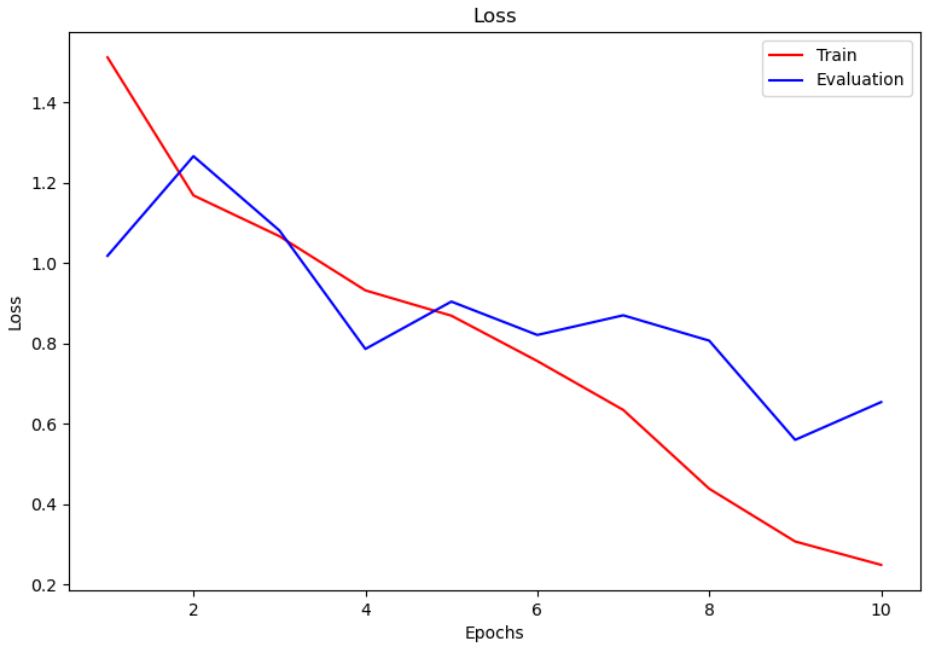


Figure A.8: Loss function values over 10 epochs, during training and validation of model ResNet50.

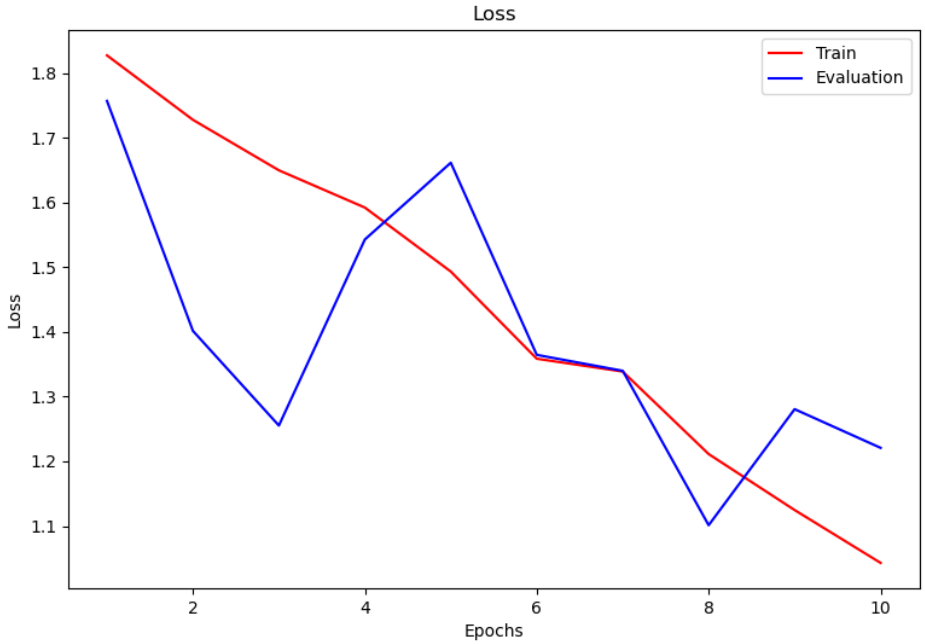


Figure A.9: Loss function values over 10 epochs, during training and validation of model VGG16.

A.2 COVID - TRAINING CURVE

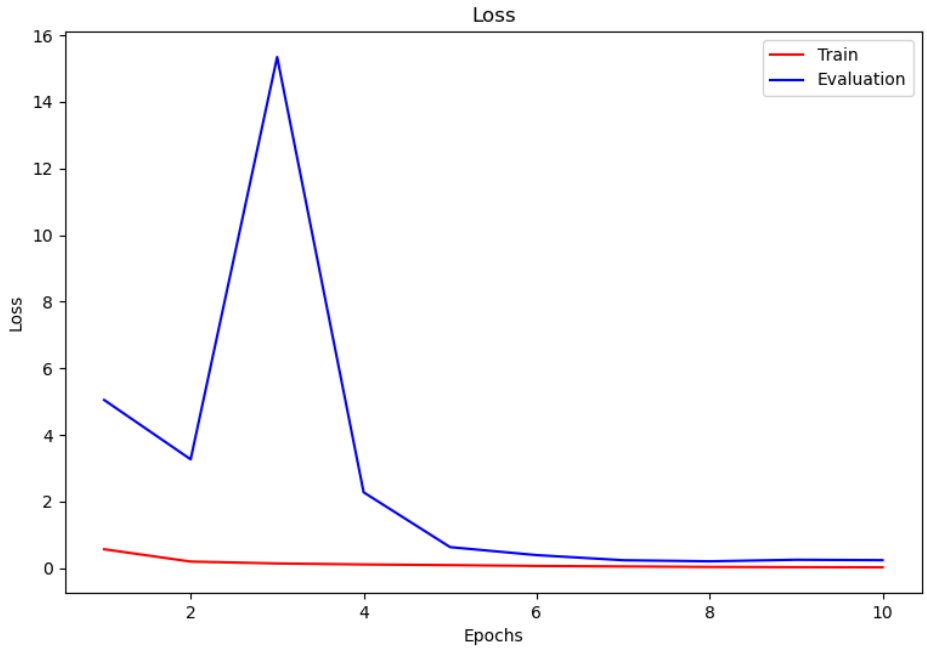


Figure A.10: Loss function values over 10 epochs, during training and validation of model EfficientNetB0.

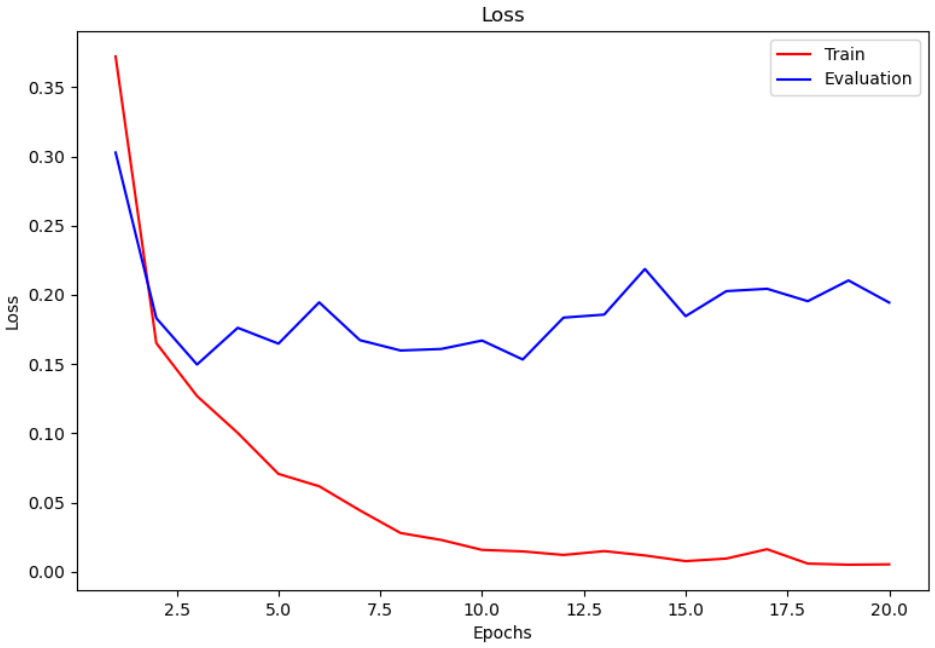


Figure A.11: Loss function values over 20 epochs, during training and validation of model EfficientNetB0.

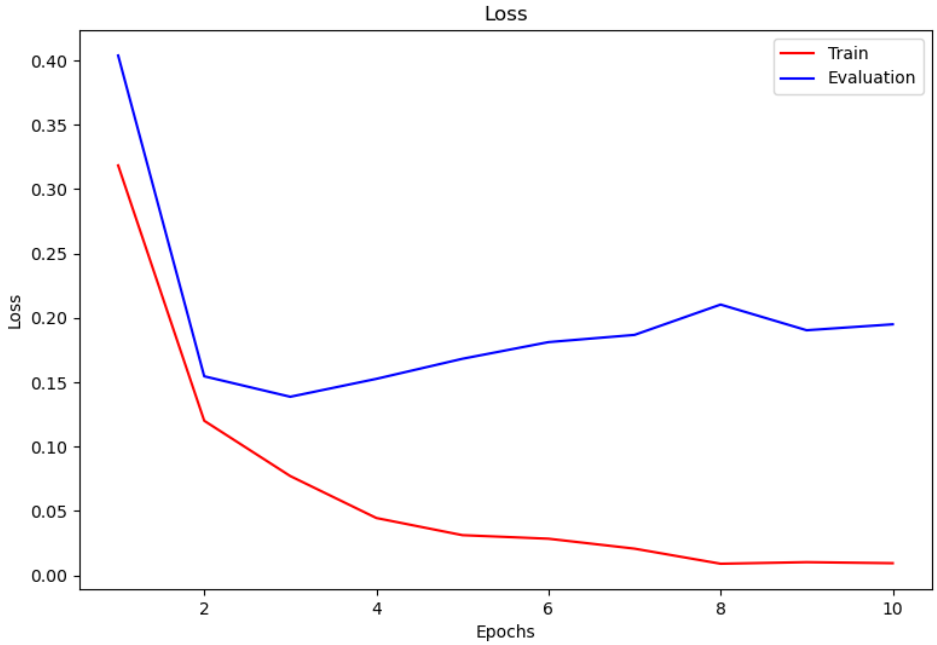


Figure A.12: Loss function values over 10 epochs, during training and validation of model EfficientNetB4.

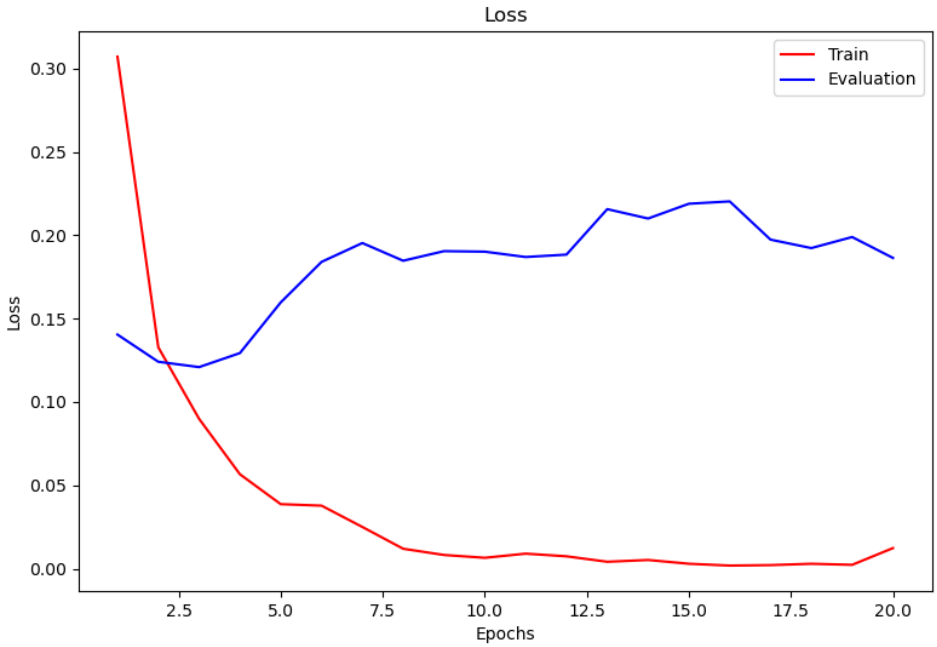


Figure A.13: Loss function values over 20 epochs, during training and validation of model EfficientNetB4.

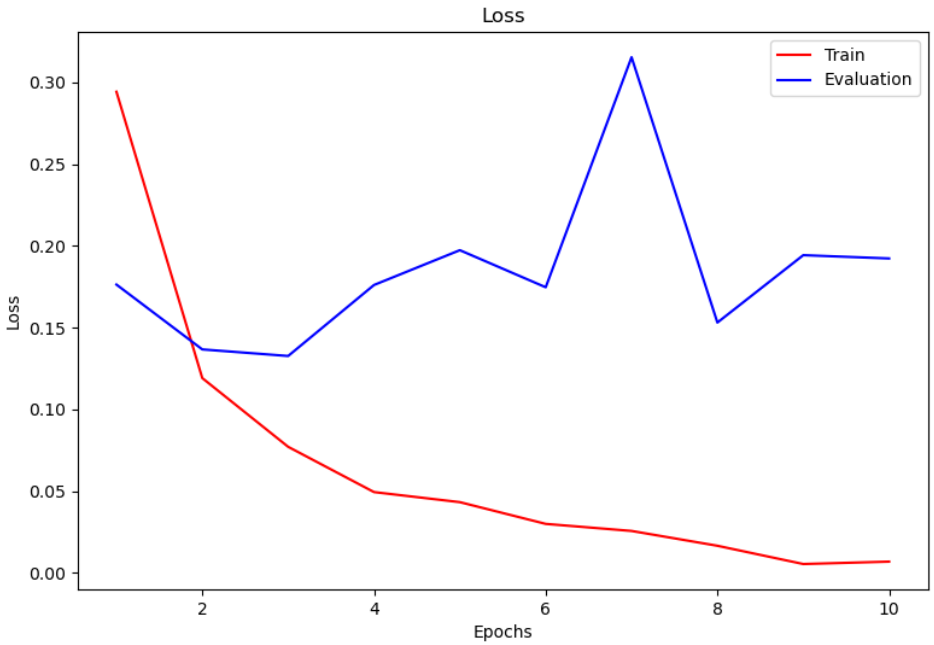


Figure A.14: Loss function values over 10 epochs, during training and validation of model EfficientNetB7.

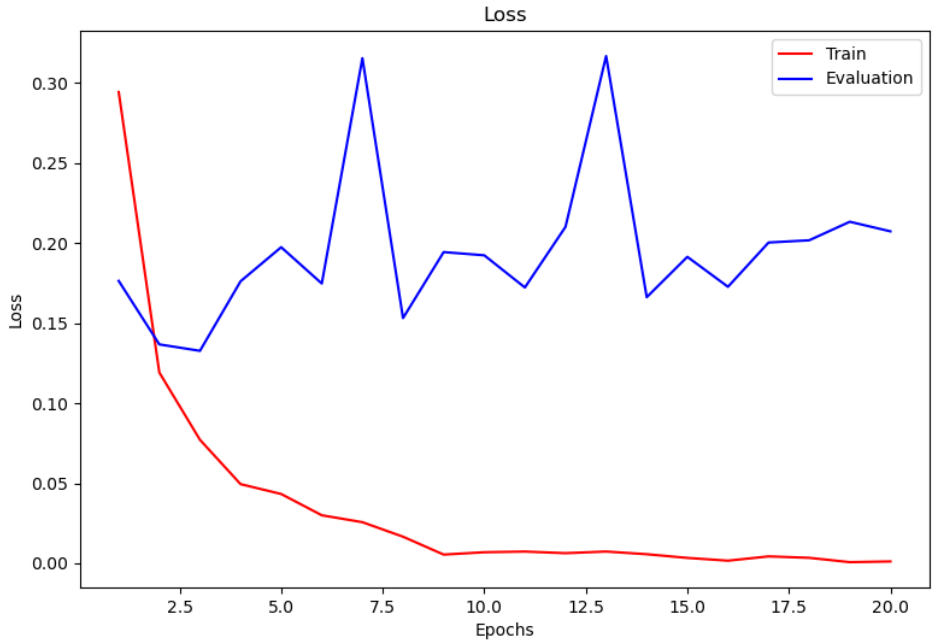


Figure A.15: Loss function values over 20 epochs, during training and validation of model EfficientNetB7.

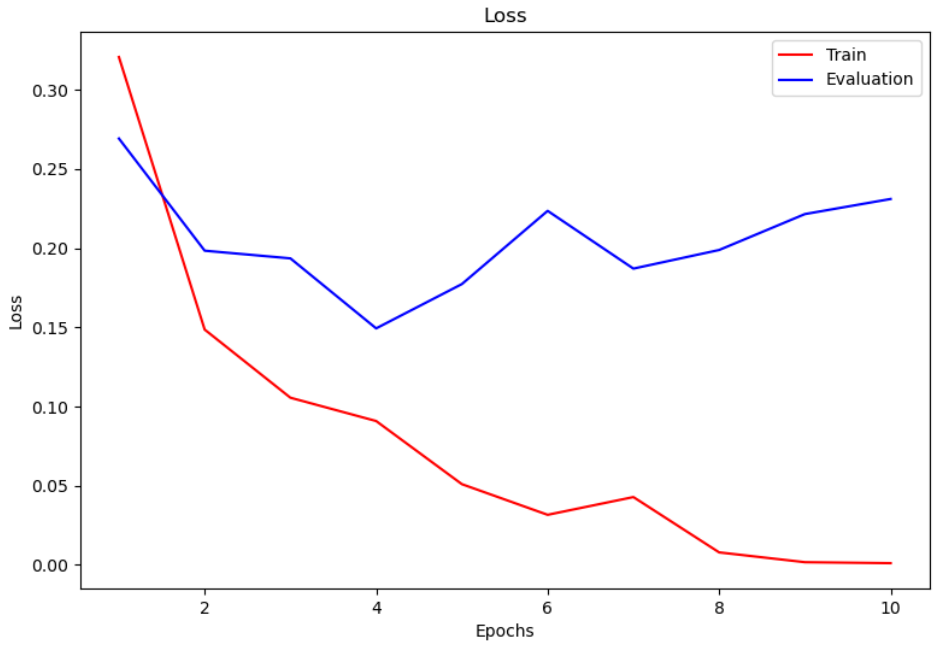


Figure A.16: Loss function values over 10 epochs, during training and validation of model ResNet101.

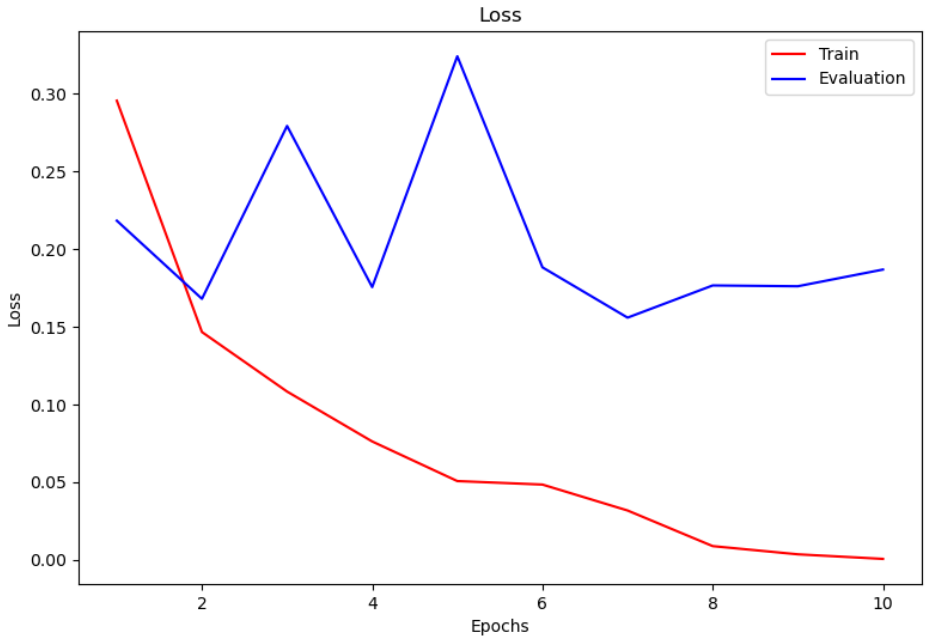


Figure A.17: Loss function values over 10 epochs, during training and validation of model ResNet50.

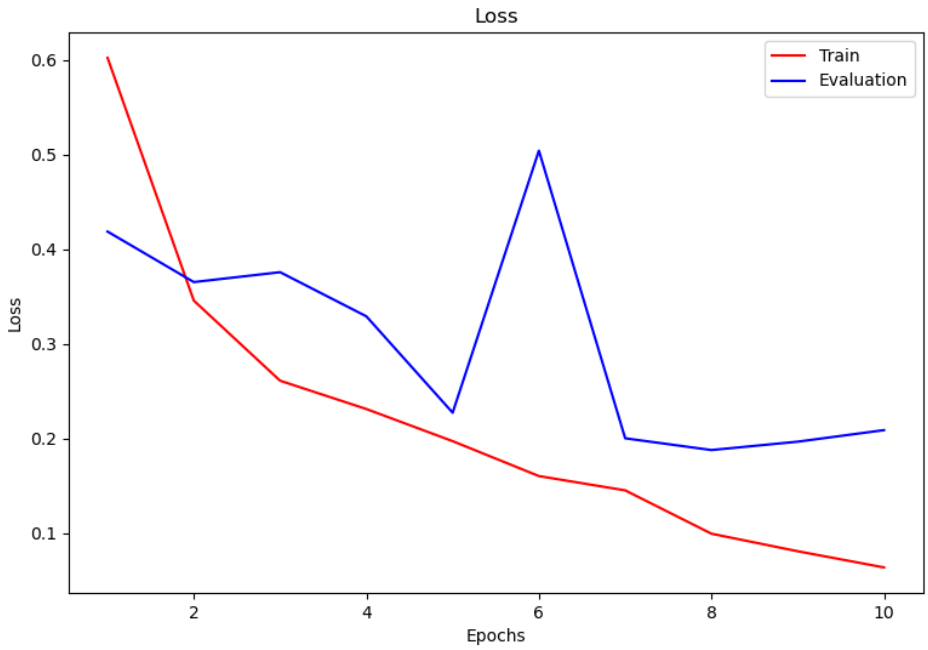


Figure A.18: Loss function values over 10 epochs, during training and validation of model VGG16.

REFERENCES

ALBAWI, S.; MOHAMMED, T. A.; ALZAWI, S. Understanding of a convolutional neural network. In: . [S.l.: s.n.], 2017.

ALMEIDA, J. V. D. R. et al. Mortalidade por pneumonia: Uma jornada de altos e baixos no brasil (2018-2022). *Revista Ibero-Americana de Humanidades, Ciências e Educação*, Revista Ibero-Americana de Humanidades, Ciencias e Educacao, v. 10, n. 4, p. 1161–1170, abr. 2024. ISSN 2675-3375. Disponível em: <http://dx.doi.org/10.51891/rease.v10i4.13514>.

ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2020.

AMANATULLAH. *Vanishing Gradient Problem in Deep Learning: Understanding, Intuition, and Solutions — amanatulla1606*. 2023. <https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54>. [Accessed 09-08-2024].

ARENAS-JIMÉNEZ, J. J.; PLASENCIA-MARTÍNEZ, J. M.; GARCÍA-GARRIGÓS, E. Cuando la neumonía no es COVID-19. *Radiol. (Engl. Ed.)*, Elsevier BV, v. 63, n. 2, p. 180–192, mar. 2021.

BERNHEIM, A. et al. Chest ct findings in coronavirus disease-19 (covid-19): relationship to duration of infection. *Radiology*, Radiological Society of North America, v. 295, n. 3, p. 685–691, 2020.

BLOCK, H.-D. The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, APS, v. 34, n. 1, p. 123, 1962.

CHAN, H.-P.; HADJIISKI, L. M.; SAMALA, R. K. Computer-aided diagnosis in the era of deep learning. *Med. Phys.*, Wiley, v. 47, n. 5, p. e218–e227, jun. 2020.

CHOLLET, F. *Deep learning with python*. New York, NY: Manning Publications, 2022.

COOP, R. *What is the Cost to Deploy and Maintain a Machine Learning Model? — phdata.io*. 2017. <https://www.phdata.io/blog/what-is-the-cost-to-deploy-and-maintain-a-machine-learning-model/>. [Accessed 13-08-2024].

DELASHMIT, W. H.; MISSILES, L. M. Recent developments in multilayer perceptron neural networks. In: . [s.n.], 2005. Disponível em: <https://api.semanticscholar.org/CorpusID:14200794>.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009. Disponível em: <http://dx.doi.org/10.1109/CVPR.2009.5206848>.

DUANMU, H. et al. Deep learning of longitudinal chest x-ray and clinical variables predicts duration on ventilator and mortality in COVID-19 patients. *Biomed. Eng. Online*, v. 21, n. 1, p. 77, out. 2022.

ENOH, J. *Building AI Models for Medical Imaging Analysis: A Multi-Cloud Deployment Guide — linkedin.com*. 2024. <https://www.linkedin.com/pulse/building-ai-models-medical-imaging-analysis-multi-cloud-john-enoh-gjsqc/>. [Accessed 20-08-2024].

FALQUETO, L. E.; PAES, R. L.; PASSARO, A. KNN e rede neural convolucional para o reconhecimento de plataformas de petróleo em imagens SAR do sentinel-1. *Spectrum*, Instituto de Aplicacoes Operacionais, v. 24, n. 1, p. 29–33, set. 2023.

FUJITA, H. AI-based computer-aided diagnosis (AI-CAD): the latest review to read first. *Radiol. Phys. Technol.*, Springer Science and Business Media LLC, v. 13, n. 1, p. 6–19, mar. 2020.

GAMAGE, L. et al. Melanoma skin cancer identification with explainability utilizing mask guided technique. *Electronics (Basel)*, MDPI AG, v. 13, n. 4, p. 680, fev. 2024.

GARDNER, M. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, v. 32, n. 14, p. 2627–2636, 1998. ISSN 1352-2310.

GERON, A. *Hands-on machine learning with scikit-learn, keras, and TensorFlow*. 2. ed. Sebastopol, CA: O'Reilly Media, 2019.

GOEL, T. et al. OptCoNet: an optimized convolutional neural network for an automatic diagnosis of COVID-19. *Appl. Intell.*, Springer Science and Business Media LLC, v. 51, n. 3, p. 1351–1366, 2021.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

HAYGOOD, T. M.; BRIGGS, J. E. World War II Military Led the Way in Screening Chest Radiography. *Military Medicine*, v. 157, n. 3, p. 113–116, 03 1992. ISSN 0026-4075. Disponível em: <https://doi.org/10.1093/milmed/157.3.113>.

HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.

HERRING, W. *Learning radiology: recognizing the basics*. [S.l.]: Elsevier Health Sciences, 2019.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, v. 9, p. 1735–80, 12 1997.

HOPKINS, J. *Pneumonia*. 2024. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia#:~:text=Pneumonia%20is%20an%20infection%20of,(lobes)%20of%20the%20lungs>. [Accessed 09-08-2024].

HOPKINS, J. *Pneumonia*. 2024. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia#:~:text=Pneumonia%20is%20an%20infection%20of,(lobes)%20of%20the%20lungs.> [Accessed 08-08-2024].

ISLAM, M. N. et al. Interpretable differential diagnosis of non-COVID viral pneumonia, lung opacity and COVID-19 using tuned transfer learning and explainable AI. *Healthcare (Basel)*, v. 11, n. 3, jan. 2023.

KANDEL, I.; CASTELLI, M.; POPOVIČ, A. Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images. *J. Imaging*, MDPI AG, v. 6, n. 9, p. 92, set. 2020.

KIM, M.-J. et al. Functionality-based processing-in-memory accelerator for deep convolutional neural networks. *IEEE Access*, PP, p. 1–1, 10 2021.

KIM, W. et al. Utility of a deep learning algorithm for detection of reticular opacity on chest radiography in patients with interstitial lung disease. *AJR Am. J. Roentgenol.*, American Roentgen Ray Society, v. 218, n. 4, p. 642–650, abr. 2022.

KLEYKO, D. et al. Perceptron theory can predict the accuracy of neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, PP, fev. 2023.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

LEUNG, H.; HAYKIN, S. The complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, v. 39, n. 9, p. 2101–2104, 1991.

MARTINEZ-MURCIA, F. J. et al. Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy. *Neurocomputing*, Elsevier BV, v. 452, p. 424–434, set. 2021.

MIKOŁAJCZYK, A.; GROCHOWSKI, M. Data augmentation for improving deep learning in image classification problem. In: *2018 International Interdisciplinary PhD Workshop (IIPhDW)*. [S.l.: s.n.], 2018. p. 117–122.

MUKHERJEE, S. *The Annotated ResNet-50 — towardsdatascience.com*. 2022. <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>. [Accessed 09-08-2024].

NASCIMENTO, F. A. d. O. et al. Computational intelligence conceptions to automated diagnosis: Feature grouping for performance improvement. *Braz. Arch. Biol. Technol.*, FapUNIFESP (SciELO), v. 66, 2023.

NUBANK. *Retreinamento automático para modelos de aprendizado de máquina: dicas e lições aprendidas*. 2022. <https://building.nubank.com.br/pt-br/retreinamento-automatico-para-modelos-de-aprendizado-de-maquina-dicas-e-licoes-aprendidas/>. [Accessed 13-08-2024].

PASCARELLA, G. et al. COVID-19 diagnosis and management: a comprehensive review. *J. Intern. Med.*, Wiley, v. 288, n. 2, p. 192–206, ago. 2020.

RAGHU, M. et al. Transfusion: Understanding transfer learning for medical imaging. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. v. 32. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2019/file/eb1e78328c46506b46a4ac4a1e378b91-Paper.pdf>.

REHMAN, Y. *How the bottleneck layers in the deep networks work and how do those layers reduce computational... — reh.yawar2*. 2024. <https://medium.com/@reh.yawar2/how-the-bottleneck-layers-in-the-deep-networks-work-and-how-do-those-layers-reduce-computational-7bc99c0d1e96> [Accessed 09-08-2024].

REZAEI-DASTJERDEHEI, M. R.; MIJANI, A.; FATEMIZADEH, E. Addressing imbalance in multi-label classification using weighted cross entropy loss function. In: *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*. [S.l.: s.n.], 2020. p. 333–338.

ROSENBLATT, F. *The perceptron, a perceiving and recognizing automaton Project Para*. [S.l.]: Cornell Aeronautical Laboratory, 1957.

SAUDE, M. da. *Câncer de pele — gov.br*. 2020. <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/c/cancer-de-pele#:~:text=Entre%20os%20tumores%20de%20pele,se%20n%C3%A3o%20for%20tratado%20adequadamente.&text=N%C3%BAmero%20de%20mortes%20no%20Brasil%3A%202.616%2C%20sendo%201.488%20homens%20e,Mortalidade%20por%20C%C3%A2ncer%20%2D%20SIM>. [Accessed 20-08-2024].

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. arXiv, 2014.

SRINIVASU, P. N. et al. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors (Basel)*, MDPI AG, v. 21, n. 8, p. 2852, abr. 2021.

SUN, L.; LIANG, F.; CUI, W. Artificial neural network and its application research progress in chemical process. *AJRCoS*, Sciencedomain International, p. 177–185, dez. 2021.

SUN, M.; HUANG, Z.; GUO, C. Automatic diagnosis of alzheimer's disease and mild cognitive impairment based on CNN+SVM networks with end-to-end training. In: *2021 13th International Conference on Advanced Computational Intelligence (ICACI)*. [S.l.]: IEEE, 2021.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning*. 2. ed. Cambridge, MA: Bradford Books, 2018. (Adaptive Computation and Machine Learning series).

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International conference on machine learning*. [S.l.], 2019. p. 6105–6114.

TAN, M.; LE, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv, 2019.

TING, K. M. Inducing cost-sensitive trees via instance weighting. In: *Principles of Data Mining and Knowledge Discovery*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, (Lecture notes in computer science). p. 139–147.

TSCHANDL, P. *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. Harvard Dataverse, 2018. Disponível em: <https://doi.org/10.7910/DVN/DBW86T>.

TÜRK, F.; KÖKVER, Y. Detection of lung opacity and treatment planning with three-channel fusion CNN model. *Arab. J. Sci. Eng.*, p. 1–13, abr. 2023.

WU, Y.-C.; FENG, J.-W. Development and application of artificial neural network. *Wirel. Pers. Commun.*, Springer Nature, dez. 2017.

YAQUB, M. et al. State-of-the-art cnn optimizer for brain tumor segmentation in magnetic resonance images. *Brain Sciences*, MDPI AG, v. 10, n. 7, p. 427, jul. 2020. ISSN 2076-3425.

YINGGE, H.; ALI, I.; LEE, K.-Y. Deep neural networks on chip - a survey. In: . [S.l.: s.n.], 2020. p. 589–592.

ZHOU; CHELLAPPA. Computation of optical flow using a neural network. In: IEEE. *IEEE 1988 international conference on neural networks*. [S.l.], 1988. p. 71–78.

ZHOU, V. *Neural Networks*. 2020. <https://victorzhou.com/series/neural-networks-from-scratch/>. [Accessed 08-08-2024].

ZHU, M. et al. A computerized tomography-based radiomic model for assessing the invasiveness of lung adenocarcinoma manifesting as ground-glass opacity nodules. *Respir. Res.*, Springer Science and Business Media LLC, v. 23, n. 1, p. 96, 2022.

ZHUANG, F. et al. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, v. 109, n. 1, p. 43–76, 2021.