



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Pablo David Minango Negrete

# **Uso de Técnicas de Deep Learning para Classificação de Imagens Dermoscópicas com Lesões na Pele**

Campinas

2020



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Pablo David Minango Negrete

**Uso de Técnicas de Deep Learning para Classificação de Imagens Dermoscópicas  
com Lesões na Pele**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. Yuzo Iano

Este exemplar corresponde à versão final da tese defendida pelo aluno Pablo David Minango Negrete, e orientada pelo Prof. Dr. Yuzo Iano

---

Campinas

2020

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

M66u Minango Negrete, Pablo David, 1992-  
Uso de técnicas de deep learning para classificação de imagens  
dermoscópicas com lesões na pele / Pablo David Minango Negrete. –  
Campinas, SP : [s.n.], 2020.

Orientador: Yuzo Iano.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade  
de Engenharia Elétrica e de Computação.

1. Neoplasias cutâneas. 2. Melanoma. 3. Aprendizado profundo. 4. Redes  
neurais convolucionais. I. Iano, Yuzo, 1950-. II. Universidade Estadual de  
Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Using deep learning techniques to classify dermoscopy images  
with skin lesions

**Palavras-chave em inglês:**

skin cancer

Melanoma

Deep learning

Convolutional neural network

**Área de concentração:** Telecomunicações e Telemática

**Titulação:** Mestre em Engenharia Elétrica

**Banca examinadora:**

Yuzo Iano [Orientador]

Rosivaldo Ferrarezi

Vicente Idalberto Becerra Sablón

**Data de defesa:** 27-07-2020

**Programa de Pós-Graduação:** Engenharia Elétrica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0002-3382-7616>

- Currículo Lattes do autor: <http://lattes.cnpq.br/5356259746044576>

## COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

**Candidato:** Pablo David Minango Negrete RA: 226760

**Data da Defesa:** 27 de julho de 2020

**Título da Tese:** “Uso de Técnicas de Deep Learning para Classificação de Imagens Dermoscópicas com Lesões na Pele”.

Prof. Dr. Yuzo Iano (Presidente)

Prof. Dr. Rosivaldo Ferrarezi

Prof. Dr. Vicente Idalberto Becerra Sablón

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

# Agradecimentos

Em primeiro lugar agradeço a Deus pelas bênçãos prestadas durante a minha estadia aqui no Brasil e sob todo ao longo da minha vida. Agradeço aos meus pais Marcelo e Silvia pelo seu amor, sacrificio e por serem meus exemplos à seguir para ser uma pessoa melhor. Ao meu irmão Juan Carlos pela sua ajuda e conselhos ao longo deste processo além da sua amizade. Também gostaria agradecer a minha namorada Karina pelo seu carinho, compressão e motivação apesar da distância.

Além do mais, desejo expressar meus sinceros agradecimentos ao meu orientador Prof. Dr. Yuzo Iano pela grande oportunidade de trabalhar no "LCV" Laboratorio de Comunicações Visuais, pelos seus conselhos, sugestões e brindarme apoio nos estudos.

Agradeço aos colegas e amigos do LCV pelo acolhimento, pelos conselhos e o grande ambiente de trabalho. Aos meus amigos da UNICAMP os quais fizeram a minha estadia mais agradáveis.

Agradeço a Universidade Estadual de Campinas (UNICAMP), a Faculdade de Engenharia Elétrica e de Computação (FEEC), ao Departamento de Comunicações (DECOM), pelo apoio no desenvolvimento da dissertação do meu mestrado.

Finalmente, agradeço à SENESCYT - IFTH pelo suporte financeiro.

Um agradecimento especial ao Dr. Santiago Palacios diretor do Centro de la Piel "CEPI" por compartilhar as imagens de lesões de pele.

*A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê. (Arthur Schopenhauer).*

# Resumo

O melanoma é o tipo mais agressivo de câncer de pele e o mais perigoso de se espalhar pelo corpo todo, caso não seja diagnosticado precocemente. No Brasil segundo o Instituto Nacional do Câncer há uma estimativa de 2920 novos casos em homens, e de 3340 novos casos em mulheres a cada ano, sendo a maioria destes, encontra-se na região sul do país. Um outro tipo de neoplasia cutânea é o não-melanoma, possui uma estimativa de 85170 novos casos para homens e 80140 para mulheres por ano, mostrando um verdadeiro problema no sistema de saúde pública. O índice alto de cura das neoplasias cutâneas, durante os seus primeiros estágios, faz do diagnóstico, com o auxílio de computador, uma ferramenta importante para o médico avaliar a lesão de pele, e tomar uma decisão rápida para o tratamento para o paciente.

A análise de pele, entre outras áreas de imagens médicas vem sofrendo uma revolução graças ao aprendizado profundo (*deep learning*) e às redes neurais convolucionais CNN ("*Convolutional Neural Networks*"). Nesta dissertação, são investigadas arquiteturas de aprendizagem profundas, para classificação de imagens do tipo melanoma versus não-melanoma.

Neste contexto são avaliadas três arquiteturas que são: AlexNet, VGG16 e VGG19, na qual estas duas ultimas fazem uso da aprendizagem por transferência (*transfer-learning*). Apesar do limitado conjunto de dados públicos é devido a que coletar e anotar imagens com lesões de pele são tarefas difíceis, com custo elevado e só possíveis a serem feitas por profissionais da área da saúde, treinamos e avaliamos estas arquiteturas com três diferentes modelos, os quais contêm diversas quantidades de imagens aumentadas de forma progressiva.

Como resultado da pesquisa, a arquitetura "AlexNet" atinge um desempenho satisfatório na tarefa de classificação binária, obtendo um **AUC** (área sob a curva) de 0,935 em comparação com as arquiteturas "VGG16" e "VGG19".

**Palavras-chaves:** Neoplasias cutâneas; melanoma; não-melanoma; *deep learning*; redes neurais convolucionais; AlexNet; VGG16; VGG19.

# Abstract

The melanoma is the more aggressive skin cancer and the more dangerous to spread around the corp if it is not early diagnosed. In Brazil by the Instituto Nacional do Câncer there are an estimated of 2920 new cases in men and 3340 new cases in women each year. The majority of cases are found in the South Region. Another type of skin cancer is non-melanoma and it has an estimated of 85170 new cases in men and 80140 in women for the year showing a serious problem in the public health system. The high probability of healing skin cancer in its initial stages, made of the diagnosed with the help computer an important tool for the medic to evaluate the skin lesion and take a quick decision to the treatment for the patient.

The skin analysis and other medical imaging areas are revolutionizing due to deep learning and convolutional neural networks (CNN). In this dissertation, we investigated deep learning architectures to images classification of melanoma versus non-melanoma.

In this context we evaluated three architectures which are the "AlexNet", and with the use of transfer-learning "VGG16" and "VGG19" architectures.

Although the public dataset is limited due to collect and annotated the skin lesion images is a difficult task, with a high cost and only made by medical professionals, we have trained and evaluated these architectures with three different models that have diverse quantities of images progressively increased.

The "AlexNet" architecture achieves optimal performance in the binary classification task, obtaining 0,935 of **AUC** (area under the curve) in comparison with "VGG16" and "VGG19".

**Keywords:** skin cancer; melanoma; non-melanoma; deep learning; convolutional neural networks; AlexNet; VGG16; VGG19.

# Lista de ilustrações

Figura 2.1 – Inteligência artificial, <i>machine learning</i> e <i>deep learning</i> . . . . .	20
Figura 2.2 – Generalização. . . . .	22
Figura 2.3 – (a) <i>Overfitting</i> ou sobreajuste, e (b) <i>Underfitting</i> ou subajuste. . . . .	23
Figura 2.4 – Neurônio. . . . .	24
Figura 2.5 – <i>Perceptron</i> . . . . .	25
Figura 2.6 – MLP ( <i>multi-layer perceptron</i> ) com três camadas intermediárias ou es- condidas. . . . .	26
Figura 2.7 – Calculando os valores de saída de um exemplo de convolução. . . . .	27
Figura 2.8 – Calculando os valores de <i>max pooling</i> . . . . .	28
Figura 2.9 – Exemplo de uma arquitetura típica CNN. . . . .	29
Figura 2.10 – Camadas da pele humana. . . . .	30
Figura 2.11 – Exemplo de imagem digital (a) Tons de cinza, e (b) RGB. . . . .	32
Figura 3.1 – Exemplos de lesões de pele do dataset ISIC (imagens de acesso aberto). . . . .	35
Figura 3.2 – Arquitetura AlexNet modificada para nosso problema de classificação. . . . .	41
Figura 3.3 – Arquitetura VGG16 modificada para nosso problema de classificação. . . . .	42
Figura 3.4 – Arquitetura VGG19 modificada para nosso problema de classificação. . . . .	43
Figura 4.1 – Desempenho do treinamento no Modelo A (a) <i>Accuracy</i> , e (b) <i>Loss</i> . . . . .	47
Figura 4.2 – Desempenho do treinamento no Modelo B (a) <i>Accuracy</i> , e (b) <i>Loss</i> . . . . .	47
Figura 4.3 – Comparação da <i>accuracy</i> dos três otimizadores no conjunto de valida- ção e teste do Modelo C. . . . .	50
Figura 4.4 – Desempenho do treinamento no Modelo C com o otimizador Adam na arquitetura AlexNet (a) <i>Accuracy</i> , e (b) <i>Loss</i> . . . . .	50
Figura 4.5 – Comparativa da <i>accuracy</i> dos três otimizadores no conjunto de valida- ção e teste do Modelo C. . . . .	52
Figura 4.6 – Desempenho do treinamento no Modelo C com o otimizador RMSprop na arquitetura VGG16 (a) <i>Accuracy</i> , e (b) <i>Loss</i> . . . . .	52
Figura 4.7 – Desempenho do treinamento no Modelo C com o otimizador RMSprop na arquitetura VGG19 (a) <i>Accuracy</i> , e (b) <i>Loss</i> . . . . .	54
Figura 4.8 – Comparativa da <i>accuracy</i> das três arquiteturas no conjunto de valida- ção e teste do Modelo C. . . . .	54
Figura 4.9 – Matriz de confusão da arquitetura AlexNet e o otimizador Adam do Modelo C. . . . .	55
Figura 4.10 – Matriz de confusão da arquitetura VGG16 e o otimizador RMSprop do Modelo C. . . . .	56

Figura 4.11–Matriz de confusão da arquitetura VGG19 e o otimizador RMSprop do Modelo C. . . . .	57
Figura 4.12–Resumo das imagens de teste com cada uma das arquiteturas treinadas com o Modelo C. . . . .	58
Figura 4.13–ROC dos três classificadores. . . . .	59

# Lista de tabelas

Tabela 1.1 – Estimativas de casos de não-melanoma por regiões. . . . .	17
Tabela 3.1 – Modelo A . . . . .	35
Tabela 3.2 – Modelo B . . . . .	35
Tabela 3.3 – Modelo C . . . . .	35
Tabela 3.4 – Transformações propostas . . . . .	36
Tabela 3.5 – Matriz de confusão para classificação binária. . . . .	37
Tabela 3.6 – Especificações do ambiente Kaggle . . . . .	39
Tabela 4.1 – Comparações dos Modelos A e B. . . . .	48
Tabela 4.2 – Avaliação do otimizador SGD sobre o conjunto de teste . . . . .	48
Tabela 4.3 – Avaliação do treinamento com o modelo C e a arquitetura AlexNet. . .	49
Tabela 4.4 – Avaliação do treinamento com o modelo C e a arquitetura VGG16. . .	51
Tabela 4.5 – Avaliação do treinamento com o modelo C e a arquitetura VGG19. . .	53
Tabela 4.6 – Medidas da área sob a curva AUC, das três arquiteturas. . . . .	59

## Lista de Abreviaturas, Acrônimos e Siglas

---

Abreviatura	Conotação
ACC	do inglês <i>"accuracy"</i>
AUC	do inglês <i>"area under the curve"</i>
AWS	do inglês <i>"amazon web service"</i>
BatchNorm	do inglês <i>"batch normalization"</i>
CBC	carcinoma basocelular
CEC	carcinoma espinocelular
CNN	do inglês <i>"convolutional neural networks"</i>
CPU	do inglês <i>"central processing unit"</i>
DL	do inglês <i>"deep learning"</i>
DNA	ácido dextrorribonucleico
FN	falso negativo
FP	falso positivo
GPU	do inglês <i>"graphics processing unit"</i>
IA	inteligencia artificial
ISIC	do inglês <i>"international skin imaging collaboration"</i>
ML	do inglês <i>"machine learning"</i>
MLP	do inglês <i>"multi-layer perceptron"</i>
RAM	do inglês <i>"random-access memory"</i>
ReLU	do inglês <i>"rectified linear unit"</i>
RGB	do inglês <i>"Red, Green, Blue"</i>
ROC	do inglês <i>"receiver operator characteristic"</i>
SGD	do inglês <i>"stochastic gradient descent"</i>
TFP	taxa de falso positivo
TVP	taxa de verdadeiro positivo
VGG	do inglês <i>"visual geometry group"</i>
VN	verdadeiro negativo
VP	verdadeiro positivo

---

## Lista de Símbolos

---

Símbolos	Conotação
$b_k$	Limiar ou bias
$d()$	Diferencial
$f$	Função
$f(u_k)$	Função de ativação
$f(x, y)$	Função bidimensional
$J(\Theta)$	Função de custo
$N$	Quantidade de imagens
$p$	Probabilidade
$\mathbb{R}$	Elemento Real
$w_{kn}$	Pesos sinápticos
$w(n)$	Resposta ao impulso
$x$	Vetor
$x_i$	Elemento do vetor
$x_n$	Entrada do perceptron
$x(n)$	Sinal discreto
$(x, y)$	Coordenadas espaciais
$y_{C1}^+$	Classe melanoma
$y_{C2}^+$	Classe não-melanoma
$y_i$	Valor verdadeiro da classe
$\hat{y}_i$	Valor estimado da classe
$y_k$	Saída do perceptron
$y(n)$	Sequencia de saída
$\mu$	Valor meio
$\sigma$	Desvio padrão

---

# Sumário

<b>1</b>	<b>Introdução</b>	<b>16</b>
1.1	Motivação e Trabalhos Relacionados	17
1.2	Objetivos	18
1.3	Estrutura da dissertação	19
<b>2</b>	<b>Conceitos Básicos</b>	<b>20</b>
2.1	<i>Machine Learning</i> (ML)	20
2.1.1	A Tarefa T	20
2.1.1.1	Classificação	21
2.1.2	A Medida do Desempenho P	21
2.1.3	A experiência E	21
2.1.3.1	Aprendizagem supervisionada	21
2.1.4	Capacidade de generalização dos dados	22
2.1.4.1	<i>Overfitting</i> e <i>underfitting</i>	22
2.2	Motivação para o uso de <i>Deep Learning</i>	23
2.3	Redes Neurais Artificiais	24
2.4	Redes Neurais Convolucionais	26
2.4.1	<i>Pooling</i>	28
2.5	Arquitetura típica CNN	29
2.6	Câncer de Pele	29
2.6.1	Biologia da pele	29
2.6.1.1	Epiderme	29
2.6.1.2	Derme	30
2.6.1.3	Hipoderme	30
2.6.2	Não-melanoma	31
2.6.3	Melanoma	31
2.7	Imagens médicas	31
2.8	Imagem digital	32
<b>3</b>	<b>Metodologia</b>	<b>34</b>
3.1	<i>Dataset</i>	34
3.1.1	Arquivo de dados ISIC	34
3.1.2	Divisão do conjunto de dados	34
3.2	Aumento de dados	36
3.3	Métricas de avaliação	36
3.3.1	<i>Accuracy</i> (exatidão)	37

3.3.2	<i>Loss</i> (erro)	37
3.3.3	Matriz de confusão	37
3.3.4	Curva ROC e a área sob a curva AUC	38
3.4	Implementação	39
3.4.1	Linguagem de programação e framework	39
3.4.2	Experimentos e hardware	39
3.5	Resolução das imagens	39
3.6	Normalização	40
3.7	Arquiteturas de <i>Deep Learning</i>	40
3.7.1	AlexNet	40
3.7.2	VGG16	41
3.7.3	VGG19	43
3.8	Função custo	44
3.9	Otimizadores	44
3.10	Parada antecipada ( <i>EarlyStopping</i> )	45
<b>4</b>	<b>Resultados e Discussões</b>	<b>46</b>
4.1	Resultados empregando a arquitetura AlexNet	46
4.1.1	Resultados do treinamento com o Modelo A	46
4.1.2	Resultados do treinamento com o Modelo B	46
4.2	Resultados do treinamento com o Modelo C	48
4.2.1	Treinamento com a arquitetura Alexnet	49
4.2.2	Treinamento com a arquitetura VGG16	51
4.2.3	Treinamento com a arquitetura VGG19	53
4.2.4	Comparativa das arquiteturas com o treinamento do Modelo C	53
4.2.4.1	Matriz de confusão com a arquitetura AlexNet	55
4.2.4.2	Matriz de confusão com a arquitetura VGG16	56
4.2.4.3	Matriz de confusão com a arquitetura VGG19	56
4.2.4.4	Curvas ROC e a área sob a curva AUC	58
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>60</b>
5.1	Conclusões	60
5.2	Trabalhos Futuros	61
	<b>Referências</b>	<b>63</b>
	<b>APÊNDICE A Listado de Publicações</b>	<b>68</b>

# 1 Introdução

Existem vários tipos de lesões de pele, que podem ser classificadas em dois tipos: melanoma e não-melanoma (INCA, 2016). O melanoma é a lesão de pele mais perigosa e a mais provável de se espalhar sobre outras partes do corpo, se não for detectada, controlada e tratada precocemente. O melanoma é produzido pela reprodução anormal dos melanócitos (células que dão a cor da pele), que começam a crescer rapidamente e fora de controle metástase (SOCIETY, 2017c).

Por outro lado, os raios ultravioletas, produzidos pelo Sol, são uns dos principais fatores na geração de neoplasias cutâneas do tipo (melanoma), uma vez que a radiação tem energia suficiente para eliminar um elétron (efeito ionizador) de um átomo ou uma molécula. Este tipo de radiação pode danificar o DNA (ácido dextrorribonucleico) das células e pode produzir o câncer (SOCIETY, 2017b).

O melanoma pode ser produzido em qualquer parte do corpo, no entanto, é mais comum no rosto no pescoço ou nos braços, pois são lugares mais suscetíveis e expostos aos raios solares. Por outro lado, o risco de ter melanoma é vinte vezes maior nas pessoas de pele branca do que pessoas com a cor de pele mais escura (SOCIETY, 2019). Outro fator de risco, para o surgimento de melanoma, é o fator genético, ou seja, que ocorre devido aos parentes de primeiro grau de consanguinidade (pais, irmãos, irmãs ou filhos), que tiveram melanoma. Segundo (SOCIETY, 2017a), 10% das pessoas com melanoma têm histórico familiar, relacionado a lesão de pele.

No Brasil, o câncer de pele é um dos tipos de câncer mais comuns e corresponde a 30% de todos os tumores malignos registrados, sendo que o melanoma representa apenas 3% de malignidade neoplástica. Embora sua incidência seja baixa, ele se torna o mais grave, devido à alta probabilidade de causar metástases (INCA, 2018), resultando uma alta taxa de mortalidade.

No Brasil, há uma estimativa de 2920 novos casos em homens e de 3340 novos casos em mulheres a cada ano. A grande maioria destes casos são encontrados na região sul do país. Um outro tipo de neoplasia cutânea é o não-melanoma e tem uma estimativa de 85170 novos casos para homens, e de 80140 para mulheres por ano. A Tabela 1.1 mostra a incidência do não-melanoma por regiões, na qual a região sul do país apresenta maior quantidades de casos por cada 100 mil habitantes (INCA, 2016).

Tabela 1.1 – Estimativas de casos de não-melanoma por regiões.

Região	Homens	Mulheres
Sul	160,80/100 mil	97,64/100 mil
Sudeste	89,80/100 mil	95,16/100 mil
Centro-Oeste	69,27/100 mil	92,66/100 mil
Nordeste	53,75/100 mil	45,59/100 mil
Norte	23,74/100 mil	27,71/100 mil

## 1.1 Motivação e Trabalhos Relacionados

A detecção precoce das neoplasias cutâneas é uma estratégia para encontrar o tumor nas suas fases iniciais, permitindo assim um maior chance de tratamento. Por este motivo, nos últimos anos, as ferramentas computacionais na medicina têm assumido um papel crucial na tomada de decisões, contribuindo como um diagnóstico complementar, aumentando a probabilidade de um melhor diagnóstico do melanoma, entre 85% e 95% nos tipo de lesões de pele do paciente (CASTRO, 2018).

A detecção do melanoma nas fases iniciais é importante, pois apresenta uma taxa de sobrevida de um 99% nos primeiros 5 anos. No entanto, a detecção nas fases finais decai para uma taxa de sobrevida de 14% (ESTEVA *et al.*, 2017).

Estudos recentes no aprendizado profundo ou *deep learning* (DL), na qual é uma subárea do aprendizado de máquina ou *machine learning* (ML), têm mostrado resultados com desempenhos aceitáveis no instante de classificar, detectar, segmentar e reconhecer imagens médicas. Assim sendo, ferramentas de visão computacional permitem criar e executar assistentes para a análise quantitativa de lesões na pele (CURIELLEWANDROWSKI *et al.*, 2019).

Devido à melhora no desempenho computacional, obtém-se maiores capacidades de processamento e diversos serviços de GPU ("*Graphics Processing Unit*") na nuvem, tais como *Colaboratory* da Google, o Kaggle e o AWS da Amazon, dentre outros (CHOLLET, 2018).

Na atualidade é possível obter bons resultados no desempenho e na *accuracy* (exatidão) sobre diferentes tarefas usando imagens médicas. Algoritmos de DL que empregam arquiteturas de redes neurais convolucionais CNN ("*Convolutional Neural Networks*") têm mostrado excelentes resultados, devido a bancos de dados com uma grande quantidade de imagens. Entretanto, problemas como o desequilíbrio das classes, rótulos errados, entre outros, afetam a *accuracy* da arquitetura (MAHAJAN *et al.*, 2018).

O objetivo principal é desenvolver um sistema de ajuda para reconhecimento e classificação de imagens médicas que possua diversas lesões de pele, facilitando o diag-

nóstico e o reconhecimento das lesões de pele, os quais são, tradicionalmente, feitos de forma visual pelo médico dermatologista, cujo processo pode ser complicado e conduzir a resultados subjetivos (BRINKER *et al.*, 2019), (HAN *et al.*, 2018).

O reconhecimento visual feito pelos dermatologistas, baseia-se na avaliação denominada "ABCDE", o acrônimo refere-se às iniciais das características Assimetria, Bordas irregulares, Cores, Diâmetro (maior que 6mm), e finalmente Evolução da lesão (NACHBAR *et al.*, 1994). Outras regras ou avaliações similares incluem uma lista de verificação de 3 pontos (SOYER *et al.*, 2004), lista de verificação de 7 pontos (ARGENZIANO *et al.*, 2011), e a avaliação "CASH", que se refere ao acrônimo oriundo dos termos Cor, Arquitetura, Simetria e a Homogeneidade da lesão de pele (HENNING *et al.*, 2007).

Segundo (ESTEVA *et al.*, 2017) apresentam uma CNN treinada desde o princípio até o fim, na qual a tarefa de classificação que foi implementada utilizou uma base de dados proprietária com 129450 imagens clínicas rotuladas por dermatologistas, dentre as quais estavam inclusas 3374 imagens dermoscópicas. Os autores empregaram a arquitetura GoogLeNet Inception v3 (SZEGEDY *et al.*, 2015), a mesma que foi pré-treinada com as imagens da ImageNet (DENG *et al.*, 2009), a qual possui 1,28 milhão de imagens divididas em 1000 classes, dita rede pré-treinada, a qual é utilizada como aprendizagem por transferência, que em inglês significa ("*transfer learning*"). A CNN atingiu uma precisão de 72,1% a mesma que foi comparada com dois dermatologistas que alcançaram precisão de 65,56% e 66% respectivamente.

Diante do exposto nesta pesquisa visamos obter o panorama do estado da arte referente aos algoritmos, arquiteturas, e modelos de *deep learning* usados para a tarefa de classificação de imagens dermoscópicas com diversas lesões de pele. Especificamente, nós focaremos na classificação do melanoma e do não-melanoma com intuito de atingir melhores desempenhos na classificação das imagens médicas.

## 1.2 Objetivos

O objetivo principal da pesquisa é explorar abordagens com *Deep Learning* na tarefa como classificador binário automático para imagens dermoscópicas com lesões de pele entre o melanoma (neoplasia maligna) e o não-melanoma (neoplasia benigna). Portanto, o trabalho possui os seguintes objetivos específicos:

- Fornecer uma revisão crítica das abordagens existentes para a tarefa de classificação de imagens presentes na literatura, fazendo uso de técnicas de *deep learning*.
- Provar e treinar algumas arquiteturas de *deep learning* modificadas e adequadas as nossas necessidades.

- Melhorar o desempenho das arquiteturas treinadas no problema de classificação automática de lesões de pele.
- Apresentar e comparar os resultados da classificação automática através de diversas métricas, que permitam avaliar o desempenho da arquitetura de *deep learning* proposta.

### 1.3 Estrutura da dissertação

Para um maior entendimento, a dissertação está organizada da seguinte forma:

- O Capítulo 2 aborda os conceitos básicos sob o que é *machine learning*, *deep learning*, as redes neurais artificiais, assim como uma sucinta descrição da pele e como são ocasionadas as neoplasias cutâneas do tipo melanoma e não-melanoma, todos estes conceitos estão focados no nosso trabalho.
- O Capítulo 3 apresenta a metodologia do trabalho, assim como o conjunto de dados empregados, as arquiteturas treinadas, e as métricas com as quais avaliamos o projeto.
- O Capítulo 4 descreve os resultados obtidos em cada um dos treinamentos conjuntamente com as métricas propostas na metodologia.
- Finalmente, o Capítulo 5 aborda as conclusões mais relevantes da dissertação, assim como os trabalhos futuros.

## 2 Conceitos Básicos

Neste capítulo serão abordados os conceitos básicos sobre DL, assim como uma descrição das redes convolucionais, filtros convolucionais, parâmetros de configuração (hiperparâmetros) e metodologias para diminuir o *overfitting* nas arquiteturas.

O DL, é uma subárea do *machine learning* (ML), que por sua vez é uma subárea de inteligência artificial (IA) como mostra a Figura 2.1. Para conseguir compreender DL, primeiramente é necessário entender o que é ML e como o DL foi tomando força nestes últimos anos (BUDUMA; LOCASCIO, 2017).

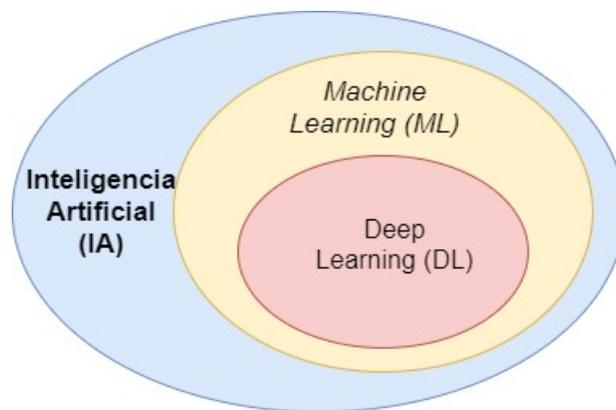


Figura 2.1 – Inteligência artificial, *machine learning* e *deep learning*.

### 2.1 *Machine Learning* (ML)

ML é a capacidade que uma máquina ou computador possui para aprender a partir dos dados com a ajuda da programação. Um conceito mais formal foi dado por Mitchell (MITCHELL, 1997), onde diz - "*Que um programa de computador aprende a partir de uma experiência  $\mathbf{E}$  com respeito a alguma tarefa  $\mathbf{T}$  e uma medida de desempenho  $\mathbf{P}$ , se este desempenho sobre  $\mathbf{T}$ , é medido por  $\mathbf{P}$ , ele melhorará com a experiência  $\mathbf{E}$* ".

#### 2.1.1 A Tarefa $\mathbf{T}$

A tarefa faz referência ao caminho como a máquina ou o computador vai conseguir aprender através dos exemplos ou dos dados. Neste caso, o exemplo será representado por um conjunto de características as quais são um objeto ou um evento. Porém, o processo de aprendizado de ML receberá uma entrada dada por um vetor  $\mathbf{x} \in \mathbb{R}^n$ , onde cada elemento  $x_i$  é uma característica. No caso de imagens, cada característica é um valor do *pixel* (GOODFELLOW *et al.*, 2016).

Existem varias tarefas que podem ser desenvolvidas pelo ML tais como: regressão, segmentação, clusterização entre outras. No entanto, no projeto vamos manter o foco em problemas de classificação.

### 2.1.1.1 Classificação

Nesta tarefa, o algoritmo de aprendizagem tenta encontrar ou produzir a função que descreve uma tarefa  $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$  onde  $y = f(\mathbf{x})$ , o vetor de entrada  $\mathbf{x}$  representa as características da imagem,  $y$  o rótulo ou *label* que identifica a classe que pertence, e  $k$  são as categorias ou classes do classificador.

## 2.1.2 A Medida do Desempenho P

Para medir o desempenho do ML, existem vários métodos que ajudarão na melhora do algoritmo de aprendizagem, entre eles os mais empregados para conhecer como o algoritmo se comporta é a *accuracy* (exatidão), a mesma que vai dizer qual porção de exemplos produz as saídas verdadeiras. No caso da classificação, a *accuracy* representa quanta porcentagem das imagens têm o rótulo verdadeiro da classe à qual pertencem. Outra medida comumente utilizada é o *loss* (erro), o qual mostra a porcentagem de exemplos que estão produzindo uma saída errada (VANHEULE; PIŽURICA, 2018).

Todas as medidas de desempenho para a aprendizagem são feitas nos três conjuntos de dados, as quais são de treinamento, de validação, e de teste. O conjunto de treinamento permite conhecer como está acontecendo a aprendizagem. As medidas no conjunto de validação ajudam a modificar os hiperparâmetros para a aprendizagem. Finalmente, o conjunto de teste mostra como o algoritmo se comporta para dados novos.

## 2.1.3 A experiência E

A forma como o algoritmo de ML vai aprender é conhecida como experiência, a qual vai depender do tipo de tarefa que vamos ensinar para o algoritmo. A experiência de aprendizagem é dada pelo conjunto de dados ou *dataset*, o qual é um conjunto de vários exemplos de algum assunto. No nosso caso, vamos utilizar o método supervisionado (ESTEVA, 2018), o qual faz referência a todos os conjuntos de exemplos ou amostras que possuem um rótulo próprio.

### 2.1.3.1 Aprendizagem supervisionada

A experiência do conjunto de dados, os quais contêm os exemplos com as suas características e além disso, cada um dos exemplos estão associados a um rótulo ou *label*

como se mostra na Equação 2.1.

$$p(y | \mathbf{x}) = \frac{p(\mathbf{x} | y) p(y)}{p(\mathbf{x})} \quad (2.1)$$

Onde cada um dos exemplos do conjunto de dados representam um vetor aleatório de características  $\mathbf{x}$  associados a um valor ou vector  $y$ . Assim, a aprendizagem vai conseguir prever  $y$  baseando-se em  $\mathbf{x}$ , dado pela estimativa ou probabilidade condicional  $p(y | \mathbf{x})$ . O termo de aprendizagem supervisionada origina-se através das visualizações do alvo  $y$  providenciados pelo instrutor que mostra ao ML como fazê-lo (GOODFELLOW *et al.*, 2016).

#### 2.1.4 Capacidade de generalização dos dados

O desafio principal da tarefa de ML, é que a aprendizagem consiga o melhor desempenho possível sob o conjunto de teste. A habilidade do modelo de se comportar bem sob este conjunto de dados não conhecidos durante o processo de treinamento, e é chamado de "generalização" dos dados (GOODFELLOW *et al.*, 2016), a Figura 2.2 mostra a capacidade de generalização dos dados.

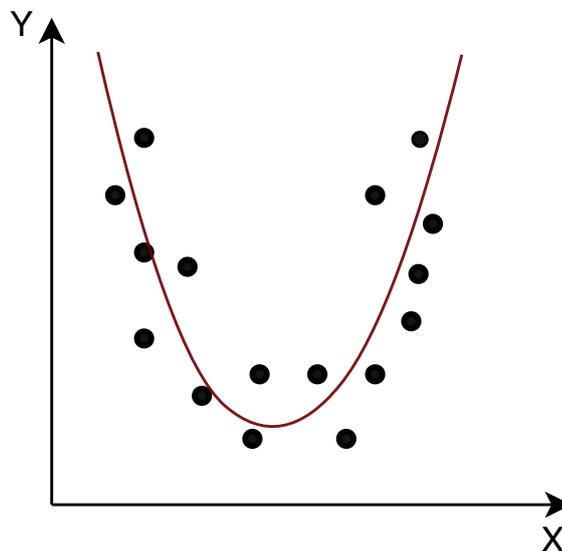


Figura 2.2 – Generalização.

##### 2.1.4.1 *Overfitting* e *underfitting*

Diante do exposto, durante a aprendizagem deseja-se evitar duas situações que vão comprometer o uso do modelo as quais são:

- *Overfitting* ou sobreajuste: mostrada na Figura 2.3a, ocorre quando a aproximação do modelo adaptou-se perfeitamente ao conjunto de dados do treinamento redu-

zindo ao máximo o *loss*, porém não tem aprendido as características necessárias para classificar novos dados (VANHEULE; PIŽURICA, 2018).

- *Underfitting* ou subajuste: nesta situação o modelo não consegue ajustar aos dados utilizados no treinamento, ocasionando um alto índice no valor de *loss* alto (PATTERSON; GIBSON, 2017), como apresenta a Figura 2.3b.

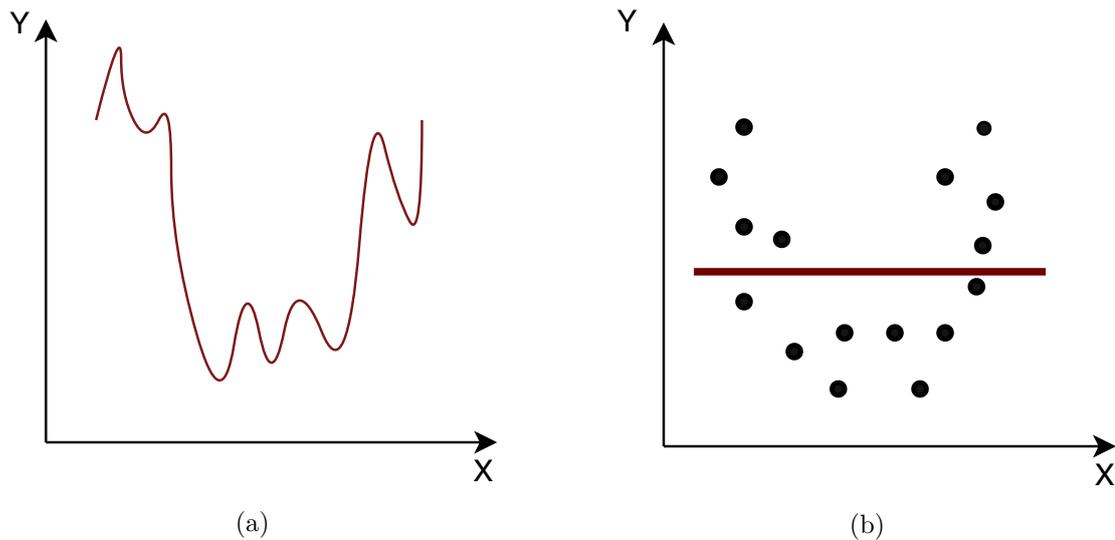


Figura 2.3 – (a) *Overfitting* ou sobreajuste, e (b) *Underfitting* ou subajuste.

## 2.2 Motivação para o uso de *Deep Learning*

Devido ao crescimento na dimensionalidade dos problemas, o bom desempenho dos algoritmos clássicos de ML começou a decair na habilidade de generalização dos dados (VANHEULE; PIŽURICA, 2018). No caso de problemas com maior complexidade, como reconhecimento de objetos ou visão computacional, esses algoritmos clássicos tendem a apresentar falhas no espaço de alta dimensionalidade.

No reconhecimento de objetos, o uso de DL permite ter mais níveis de abstração, de forma que as estruturas específicas da imagem, simplifica a tarefa de classificação (VANHEULE; PIŽURICA, 2018)(LECUN *et al.*, 2015). Dita propriedade de extrair características em conjunto com a ajuda de outras técnicas como a transferência de aprendizagem, tem mostrado resultados bem promissores para tarefas de classificação (PATTERSON; GIBSON, 2017).

Anteriormente, este problema era dividido em duas operações principais:

- A primeira parte era a construção de um extrator de características altamente específico para certas tarefas.

- Posteriormente um classificador dos recursos extraídos era realizado.

No entanto, o DL, permite aprofundar e descobrir níveis mais altos de abstração os quais são úteis no processo de classificação, sem a necessidade ou interferência de um *designer*, e é aqui onde está a força principal do DL (LECUN *et al.*, 2015).

## 2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais "ANN" ("*Artificial Neural Networks*"), são representações inspiradas pelas estruturas dos neurônios do cérebro humano, o mesmo que está composto por bilhões de neurônios interconectados que são responsáveis pela tomada de decisões.

A Figura 2.4 mostra o neurônio biológico que recebe estímulos eletroquímicos (sinapses), a partir dos neurônios vizinhos através dos dendritos. O núcleo do neurônio processará aqueles estímulos para serem integrados, e levará à geração de uma resposta elétrica a qual será enviada pelo axônio, que é encarregado de levar a informação processada para o neurônio seguinte, através de sinapses.

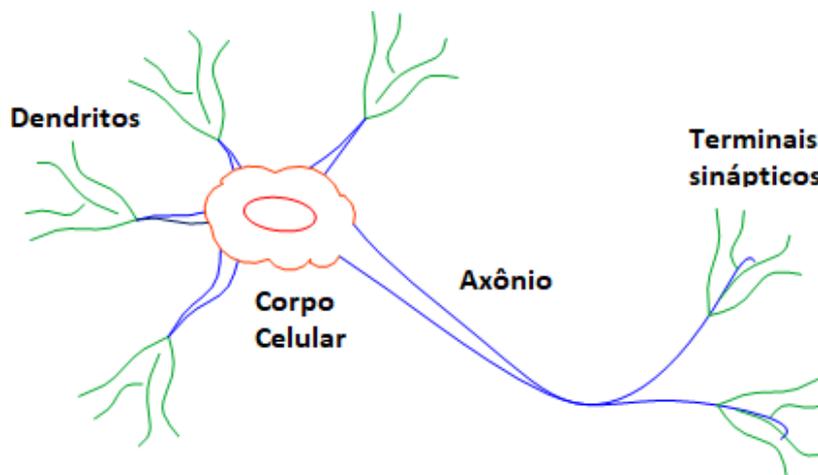
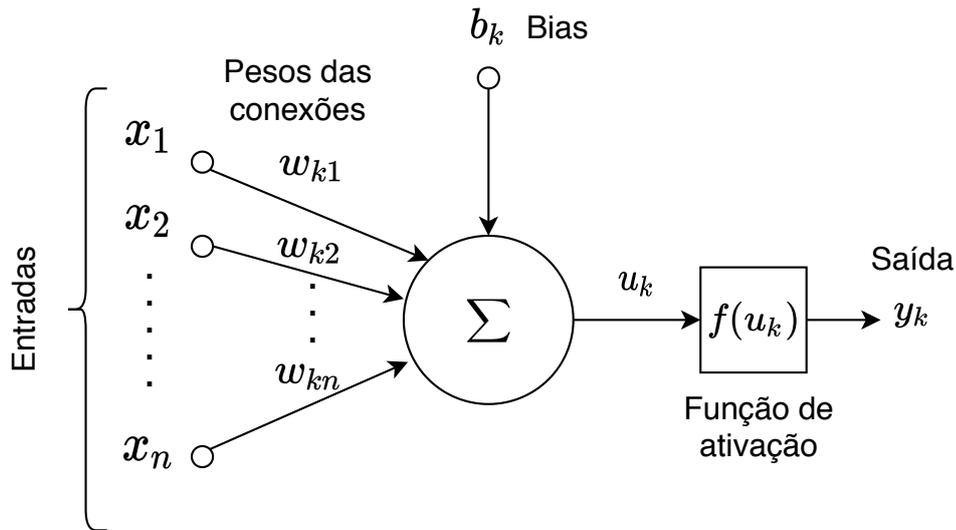


Figura 2.4 – Neurônio.

(MCCULLOCH; PITTS, 1943) propuseram o primeiro modelo matemático, onde assumiam que o neurônio biológico está executando uma função lógica. Dentre as principais premissas foi que a atividade do neurônio é binária denominada de "tudo ou nada", sendo dois estados lógicos 1 para ativo, e 0 para inativo. Deste modelo proposto, foi possível o estabelecimento da relação entre um neurônio e a lógica proposicional.

A partir deste primeiro modelo matemático, (ROSENBLATT, 1958) propôs o modelo conhecido como *perceptron*, o qual dá uma ideia geral de como vai funcionar uma rede neural. Então o *perceptron* também é um modelo matemático, o qual receberá várias entradas  $x_1, x_2, x_3, \dots, x_n$  e produzirá uma única saída com resposta binária.

Figura 2.5 – *Perceptron*.

Como mostra a Figura 2.5, o modelo do *perceptron* tem as entradas  $x_1, x_2, \dots, x_n$  a proposta do Rosenblatt é que para ser calculada a saída são introduzidos os pesos das conexões  $w_{k1}, w_{k2}, \dots, w_{kn}$  a cada uma das entradas representadas por números reais e que mostram a importância das entradas.

A ideia é que a saída representa a ativação do neurônio, ativada pelos estímulos de entrada sendo uma combinação linear das entradas  $x_n$ , ponderadas pelos pesos sinápticos  $w_{kn}$ . Se dita saída ou ativação excede certo limiar (*threshold*), controlado pelo *bias*  $b_k$ , ocorre um "disparo", o qual pode ser representado como a função de ativação  $f(u_k)$  do tipo degrau dada pela Equação 2.2. Cabe destacar que o *perceptron* só é capaz de classificar aqueles dados que sejam linearmente separáveis (MCCULLOCH; PITTS, 1943).

$$y_k = f(u_k) = f\left(\sum_{i=1}^m x_i w_{ki} + b_k\right) = \begin{cases} 1, & \text{se } \sum_{i=1}^m x_i w_{ki} + b_k \geq 0 \\ 0, & \text{se } \sum_{i=1}^m x_i w_{ki} + b_k < 0 \end{cases} \quad (2.2)$$

O *perceptron* só tem uma única camada, o que limita o seu poder de atuação apenas para separação linear. Ao adicionar mais camadas intermediárias, consegue-se superar aquela limitação, que é o ponto fundamental para conhecer a razão da rede neural, denominada *perceptron* de múltiplas camadas "MLP" ("*multi-layer perceptron*").

A Figura 2.6 mostra uma rede MLP, a primeira camada é a entrada a qual recebe a informação das variáveis a serem encaminhadas durante o processo. Neste exemplo temos três camadas intermediárias que vão realizar o mapeamento não-linear, e que utilizam a informação mais explícita que está contida nos dados de entrada de acordo com a tarefa que será efetuada.

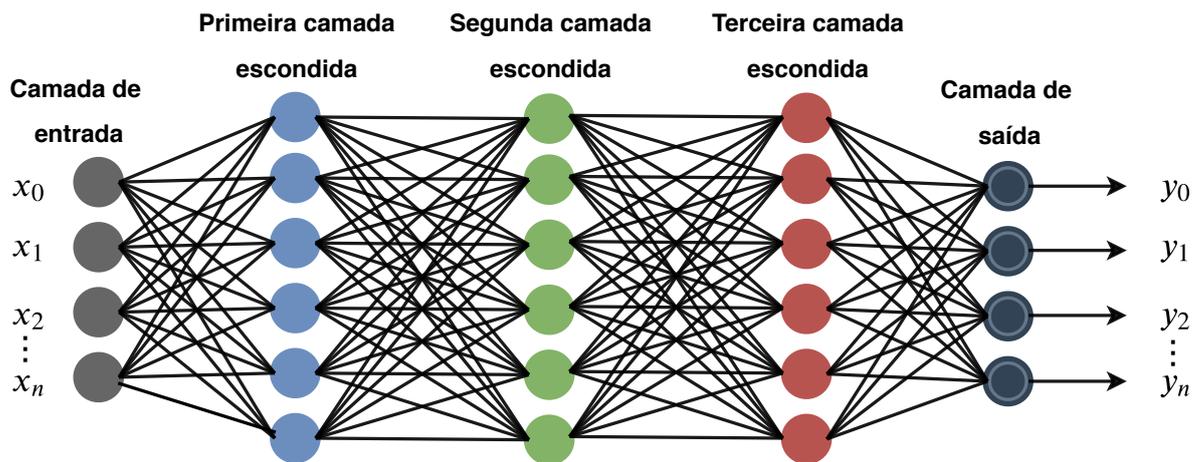


Figura 2.6 – MLP (*multi-layer perceptron*) com três camadas intermediárias ou escondidas.

A camada de saída avalia e combina as informações fornecidas pela última camada intermediária. A popularidade das redes neurais ganhou força com a reinvenção do termo retropropagação (*backpropagation*), que se refere à caminhada de frente para trás com a retro propagação do erro, permitindo resultados bons nas aplicações de reconhecimento (RUMELHART *et al.*, 1988).

## 2.4 Redes Neurais Convolucionais

As redes neurais convolucionais "CNN" (*Convolutional Neural Network*), são uma classe do DL sendo a parte fundamental nos novos avanços atingidos nesta área (DUMOULIN; VISIN, 2016). A convolução é a operação principal nas redes neurais, e é a encarregada de extrair a informação ou mapa de características.

A convolução é dada pela Equação 2.3 na qual mostra  $x(n)$  como um sinal discreto, e  $w(n)$  como a resposta ao impulso de um sistema linear e invariável em relação ao tempo, onde sua resposta será a sequência  $y(n)$  (GOODFELLOW *et al.*, 2016). Os tipos de entradas podem ser de diversas origens como trechos de sons, imagens, entre outros tipos de dados (LECUN *et al.*, 2015).

A convolução é uma operação linear a qual conserva a noção de ordem e permite explorar informações em estruturas organizadas, como series temporais ou no espaço como imagens (RAMPAZZO, 2019).

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)w(n-k) = (x * w)(n) \quad (2.3)$$

A Figura 2.7 mostra um exemplo de convolução discreta do filtrado de imagens,

onde a matriz de cor azul é a entrada  $X_{m \times n}$  conhecida como mapa de características de entrada com dimensão  $5 \times 5$ . O filtro ou *kernel*  $K_{ij}$  de valores de dimensão  $3 \times 3$  de cor cinza, é deslizado através da matriz de entrada.

Neste exemplo o passo ou *stride* é de  $1 \times 1$ , isso faz com que a janela do *kernel* movimente-se um passo por vez.

Cada posição da matriz resposta gerada  $Y_{3 \times 3}$  de cor verde é preenchida pelo resultado da soma total de cada um dos elementos *kernel*  $K_{ij}$ , multiplicado por cada elemento do mapa de características de entrada  $X_{ij}$ , onde o *kernel* vai movimentar-se por toda a entrada.

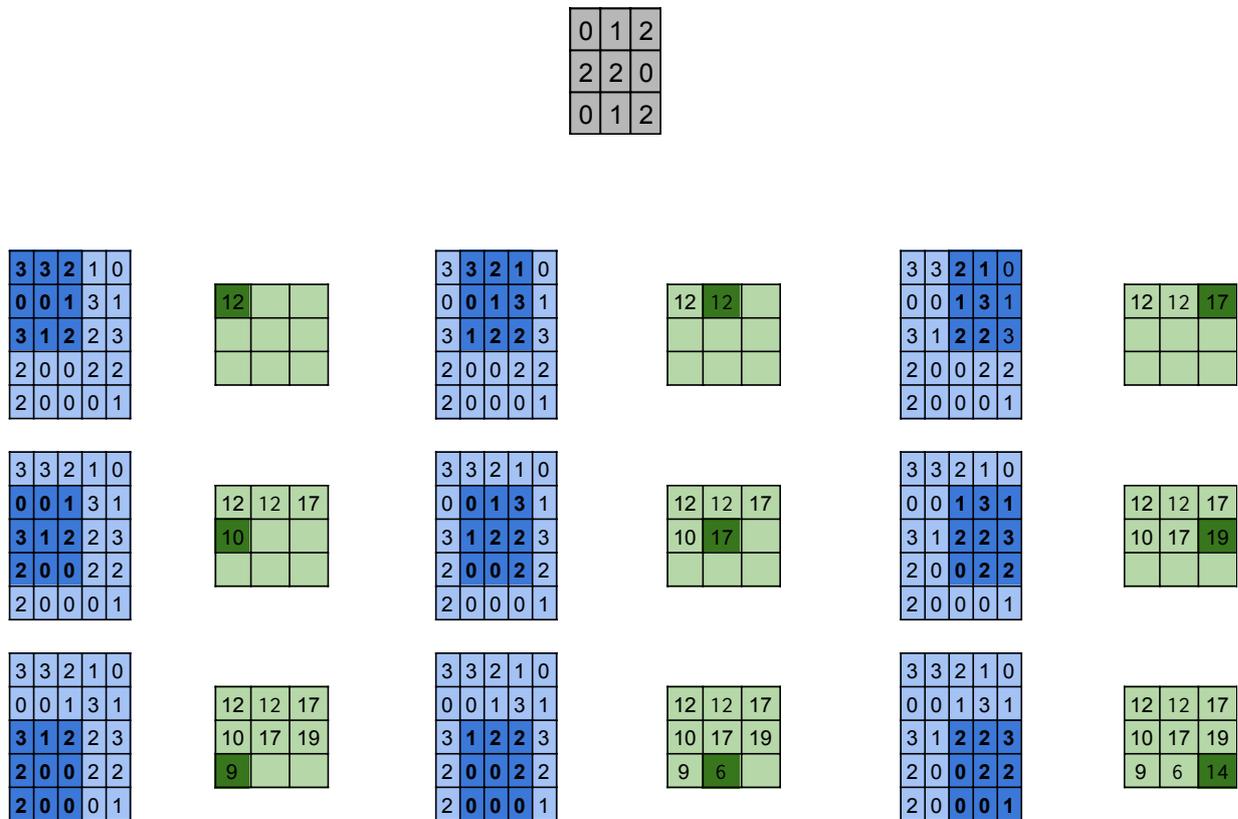


Figura 2.7 – Calculando os valores de saída de um exemplo de convolução.

A Equação 2.4 mostra a resposta do primeiro elemento da matriz resposta  $Y_{1,1}$ .

$$Y_{3 \times 3} = \sum_{i=1}^n \sum_{j=1}^m X_{ij} K_{ij} \tag{2.4}$$

$$Y_{1,1} = X_{11}K_{11} + X_{12}K_{12} + X_{13}K_{13} + X_{21}K_{21} + X_{22}K_{22} + X_{23}K_{23} + X_{31}K_{31} + X_{32}K_{32} + X_{33}K_{33}$$

$$Y_{1,1} = (3 * 0) + (3 * 1) + (2 * 2) + (0 * 2) + (0 * 2) + (1 * 0) + (3 * 0) + (1 * 1) + (2 * 2)$$

$$Y_{1,1} = 12$$

Nas redes neurais convolucionais CNN's o *kernel* vai aprendendo a partir dos dados com o objetivo de encontrar os melhores valores, para extrair os mapas de características e a informação mais explícita da imagem. No caso de múltiplas entradas, o *kernel* deve ser da mesma dimensionalidade da entrada, para conseguir efetuar a operação de convolução.

O uso das camadas convolucionais em imagens tem duas características interessantes as quais são:

- Depois que uma camada convolucional aprendeu sobre um certo padrão na imagem, ele vai conseguir reconhecê-lo em qualquer outro lugar da imagem, isso faz destas camadas convolucionais eficientes as variações de traslações dos padrões. As camadas convolucionais necessitam de treinamento com varias amostras para ter uma alta generalização (CHOLLET, 2018).
- As camadas convolucionais aprendem de hierarquias espaciais dos padrões, isto significa que a primeira camada convolucional poderia aprender de pequenos padrões como as bordas, a segunda camada poderia aprender maiores padrões fornecidos pelas caraterísticas aprendidas pela camada anterior, e assim por diante (CHOLLET, 2018).

### 2.4.1 Pooling

O seguinte estágio das camadas convolucionais na CNN, é a aplicação da operação de *pooling* (NIELSEN, 2015). O *pooling* é importante para tornar os dados invariantes a pequenas traslações da entrada, isto é importante para que a rede neural consiga reconhecer padrões das imagens em diferentes posições (DUMOULIN; VISIN, 2016).

A Figura 2.8 mostra a aplicação de *max pooling*, o qual escolhe o valor máximo do mapa de caraterísticas de saída na janela deslizante, avaliando entre todos os vizinhos.

A janela deslizante é de dimensão  $2 \times 2$ .

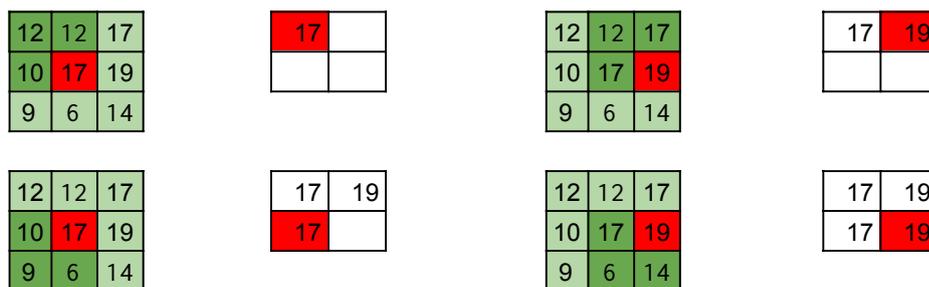


Figura 2.8 – Calculando os valores de *max pooling*.

## 2.5 Arquitetura típica CNN

Juntando todos os conceitos analisados nas seções anteriores, as redes neurais convolucionais podem ser usadas para classificar imagens, conseguindo agrupá-las por similaridade e reconhecer padrões dentro das imagens.

A Figura 2.9 mostra que a rede convolucional recebe as imagens como uma matriz ( $m \times n$ ) sendo  $m$  a largura e  $n$  a altura, no caso de imagens tridimensionais RGB ("Red - Green - Blue"), a rede é dada pela mesma matriz acrescentando a profundidade da imagem.

A matriz vai passando pelos filtros convolucionais, pelo *max pooling*, pelas funções de ativação, que mostram na sua saída o mapa de características aprendidas até chegar nas camadas densas ou totalmente conectadas e conseguir classificar as imagens.

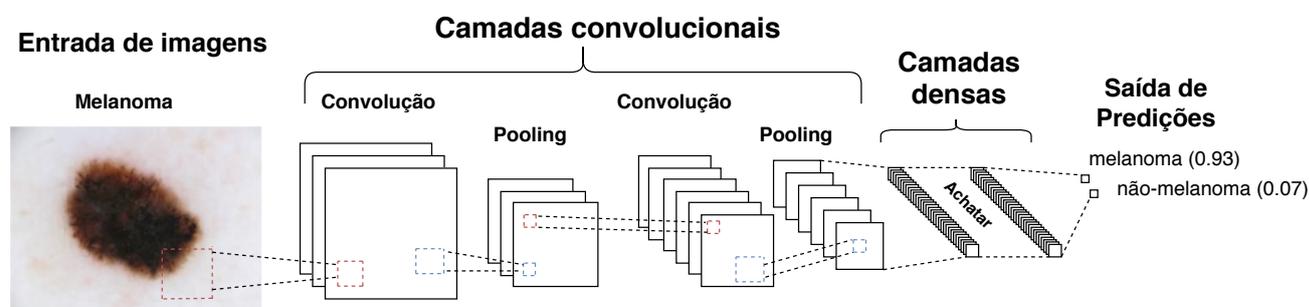


Figura 2.9 – Exemplo de uma arquitetura típica CNN.

## 2.6 Câncer de Pele

### 2.6.1 Biologia da pele

A pele é o maior órgão de defesa e revestimento externo que recobre a superfície do corpo humano com uma área média de 1,5 - 2,0 metros quadrados (CESTARI, 2012); A pele está segmentada em três camadas distintas que são: epiderme, derme, e hipoderme. (Figura 2.10). A pele desempenha múltiplas funções graças às suas camadas e sua capacidade de se adaptar às necessidades do organismo, às variações do ambiente, entre outras.

#### 2.6.1.1 Epiderme

É a camada superior e a mais superficial; sua espessura é variável e depende da região do corpo, sendo a mais grossa na palma das mãos e na planta dos pés (JUNQUEIRA; CARNEIRO, 2013). A epiderme contém varias células entre as quais encontra-se:

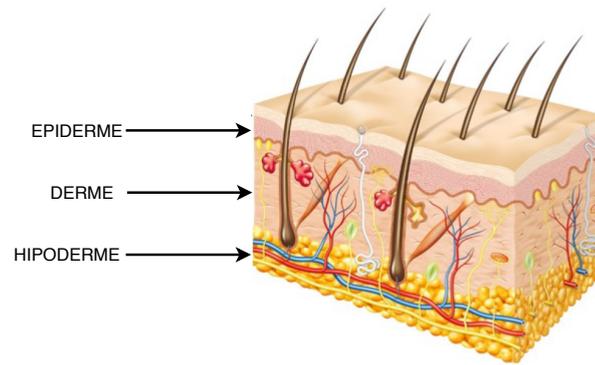


Figura 2.10 – Camadas da pele humana.

- **Células de Merkel:** São células sensoriais que tem íntimo contato com as fibras nervosas da derme e são principalmente encontradas nos lábios, dedos, gengivas, e bainha externa dos folículos pilosos.
- **Células de Langerhans:** Originadas na medula óssea que migram para se tornarem residentes na epiderme e são as encarregadas da função imunológica da pele.
- **Melanócitos:** São os encarregados de produzir a melanina que é o pigmento que determina o cor da pele e absorve a radiação ultravioleta; encontram-se na junção da derme com a epiderme. Os melanócitos são distribuídos para diversos locais do corpo humano como a pele, matriz dos pelos, mucosas, entre outras, e elas tem estreito relacionamento com as fibras nervosas da derme (CESTARI, 2012).

#### 2.6.1.2 Derme

É o tecido de sustentação da epiderme e apresenta espessura variável de 0,3 a 3,0 mm de acordo com a região do corpo (CESTARI, 2012). Possui pequenos vasos sanguíneos que são responsáveis pela oxigenação e nutrição da epiderme. Dentre os componentes principais da derme estão a elastina, que proporciona a elasticidade e o colágeno o qual dá a resistência.

#### 2.6.1.3 Hipoderme

É a camada mais profunda da pele que atua contra choques mecânicos, isolamento térmico e na reserva de energia. Este tecido armazena a gordura nas células denominadas adipócitos (CESTARI, 2012). A hipoderme conecta a pele com os ossos e os músculos.

### 2.6.2 Não-melanoma

Segundo o Instituto Nacional do Câncer (INCA, 2016) o não-melanoma representa 30% do total de casos de câncer registrados no Brasil até o ano 2018, ele é dividido em duas classes que podem ser o carcinoma basocelular (CBC) (GORDON, 2013), que origina-se nas células basais que se encontram na camada mais profunda da epiderme e é o mais prevalente (ALBERT; WEINSTOCK, 2003). A sua origem, comumente, ocorre por exposição prolongada aos raios solares, desenvolve-se lentamente e é difícil de desenvolver metástase.

O segundo tipo de neoplasia cutânea do tipo não-melanoma é o carcinoma espinocelular (CEC) (GORDON, 2013), que se manifesta nas células escamosas que se encontram em sua maioria nas camadas mais superiores da pele, a maioria dos casos também ocorre por longos períodos de tempo de exposição a luz solar, porém existem vários casos de estar associados a feridas crônicas ou cicatrizes na pele causadas pelo uso de drogas, exposição a alguns agentes químicos, radiação ou transplante de órgãos.

### 2.6.3 Melanoma

O melanoma é produzido pelo crescimento anormal e descontrolado das células melanócitos (células produtoras da melanina), causado por mutações genéticas no DNA (ácido desoxirribonucleico) (SBD, 2017), e tem uma alta probabilidade de se espalhar pelo corpo até produzir metástase, caso não seja controlada na suas fases iniciais. No Brasil representa 3% de todos os tumores malignos registrados no país (INCA, 2016), e na maioria dos casos o melanoma apresenta-se como uma pinta ou mancha sobre a pele; as partes que se encontram mais expostas ao raios do sol são comumente os principais lugares onde aparece o melanoma.

Nos dois tipos de neoplasias cutâneas analisadas acima, o tratamento mais indicado é a cirurgia para a remoção da mancha com características de melanoma. A quimioterapia e a radioterapia também podem ser utilizadas. Hoje há novos tratamentos quando a doença é avançada, onde o objetivo principal é postergar a evolução oferecendo maior chance de sobrevivida para o paciente (INCA, 2016).

## 2.7 Imagens médicas

As análises de dados médicos vêm sendo uma área de interesse neste últimos anos, (SHI *et al.*, 2009), (LITJENS *et al.*, 2017) e mostram um crescimento na área medica com o uso de redes neurais, que provêm uma grande ajuda aos médicos para a conclusão de diagnósticos com base na utilização de imagens médicas.

O crescimento do uso de sistemas automáticos capazes de ajudar a extrair informação, a partir de diversas formas de dados, com a finalidade de assistir ao médico no diagnóstico do paciente, de uma forma rápida, eficaz e segura, sobretudo na área onde é necessário ter um diagnóstico nas fases iniciais como é o caso das neoplasias cutâneas, permitindo conseguir uma melhora na sobrevivência do paciente.

A disponibilidade de imagens médicas ainda é muito escassa devido à sensibilidade dos dados, e ao fato de necessitar das anotações de médicos com especialidade na área de dermatologia e oncologia.

## 2.8 Imagem digital

Uma função bidimensional  $f(x, y)$  pode definir uma imagem digital, onde  $x$  e  $y$  são as coordenadas espaciais (que representam a altura e a largura da imagem), e  $f$  é a amplitude para qualquer par de coordenadas, conhecida como a intensidade da imagem. Cada uma destas intensidades são conhecida como *pixel*, que é uma informação necessária para formar a imagem (GONZALEZ; WOODS, 2018).

Nas imagens em tons de cinza (ver Figura 2.11a) o *pixel* descreve quão brilhante ou escuro é a intensidade do *pixel* na posição apresentada. A Figura 2.11b é uma imagem colorida que é representada pela combinação de três imagens individuais; o modelo RGB ("*Red - Green - Blue*") representa uma imagem colorida composta por três imagens individuais, referidas às componentes primárias vermelho, verde e azul.

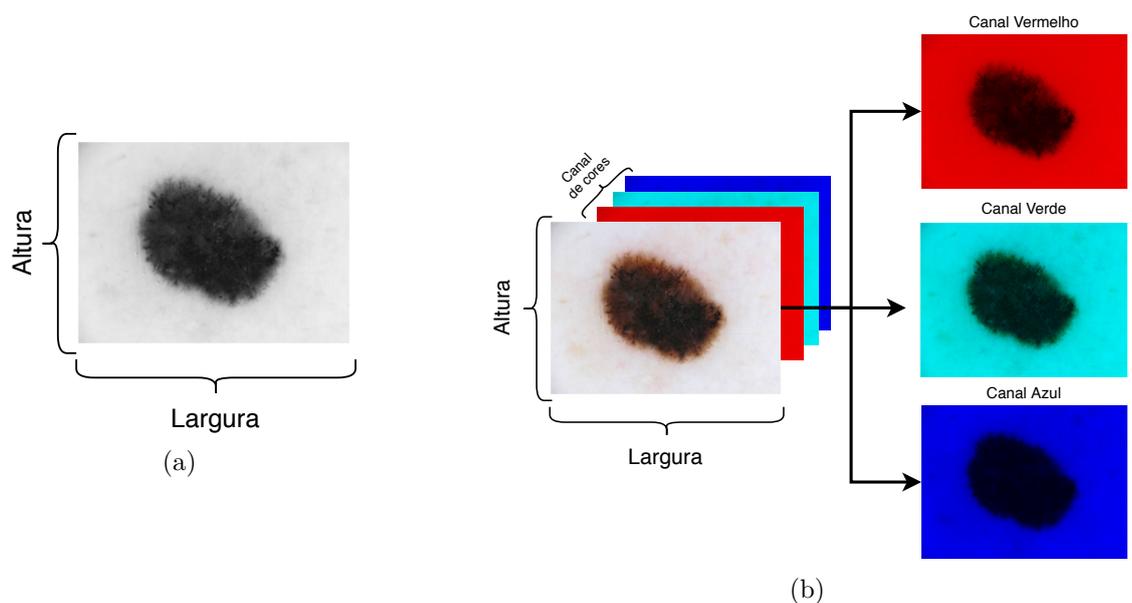


Figura 2.11 – Exemplo de imagem digital (a) Tons de cinza, e (b) RGB.

O uso de imagens coloridas do tipo RGB são os mais perceptíveis ao olho humano que consegue ver uma variabilidade de combinações destas cores primárias e, por esse motivo as imagens coloridas são um poderoso descritor na identificação e extração de características de uma cena (GONZALEZ; WOODS, 2018).

## 3 Metodologia

Neste capítulo vamos descrever a *dataset* utilizado na dissertação juntamente com as métricas usadas na avaliação das arquiteturas. Logo, apresentaremos as descrições das arquiteturas de rede neural convolucional "AlexNet", "VGG16", e "VGG19" as quais foram adaptadas ao nosso problema de classificação. Além disso, uma introdução das camadas e a seleção dos hiperparâmetros serão apresentados.

### 3.1 *Dataset*

#### 3.1.1 Arquivo de dados ISIC

O conjunto de imagens empregadas são provenientes da base de dados ISIC ("*International Skin Imaging Collaboration*") (PROJECT, 2018), a qual é de acesso livre. ISIC é uma organização que tem como objetivo principal diminuir os altos índices de morte ocasionadas pelas neoplasias malignas conhecido também como melanoma. Assim, o ISIC começou com a coleta de imagens que contém o próprio diagnóstico hispatológico.

ISIC é um coletor de dados com 23906 imagens de diferentes tipos de lesões de pele. As imagens são do tipo dermoscópicas, e se referem a uma técnica que permite visualizar a lesão da pele sem os reflexos da superfície analisada, além de obter imagens de melhor qualidade na região de interesse.

Devido ao desequilíbrio das classes, isto é diferentes quantidades de imagens que pertencem a uma lesão de pele em particular, o projeto de pesquisa tem como foco a classificação das imagens escolhendo duas classes principais, as quais são o melanoma e o não-melanoma, o qual representa uma classificação de tipo binária. A Figura 3.1 ilustra exemplos de imagens que contém o conjunto de dados da ISIC.

#### 3.1.2 Divisão do conjunto de dados

A principio iremos avaliar arquiteturas de *deep learning* com diversas quantidades de imagens as quais serão acrescentadas de forma progressiva com o objetivo de conferir com índice maior de imagens. Dessa maneira há o melhoramento da aprendizagem devido a variabilidade dos dados.

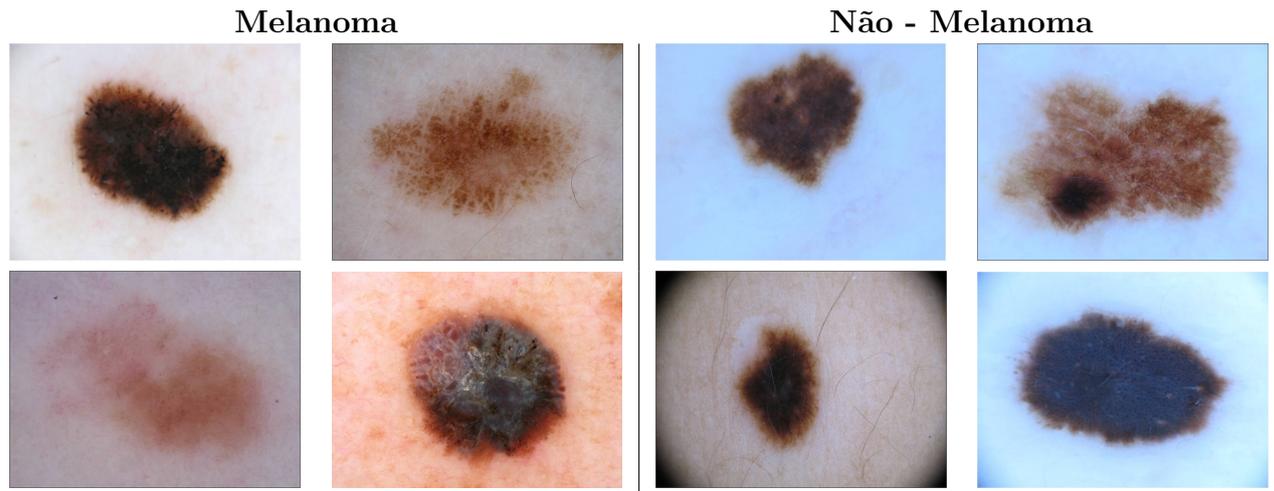


Figura 3.1 – Exemplos de lesões de pele do dataset ISIC (imagens de acesso aberto).

A metodologia de avaliação empregada foi a escolha de imagens de forma aleatória para gerar três pastas diferentes que são:

- Conjunto de treinamento.
- Conjunto de validação.
- Conjunto de teste

Esses conjuntos vão conter diversas quantidades de imagens as quais estão descritas nas tabelas a seguir.

Tabela 3.1 – Modelo A

Dataset	Rótulos das imagens	# de imagens
ISIC archive	1000 melanomas 1000 não-melanomas	Treinamento: 1400 Validação: 400 Teste: 200

Tabela 3.2 – Modelo B

Dataset	Rótulos das imagens	# de imagens
ISIC archive	2000 melanomas 2000 não-melanomas	Treinamento: 2400 Validação: 1000 Teste: 600

Tabela 3.3 – Modelo C

Dataset	Rótulos das imagens	# de imagens
ISIC archive	4522 melanomas 4522 não-melanomas	Treinamento: 8244 Validação: 600 Teste: 200

## 3.2 Aumento de dados

Usamos a ferramenta de aumento de dados de forma *online*, através de linha de código para cada pacote de imagens treinado. As transformações das imagens foram escolhidas do conjunto de opções disponíveis no Keras (CHOLLET *et al.*, 2015). A Tabela 3.4 ilustra as transformações escolhidas no processo de treinamento.

Tabela 3.4 – Transformações propostas

<b>Tipo de transformação</b>	<b>Valores possíveis para as amostras</b>
Ampliação	0 até 20%
Rotação	0 até 270
Mudanças verticais	Sim
Mudanças horizontais	Sim
Modo de preenchimento	Reflexão

É preciso prestar atenção quais tipos de transformações de imagens vamos escolher, já que há varias que podem atrapalhar a aprendizagem; isso acontece principalmente quando aplicamos rotação das imagens, pois é necessário escolher o preenchimento com os *pixels* vizinhos, já que o algoritmo padrão preenche com preto os cantos onde não se encontra a imagem e isso geraria resultados ruins. No nosso caso não é uma boa opção a ampliação excessiva da imagem, já que a lesão encontra-se na maioria das vezes no meio da imagem e se ampliamos muito há riscos da lesão sumir da imagem.

Treinamentos preliminares não vem mostrando bons resultados com as transformações testeadas. A opção de normalizar cada uma das imagens pelo seu desvio padrão e que cada amostra tenha média zero, com isso apresentam melhores resultados no desempenho do treinamento.

## 3.3 Métricas de avaliação

Por ser um problema de classificação binária, vamos avaliar o desempenho do modelo com a ajuda da métrica da *accuracy*, e o *loss*, as quais são utilizadas frequentemente para métodos supervisionados de classificação (HOSSIN; M.N, 2015). Essas métricas serão empregadas para avaliar o desempenho para os três conjuntos de dados, é dizer para o conjunto de treinamento, conjunto de validação, e o conjunto de teste.

Para determinar qual arquitetura teve uma melhor aprendizagem sob os dados e tem maior desempenho no momento da classificação do melanoma e do não-melanoma, vamos avaliar o conjunto de teste com a ajuda da matriz de confusão, a curva ROC ("*Receiver Operating Characteristic*"), e a área sob a curva AUC ("*Area Under the Curve*")

as duas explicadas na Seção 3.3.4, os quais vão permitir-nos conferir como a aprendizagem se comportará sob novos dados e assim escolher o modelo mais apropriado nesta tarefa.

### 3.3.1 Accuracy (exatidão)

A *accuracy* avalia a quantidade de imagens classificadas corretamente em cada uma das duas classes (melanoma e não-melanoma) baseando-se nos rótulos verdadeiros dividida pela quantidade total das imagens empregadas  $N$ , e é dada pela Equação 3.1 a seguir:

$$Acc = \frac{\sum_{i=1}^{n_i} \hat{y}_{c1}^+ + \sum_{i=1}^{n_i} \hat{y}_{c2}^+}{N}, \quad (3.1)$$

onde  $\hat{y}_{c1}^+$  representa as imagens classificadas corretamente na classe melanoma  $c1$  e  $\hat{y}_{c2}^+$  são as imagens corretamente classificadas para a classe não-melanoma  $c2$ .

### 3.3.2 Loss (erro)

Faz referência aos valores obtidos pela função de custo em cada um das rodadas dos dados por toda a arquitetura empregada. A *loss* será explicada com maior detalhe na Seção 3.8.

### 3.3.3 Matriz de confusão

É principalmente usado em problemas de classificação já que avalia ou contabiliza o número de classificações corretas conhecidas como verdadeiro positivo (VP), e as incorretas ou verdadeiro negativo (VN) em cada uma das classes existentes, as quais vão ser achadas na diagonal principal da matriz. Os outros componentes da matriz são conhecidos como falso positivo (FP), e são as imagens classificadas como classe positiva, mas que o rótulo verdadeiro é da classe negativa, e os falsos negativos (FN) que são imagens classificadas na classe negativa, mas seu rótulo pertence à classe positiva. A Tabela 3.5 ilustra a matriz de confusão anteriormente descrita.

Tabela 3.5 – Matriz de confusão para classificação binária.

<b>Classe Verdadeira</b>	<b>Melanoma</b>	Verdadeiro positivo (VP)	Falso negativo (FN)
	<b>Não-melanoma</b>	Falso Positivo (FP)	Verdadeiro negativo (VN)
		<b>Melanoma</b>	<b>Não-melanoma</b>
		<b>Classe estimada</b>	

(SOKOLOVA; LAPALME, 2009) tem mostrado que a informação apresentada na Tabela 3.5 permite conhecer quais classes apresentam maior dificuldade no momento de classificar. Além disso, a matriz de confusão permite computar diversas métricas de desempenho as quais são:

- **Precision (precisão)** corresponde as imagens classificadas corretamente da classe positiva, em relação a todas as imagens atribuídas na classe positiva e dada pela Equação (3.2).

$$Precision = \frac{VP}{VP + VN} \quad (3.2)$$

- **Recall (sensibilidade) TVP** faz referência ao número de imagens classificadas corretamente em relação ao número de imagens que pertencem à classe positiva, e é dado pela Equação (3.3). TVP também é conhecida como taxa de verdadeiros positivos, e mostra a efetividade do classificador para identificar os rótulos positivos. No nosso problema mostra quanto é eficiente a arquitetura na identificação dos melanomas.

$$Recall = \frac{VP}{VP + FN} \quad (3.3)$$

- **Specificity (especificidade) TFP** é dada pelo número de imagens classificadas corretamente em relação ao número de imagens que pertencem à classe negativa e é dada pela Equação (3.4). TFP também é conhecida como taxa de falso positivo e mostra a efetividade do classificador para identificar os rótulos negativos. No nosso problema mostra quanto é eficiente a arquitetura na identificação dos não-melanomas.

$$Specificity = \frac{VN}{VN + FP} \quad (3.4)$$

### 3.3.4 Curva ROC e a área sob a curva AUC

A análise dada pelo *Receiver Operator Characteristic* (ROC) é um dos mais empregados na comunidade médica para o desenvolvimento de decisões assistidas por computador (MAZUROWSKI *et al.*, 2008), já que descreve a relação entre dois indicadores que são: a taxa de verdadeiros positivos ou sensibilidade e a taxa de verdadeiros negativos ou especificidade.

A curva ROC é com frequência empregada para escolher os classificadores com desempenho melhor, e isso acontece quanto mais para esquerda e para a cima está a curva do classificador.

A área sob a curva AUC dada pela Equação (3.5) mostra uma medida geral do desempenho do classificador, onde  $AUC = 0,5$  dada pela diagonal (TVP = TFP) indica

um classificador aleatório e  $AUC = 1$  é o classificador ideal (VANHEULE; PIŻURICA, 2018).

$$AUC = \int_0^1 TVP \cdot TFP d(TFP) \quad (3.5)$$

## 3.4 Implementação

### 3.4.1 Linguagem de programação e framework

O *Python* foi empregado como linguagem de programação neste projeto. Todos os experimentos e arquiteturas que foram avaliadas e implementadas empregando o pacote Keras (CHOLLET *et al.*, 2015) como a biblioteca de *deep learning*. Além disso, para a visualização dos resultados foi empregado o pacote Matplotlib (HUNTER, 2007).

### 3.4.2 Experimentos e hardware

Todos os códigos, assim como a montagem das arquiteturas foram implementados no Kaggle, o qual é um ambiente de acesso livre para ser utilizado e que possui as características apresentadas na Tabela 3.6.

Tabela 3.6 – Especificações do ambiente Kaggle

Especificações	CPU	GPU
<b>Modelo</b>	INTEL Xeon	NVIDIA Tesla P100
<b>Gigabytes de RAM</b>	13	16
<b>Cores</b>	1 por socket até 2	4

Todos os experimentos foram desenvolvidos no modo GPU pois o tempo de processamento é muito menor ao treinamento em modo CPU (WANG *et al.*, 2019).

## 3.5 Resolução das imagens

Todas as imagens fornecidas e descarregadas do ISIC *dataset* são de tipo RGB com diversos tamanhos, pelo qual nós redimensionamos com o software *Resizer for Windows* para que todas as imagens possuam um tamanho fixo, assim que treinamos com imagens de tamanho  $200 \times 200$  *pixels* para o treinamento com a arquitetura AlexNet. Para as arquiteturas VGG16 e VGG19 as imagens foram redimensionadas para  $224 \times 224$  *pixels*, já que nessas arquiteturas utilizamos transferência de aprendizado e o tamanho do pré-treinamento é o tamanho de imagem mencionado.

Cabe destacar quanto maior o tamanho de imagens não adiantaria na eficácia do desempenho da arquitetura, apenas resultaria em um treinamento mais devagar.

## 3.6 Normalização

Segundo (SOLA; SEVILLA, 1997), a normalização é um passo importante no pré-processamento para o treinamento de redes neurais com a finalidade de obter bons resultados. A normalização é uma transformação linear sob os dados originais e é comumente usada a normalização *standard* (padrão) dada na Equação (3.6), a qual subtrai o valor meio  $\mu$  e logo após divide pelo desvio padrão da população  $\sigma$ .

$$z = \frac{x - \mu}{\sigma}, \quad (3.6)$$

Na dissertação empregamos dois tipos de normalizações as quais são:

- Normalização de cada uma das amostras divididas para o desvio padrão de todo o conjunto de treinamento.
- Normalização de cada uma das amostras divididas para desvio padrão de cada um delas.

## 3.7 Arquiteturas de *Deep Learning*

### 3.7.1 AlexNet

A arquitetura AlexNet (KRIZHEVSKY *et al.*, 2012), foi a escolhida para atingir a nossa proposta de classificação binária entre o melanoma e o não melanoma. Dita arquitetura está perfeitamente documentada e é comumente utilizada para problemas de classificação de imagens.

A Figura 3.2 apresenta o diagrama da arquitetura proposta, a qual consiste em cinco camadas convolucionais com diferentes tamanhos dos filtros ou *kernels*, as quais têm como função principal diminuir a dimensionalidade das imagens assim como extrair a maior informação possível da mesma.

O primeiro bloco representa a entrada do sistema denominada "*IN*", onde as imagens foram redimensionadas em  $200 \times 200$  *pixels* RGB com o objetivo de não sobrecarregar o processamento já que é necessário ter em conta a quantidade de pacotes de imagens que vão ser enviadas para o treinamento.

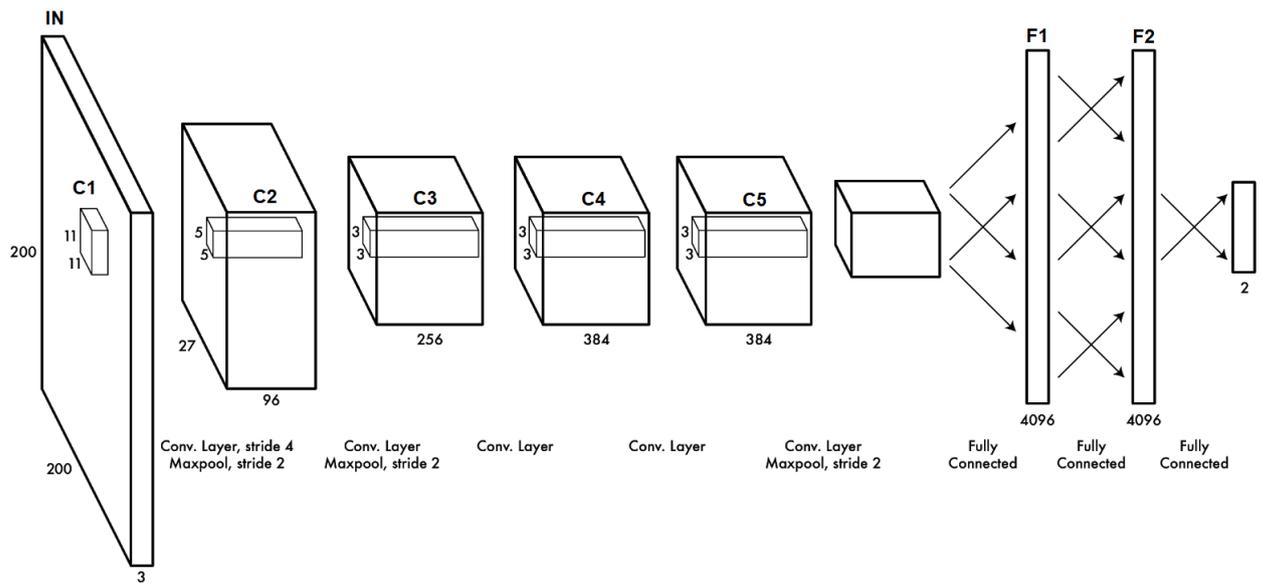


Figura 3.2 – Arquitetura AlexNet modificada para nosso problema de classificação.

Os blocos seguintes são as camadas convolucionais onde o tamanho dos *kernels* são: para a camada C1 o filtro tem um tamanho de  $11 \times 11$ , para a camada C2 o tamanho é de  $5 \times 5$ , e para as camadas C3, C4 e C5 os tamanhos dos filtros são de  $3 \times 3$ .

Em cada uma dessas camadas a função de ativação utilizada é a ReLU "*Rectified Linear Unit*". Depois de cada uma das camadas convolucionais foi empregado o conceito de *MaxPooling* (WU; GU, 2015) em todas as camadas para a redução da dimensionalidade na qual o tamanho dos filtros é de  $3 \times 3$ . As saídas de cada camada foram normalizadas com o *BatchNormalization* (BatchNorm) (IOFFE; SZEGEDY, 2015), a fim de estabilizar o processo de aprendizado e ter uma convergência mais rápida.

Os últimos três blocos pertencem as camadas densas, assim nas camadas F1 e F2 aplicamos *Dropout* (GÉRON, 2017), o qual permite ter a probabilidade de desligar certas quantidades de neurônios randomicamente. No entanto, ditos neurônios serão ativados na seguinte rodada com a finalidade de diminuir os problemas de *overfitting*.

No último bloco, tem-se dois neurônios, os quais tentam prever o rótulo verdadeiro da imagem, isto é, se a imagem pertence ou não a sua própria classe. Para nosso problema de classificação binária a melhor função de ativação é a *sigmoid*, pois é a mais utilizada para prever valores entre 0 e 1.

### 3.7.2 VGG16

A arquitetura VGG16 (do inglês "*Visual Geometry Group*") (SIMONYAN; ZISSERMAN, 2014) é aplicável para o reconhecimento de imagens de diversos banco de dados.

Neste projeto foi escolhido para ser empregado como classificador binário. VGG16 foi implementada com a ajuda da transferência de aprendizado ou *transfer-learning* onde dita arquitetura primeiramente sendo pre-treinada com um grande banco de dados da ImageNet (DENG *et al.*, 2009) e portanto as características da VGG16 podem ser aproveitadas e adaptadas para este problema.

Devido a inexistência de um banco de dados extenso, a portabilidade do recurso aprendido que permite-nos ter o *deep learning* torna a VGG16 mais eficazes nestes tipos de problemas. Nesta arquitetura congelamos as camadas convolucionais e empregamos a aprendizagem deles em o nosso banco de dados, já as camadas totalmente conectadas ou *fully-connected* foram treinadas com nossos dados. A Figura 3.3 mostra os componentes de cada camada, as quantidades de filtros empregados, as funções de ativação, e as quantidades de neurônios das camadas totalmente conectadas.

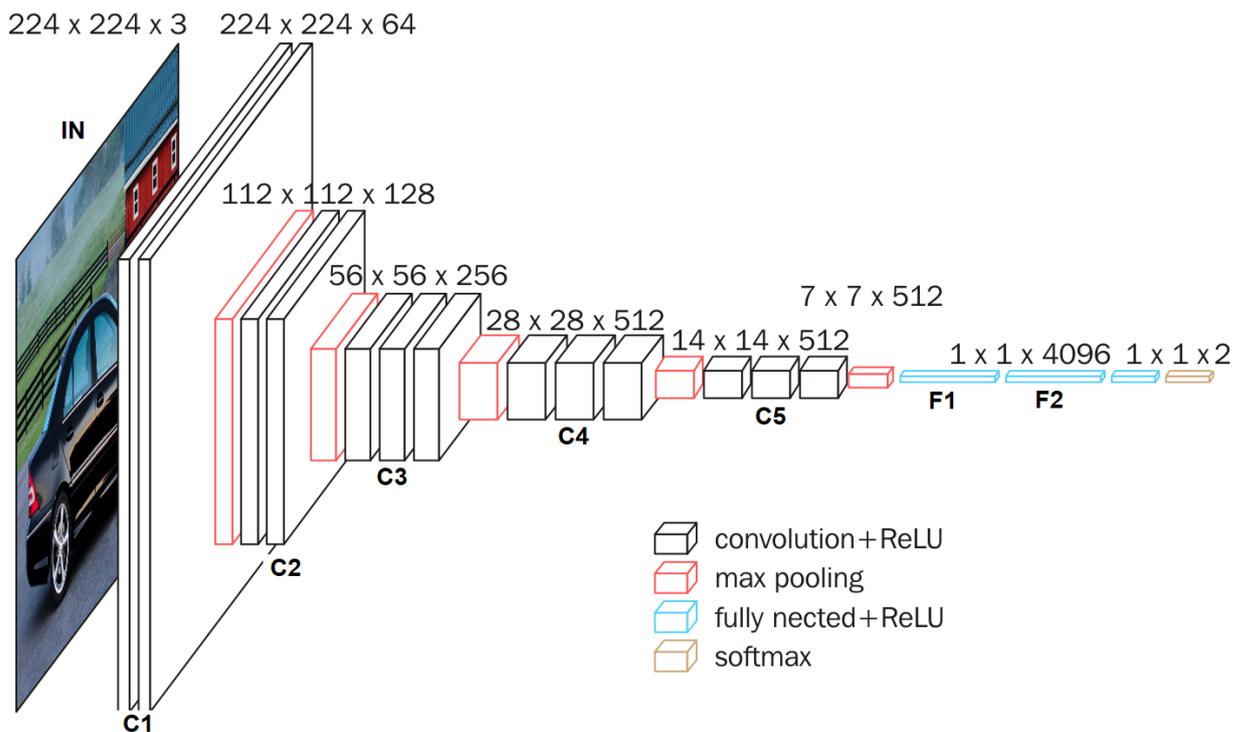


Figura 3.3 – Arquitetura VGG16 modificada para nosso problema de classificação.

A VGG16 baseia-se no conceito de melhorar a arquitetura original da Alex-Net, onde um dos pontos principais foi fixar alguns parâmetros da arquitetura e aumentar continuamente a profundidade do mesmo, adicionando mais camadas convolucionais e diminuir o tamanho dos *kernels* fixando para uma matriz de  $3 \times 3$  em cada camada. O primeiro bloco é a entrada do sistema denominada "IN", onde as imagens foram redimensionadas em  $224 \times 224$  pixels RGB com o objetivo de não sobrecarregar o processamento e tornar o treinamento mais fiável e rápido.

Os blocos restantes são as camadas convolucionais onde o tamanho dos *kernels*

são: para a camada C1 dois pacotes de 64 filtros cada um do tamanho  $3 \times 3$ , para a camada C2 dois pacotes de 128 filtros cada um, na camada C3 três pacotes de 256 filtros e para as camadas C4 e C5 três pacotes de 512 filtros.

Em cada uma dessas camadas a função de ativação utilizada é a ReLU "*Rectified Linear Unit*". Depois de cada uma das camadas convolucionais foi empregado o conceito de *MaxPooling* (WU; GU, 2015) para a redução da dimensionalidade, onde o tamanho dos filtros é de  $2 \times 2$ .

Os últimos três blocos pertencem as camadas densas as quais foram treinadas anteriormente com o banco de dados do melanoma e não-melanoma. Nas camadas F1 e F2 aplicamos *Dropout* (GÉRON, 2017), o qual permite ter a probabilidade de desligar certas quantidades de neurônios randomicamente, no entanto, ditos neurônios serão ativados na seguinte rodada com a finalidade de diminuir os problemas de *overfitting*.

No último bloco, temos dois neurônios, os quais tentam prever o rótulo verdadeiro da imagem, isto é, se a imagem pertence ou não a sua própria classe. Para o nosso problema de classificação binária a melhor função de ativação é a *sigmoid*, pois é a mais utilizada para prever valores entre 0 e 1.

### 3.7.3 VGG19

A arquitetura VGG19, é uma melhora da VGG16, já que aumenta-se a profundidade da aprendizagem. A Figura 3.4 mostra as características da cada uma das camadas tanto das convolucionais como das densas da VGG19.

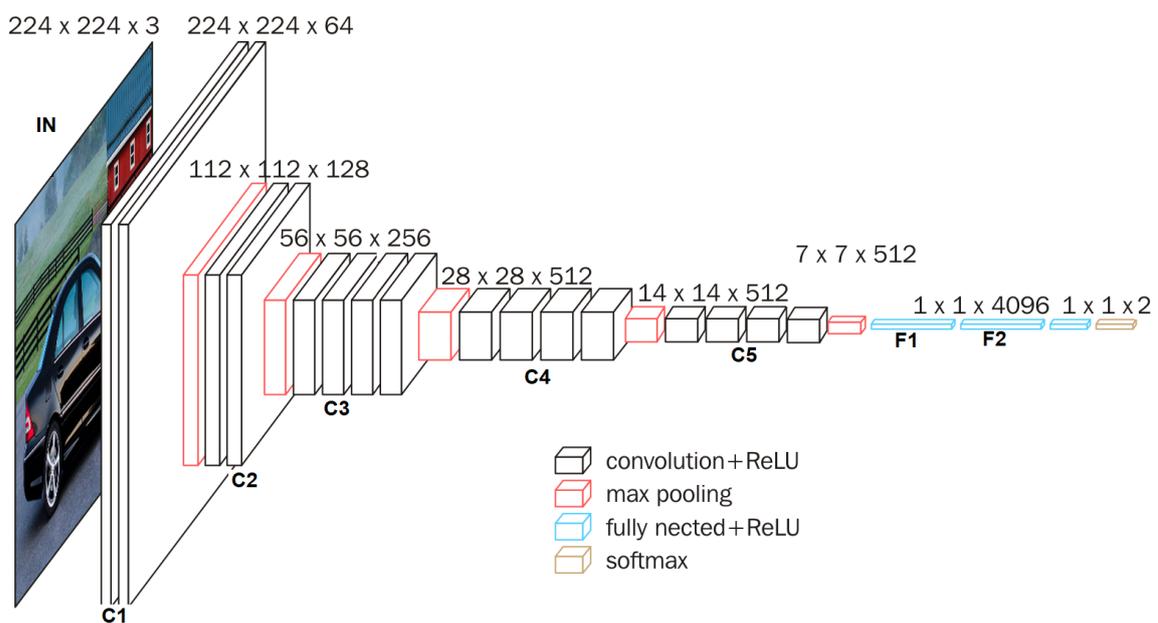


Figura 3.4 – Arquitetura VGG19 modificada para nosso problema de classificação.

O primeiro bloco da VGG19 é o de entrada "IN", os blocos convolucionais C1 e C2 são similares aos da arquitetura VGG16. Para a camada convolucional C3 aumenta-se a profundidade ao adicionar um pacote a mais de filtros tendo assim, quatro pacotes de 256 filtros. O mesmo acontece com as camadas C4 e C5, onde temos quatro pacotes de 512 filtros. As camadas densas são as mesmas da VGG16 mantendo dois neurônios na última camada a fim de atingir nosso problema de classificação binária.

### 3.8 Função custo

A função de custo que na literatura representa a entropia cruzada binária dada pela Equação (3.7), e na qual é usada com modelos que produzem valores de probabilidades na saída, e será usada no nosso problema de classificação binária. O uso da entropia cruzada é uma métrica empregada para obter a distância entre as distribuições de probabilidade, em nosso caso, ela tem como intuito medir a qualificação de um conjunto de probabilidades da classe estimada corresponde as classes verdadeiras.

$$J(\Theta) = - \sum_{i=1}^C y_i \log(\hat{y}_i). \quad (3.7)$$

Da Equação (3.7) temos que  $y_i$  é o valor verdadeiro da classe da imagem e  $\hat{y}_i$  é a estimativa de pertencer ou não aquela classe, para cada classe  $i$  em  $C$ .

No caso de classificação binária,  $C = 2$  e podemos definir a entropia cruzada como a dada pela Equação (3.8).

$$J(\Theta) = - \sum_{i=1}^{C=2} y_i \log(\hat{y}_i) = -y_1 \log(\hat{y}_1) - (1 - y_1) \log(1 - \hat{y}_1). \quad (3.8)$$

Da Equação 3.8 temos duas classes  $C_1$  e  $C_2$  para os valores verdadeiros de  $y_1[0, 1]$  e  $\hat{y}_1$  representa as estimativas da  $C_1$ . Para a classe  $C_2$  os valores verdadeiros estão dados por  $y_2 = 1 - y_1$  e as estimativas por  $\hat{y}_2 = 1 - \hat{y}_1$ .

### 3.9 Otimizadores

Avaliamos o desempenho das três arquiteturas propostas com o uso de três tipos de otimizadores os quais são o Adam, o RMSprop (KINGMA; BA, 2014) e o gradiente estocástico descendente (SGD) (RUDER, 2016). Todos eles são inicializados com uma taxa de aprendizagem de 0,001. Por outro lado, para o otimizador SGD foi adicionando o momentum de 0,9 que tem o intuito de acelerar o processo para conseguir sair dos mínimos locais onde não aprendia mais do conjunto de treinamento.

### 3.10 Parada antecipada (*EarlyStopping*)

Depois de efetuar várias provas deixando as arquiteturas serem treinadas até atingir os valores mais eficientes que elas conseguiram, notamos que com o passar das épocas elas iam tendendo ao sobre ajuste, devido a que o banco de dados não possui uma grande quantidade de imagens, sendo assim que nós empregamos a parada antecipada a qual permite parar o treinamento.

A parada antecipada foi configurada para avaliar o desempenho da *accuracy* no conjunto de validação e quando a *accuracy* não melhora no seu último valor atingido passado 10 épocas, o programa ira interromper e salvar automaticamente o melhor resultado obtido na aprendizagem (CHOLLET *et al.*, 2015).

## 4 Resultados e Discussões

Neste trabalho, nós focamos nas arquiteturas AlexNet, VGG16, e VGG19, o principal motivo de empregar as arquiteturas antes descritas é que pequenas redes são tipicamente mais rápidas para convergir no treinamento e são menos prováveis de produzir sobre-ajuste ou *overfitting* uma vez que o conjunto de dados não é muito longo.

Neste capítulo vamos apresentar os resultados obtidos com diferentes quantidades de imagens em cada um dos três conjuntos de dados (treinamento, validação e teste). No Modelo C o qual tem a maior quantidade de imagens para serem treinadas, nós avaliamos as arquiteturas VGG16 e VGG19 com ajuda da técnica transferência de aprendizado ou *transfer-learning*. Finalmente, avaliamos a aprendizagem com a ajuda da matriz de confusão, a qual permite obter maiores informações sobre o comportamento da arquitetura.

### 4.1 Resultados empregando a arquitetura AlexNet

#### 4.1.1 Resultados do treinamento com o Modelo A

A Figura 4.1a mostra o desempenho das *accuracys* dos três otimizadores apresentados na Seção 3.9, onde podemos observar que o otimizador SGD tem um melhor desempenho no processo de treinamento atingindo 97,66% de *accuracy*. Por outro lado, o otimizador Adam (curva de cor vermelho) é aquele que mostra uma aprendizagem mais rápida, já que sua curva começa com valores altos de 98,00% na *accuracy*.

Na Figura 4.1b o *loss* no processo de treinamento é apresentado, onde o otimizador SGD possui o menor *loss* em comparação aos dois otimizadores. Assim, ele atinge um *loss* de 0,0567. Por outro lado, o otimizador RMSprop mostra um decaimento mais pronunciado nas primeiras épocas.

#### 4.1.2 Resultados do treinamento com o Modelo B

Neste segundo modelo, incrementamos a quantidade de imagens para os dois conjuntos de dados (treinamento e validação). Na Figura 4.2a podemos observar como existe uma melhora no desempenho em termos da *accuracy* na classificação já que o modelo consegue atingir 99,79% no treinamento tendo uma melhora de 2,13% na aprendizagem em comparação ao Modelo A.

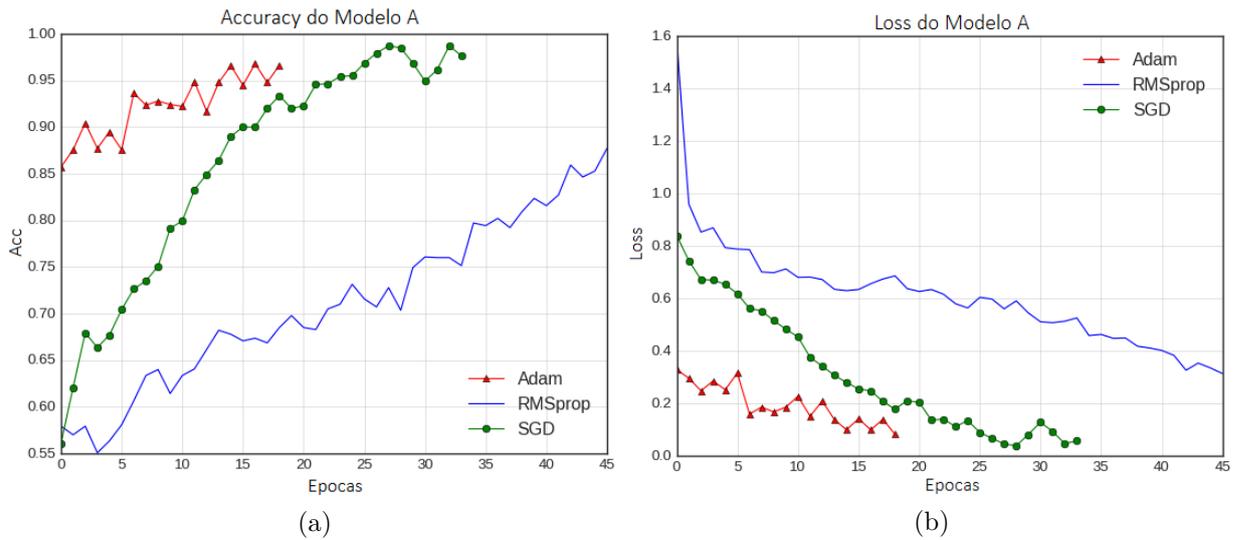


Figura 4.1 – Desempenho do treinamento no Modelo A (a) *Accuracy*, e (b) *Loss*.

Na Figura 4.2b visualizamos que o *loss* tem uma convergência rápida no otimizador SGD, assim o *loss* aproxima-se a zero.

Dado aos valores altos atingidos tanto pelo Modelo A quanto pelo Modelo B no treinamento, é possível que o modelo não esteja generalizando os dados e consequentemente esteja memorizando-os. Por tal motivo é necessário conferir que o desempenho da *accuracy* do treinamento seja quase semelhante à *accuracy* da validação. Além disso, é necessário avaliar os valores do *loss* nos dois conjuntos de dados (treinamento e validação), os quais vão mostrar se o modelo não está tendendo para um *overfitting*.

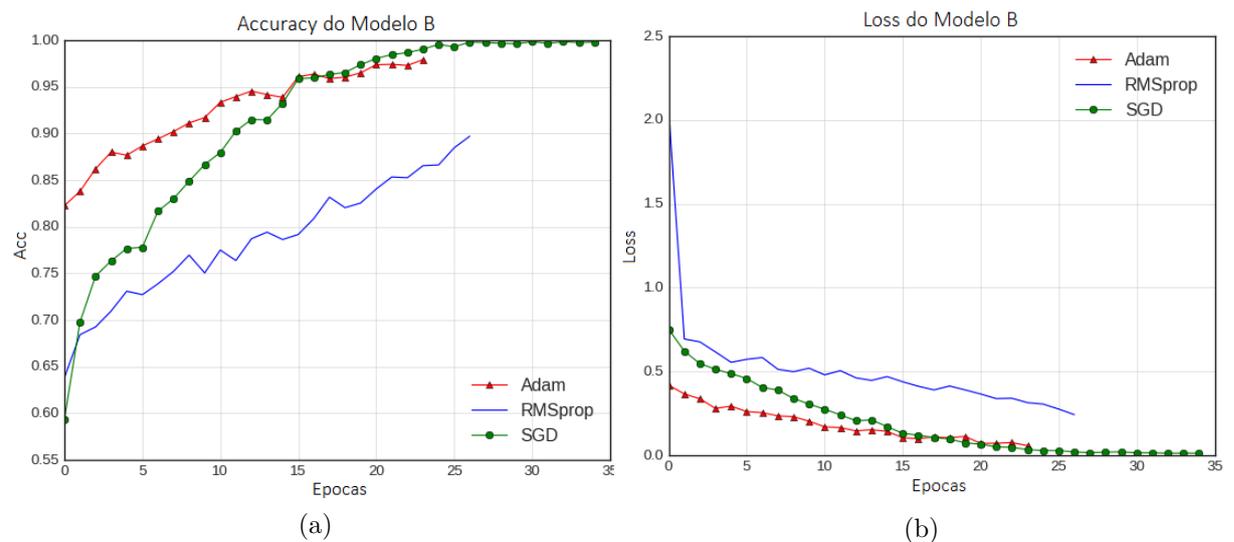


Figura 4.2 – Desempenho do treinamento no Modelo B (a) *Accuracy*, e (b) *Loss*.

Depois de avaliar o desempenho dos três otimizadores com diferentes quantidades de imagens, notamos que o SGD é aquele que atinge o melhor desempenho em

termos de *accuracy* nos três conjuntos de dados, tanto no modelo A, quanto no modelo B, o qual podemos visualizar na Tabela 4.1. Ao avaliar os resultados obtidos na validação, devemos prestar atenção quando o *loss* crescer muito, assim, como a possível separação das curvas da *accuracy* no treinamento e na validação, já que com isso podemos perceber quando o treinamento caiu no sobre-ajuste ou *overfitting*.

Tabela 4.1 – Comparações dos Modelos A e B.

		Modelo A			Modelo B		
		Adam	RMSprop	SGD	Adam	RMSprop	SGD
<b>Treinamento</b>	<i>Acc (%)</i>	96,66	87,64	<b>97,66</b>	97,85	89,72	<b>99,79</b>
	<i>Loss</i>	0,0817	0,3141	<b>0,0567</b>	0,0591	0,2434	<b>0,0120</b>
<b>Validação</b>	<i>Acc (%)</i>	75,50	70,25	<b>78,00</b>	77,70	79,60	<b>81,50</b>
	<i>Loss</i>	0,8227	0,7583	<b>0,4360</b>	0,8047	0,5528	<b>0,4098</b>

A Tabela 4.2 mostra os resultados obtidos no conjunto de teste com o otimizador SGD. Notamos que os valores de *accuracy* melhoram conforme aumentamos a quantidade de imagens já que conseguimos acrescentar a variabilidade de dados para que o nosso modelo aprenda mais ao respeito do conjunto de dados que se deseja classificar. O melhor resultado no conjunto de teste foi obtido pelo treinamento com o Modelo B. Assim, nós obtemos uma *accuracy* de 79,19%, o qual representa uma melhoria de 8,67% em comparação com o treinamento do Modelo A.

Tabela 4.2 – Avaliação do otimizador SGD sobre o conjunto de teste

		Modelo A	Modelo B
<b>TEST</b>	<i>Acc (%)</i>	70,50	79,17
	<i>Loss</i>	0,2950	0,2083

## 4.2 Resultados do treinamento com o Modelo C

Neste último modelo de treinamento proposto, foram feitos em torno de dez provas com cada um dos três otimizadores (Adam, RMSprop, e SGD), das quais foram escolhidas os cinco melhores resultados para serem comparados. Neste último modelo também foi feito o treinamento com as arquiteturas AlexNet, VGG16 e VGG19. As duas últimas arquiteturas foram implementadas por meio da transferência de aprendizado ou *transfer-learning*.

### 4.2.1 Treinamento com a arquitetura Alexnet

A Tabela 4.3 mostra os resultados obtidos tanto da *accuracy* como do *loss* para o conjunto de dados no processo de validação e do teste. Além disso, em cada otimizador mostra-se de cor mais escura os treinamentos que atingiram os melhores resultados. Podemos perceber que o modelo de treinamento com maior quantidade de imagens melhorou notavelmente seu desempenho tanto em termos de *accuracy* quanto do *loss*.

Tabela 4.3 – Avaliação do treinamento com o modelo C e a arquitetura AlexNet.

		AlexNet					
		Adam		RMSprop		SGD	
		Validação	Teste	Validação	Teste	Validação	Teste
1	Acc (%)	<b>93,17</b>	<b>92,50</b>	92,00	91,00	<b>93,00</b>	<b>92,33</b>
	Loss	<b>0,0683</b>	<b>0,0750</b>	0,0800	0,0900	<b>0,0700</b>	<b>0,0767</b>
2	Acc (%)	91,67	91,00	90,83	87,50	93,50	91,67
	Loss	0,0833	0,0900	0,0917	0,1250	0,0650	0,0833
3	Acc (%)	90,33	89,00	91,17	91,00	91,83	91,50
	Loss	0,0967	0,1100	0,0883	0,0900	0,0817	0,0850
4	Acc (%)	92,17	89,50	91,33	89,00	92,50	92,00
	Loss	0,0783	0,1050	0,0867	0,1100	0,0750	0,0800
5	Acc (%)	91,33	91,00	<b>92,50</b>	<b>91,50</b>	93,17	91,50
	Loss	0,0867	0,0900	<b>0,0750</b>	<b>0,0850</b>	0,0683	0,0850

A Figura 4.3 apresenta os melhores valores atingidos pelos otimizadores empregando a arquitetura AlexNet, dos quais o que atingiu o melhor desempenho foi o otimizador Adam no referente ao valor da *accuracy* obtida no conjunto de teste, já que nesta etapa conseguimos verificar como nossa arquitetura aprendeu sobre o conjunto de dados.

Devido à variação dos valores que foram avaliados em cada uma das rodadas empregando a arquitetura AlexNet, notamos que tanto as curvas da *accuracy* quanto do *loss* mostram-se ruidosas, por tanto elas foram suavizadas por motivos de visibilidade.

Na Figura 4.4a as curvas da *accuracy* tanto no treinamento como na validação surgem sem muita variação, onde podemos observar que a curva na validação acompanha a curva do treinamento até estabilizar-se entorno da época 20 onde as duas se separam, por tal motivo o parâmetro de parada antecipada para o treinamento quando ele atingiu o seu valor máximo de 0,9310.

Na Figura 4.4b observamos que o *loss* no treinamento tem um decaimento com o aumento das épocas. Por outro lado, a curva da validação não mostra o mesmo decaimento. No entanto, nós consideramos e salvamos o melhor valor atingido o qual é em torno da época 25 já depois da rodada 25 a aprendizagem começa a apresentar *overfitting*

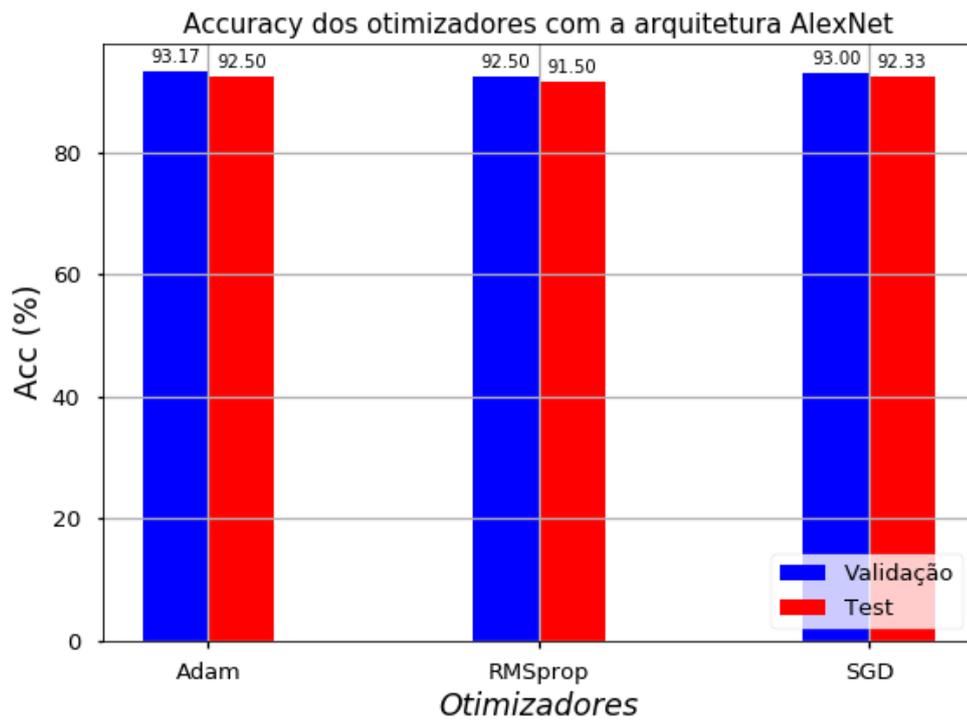


Figura 4.3 – Comparação da *accuracy* dos três otimizadores no conjunto de validação e teste do Modelo C.

ou sobre ajuste, em outras palavras já não existe aprendizado.

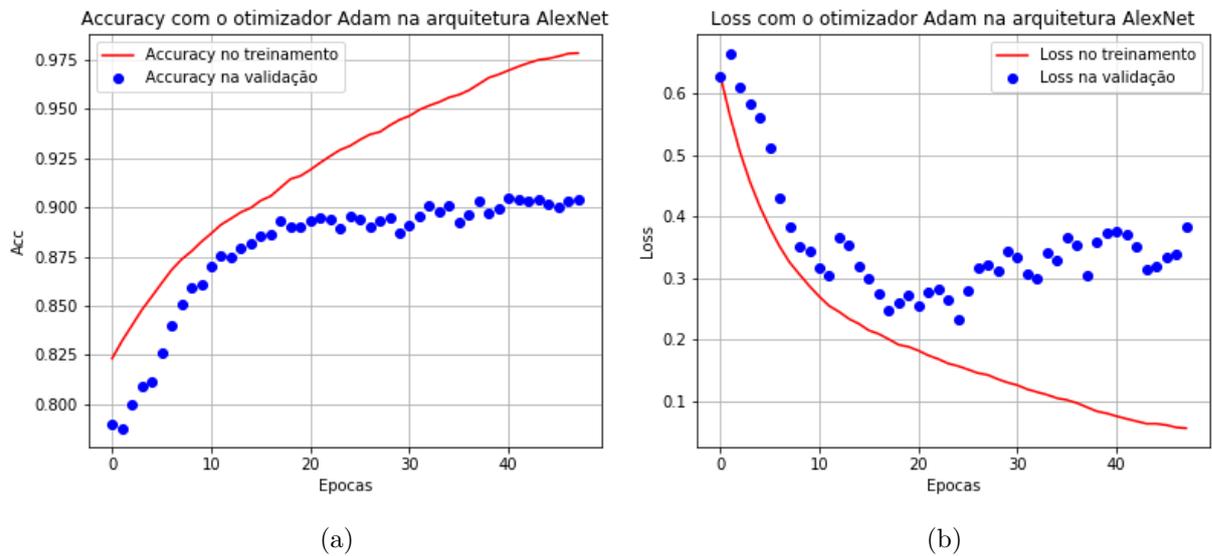


Figura 4.4 – Desempenho do treinamento no Modelo C com o otimizador Adam na arquitetura AlexNet (a) *Accuracy*, e (b) *Loss*.

## 4.2.2 Treinamento com a arquitetura VGG16

Neste treinamento usamos as arquiteturas pré-treinadas com o objetivo de aproveitar os pesos aprendidos e obter melhores valores na extração de características para nossa classificação. A Tabela 4.4 mostra os valores obtidos no treinamento usando a arquitetura VGG16, no qual podemos notar que os valores que aparecem em negrito foram aqueles treinamentos, onde cada otimizador atingiu seu melhor desempenho tanto no conjunto de validação quanto no conjunto de teste.

Tabela 4.4 – Avaliação do treinamento com o modelo C e a arquitetura VGG16.

		VGG16					
		Adam		RMSprop		SGD	
		Validação	Teste	Validação	Teste	Validação	Teste
1	Acc (%)	92,50	90,50	<b>92,50</b>	<b>91,00</b>	<b>92,00</b>	<b>89,33</b>
	Loss	0,0750	0,0950	<b>0,0750</b>	<b>0,0900</b>	<b>0,0800</b>	<b>0,1067</b>
2	Acc (%)	<b>93,50</b>	<b>90,83</b>	94,00	90,67	89,50	88,83
	Loss	<b>0,0650</b>	<b>0,0917</b>	0,0600	0,0933	0,1050	0,1117
3	Acc (%)	93,50	90,50	93,00	90,50	90,00	88,83
	Loss	0,0650	0,0950	0,0700	0,0950	0,1000	0,1117
4	Acc (%)	93,50	90,33	94,00	90,83	90,50	88,83
	Loss	0,0650	0,0967	0,0600	0,0917	0,0950	0,1117
5	Acc (%)	93,50	90,50	92,50	90,83	91,00	88,50
	Loss	0,0650	0,0950	0,0750	0,0917	0,0900	0,1150

Dos treinamentos feitos, podemos concluir que não existe muita variação entre os valores encontrados em cada uma das rodadas dos otimizadores. Os valores obtidos no teste permitem conferir se a arquitetura esta generalizando os dados.

A Figura 4.5, compara os melhores desempenhos obtidos pela arquitetura treinada, onde notamos que o otimizador Adam e o RMSprop atingiram bom desempenho na validação, mas no teste o RMSprop foi aquele que apresentou um melhor *accuracy* na classificação.

Note-se que as curvas da *accuracy* e do *loss* foram suavizadas para se tornar mais visíveis na comparação. Na Figura 4.6a observamos que a curva do treinamento mostra um crescimento com o aumento das épocas, porém a curva da validação no começo do treinamento apresentou valores mais altos com relação aos valores do treinamento dado que eles estavam se adaptando, pois os pesos dos filtros foram treinados com a base de dados da Imagenet (DENG *et al.*, 2009). A aprendizagem na validação vai aumentando desde os 87,00% até atingir seu valor máximo de 92,50%.

O *loss* da validação com a arquitetura VGG16 não apresenta um decaimento como observamos na Figura 4.6b, isso acontece devido a que o conjunto de dados da

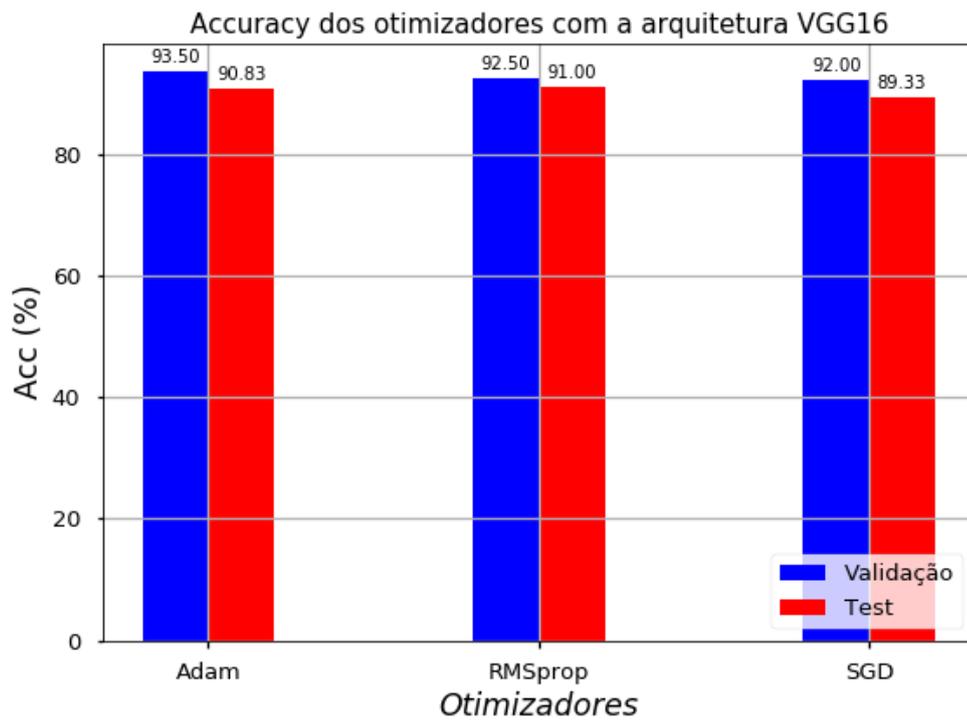


Figura 4.5 – Comparativa da *accuracy* dos três otimizadores no conjunto de validação e teste do Modelo C.

validação não contém muita informação para avaliar a aprendizagem. Por outro lado, observamos que na época 25 ele tende para um *overfittig*. Porém, a curva do *loss* no treinamento apresenta um decaimento.

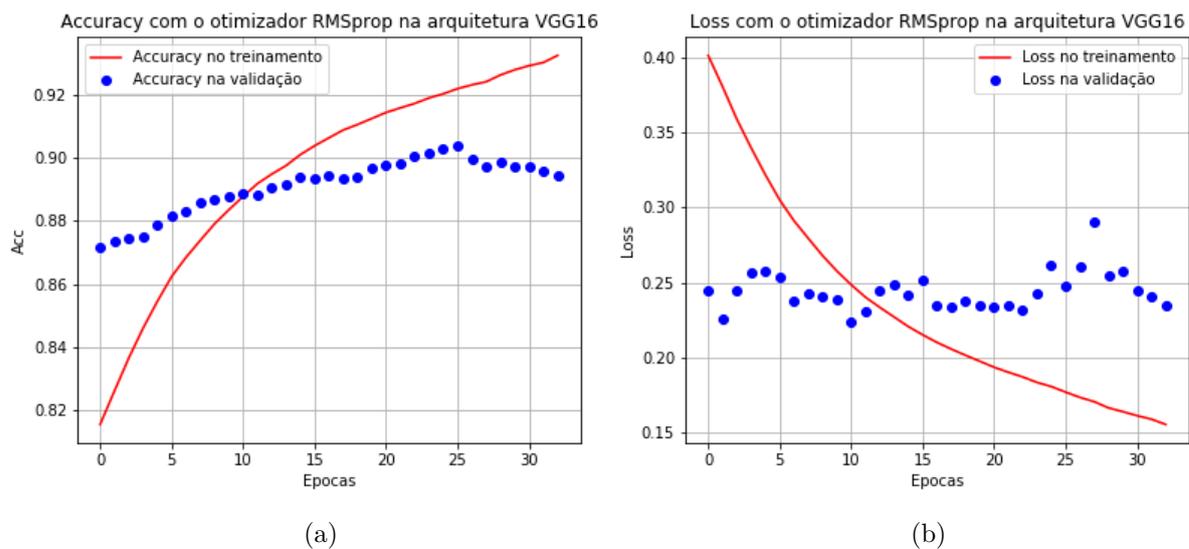


Figura 4.6 – Desempenho do treinamento no Modelo C com o otimizador RMSprop na arquitetura VGG16 (a) *Accuracy*, e (b) *Loss*.

### 4.2.3 Treinamento com a arquitetura VGG19

Neste último treinamento, usamos a arquitetura pre-treinada VGG19. Na Tabela 4.5 mostramos os valores obtidos no treinamento sob os conjuntos de validação e teste, dos quais podemos observar que os valores que aparecem em negrita foram aqueles que atingiram o melhor desempenho tanto no conjunto de validação como no teste.

Tabela 4.5 – Avaliação do treinamento com o modelo C e a arquitetura VGG19.

		VGG19					
		Adam		RMSprop		SGD	
		Validação	Teste	Validação	Teste	Validação	Teste
1	Acc (%)	91,33	91,00	<b>93,00</b>	<b>91,17</b>	87,50	87,00
	Loss	0,0867	0,0900	<b>0,0700</b>	<b>0,0883</b>	0,1250	0,1300
2	Acc (%)	<b>94,00</b>	<b>91,00</b>	91,83	91,00	<b>88,00</b>	<b>87,00</b>
	Loss	<b>0,0600</b>	<b>0,0900</b>	0,0817	0,0900	<b>0,1200</b>	<b>0,1300</b>
3	Acc (%)	92,50	91,00	92,00	88,50	88,50	86,50
	Loss	0,0750	0,0900	0,0800	0,1150	0,1150	0,1350
4	Acc (%)	91,67	91,00	91,50	90,50	88,50	85,00
	Loss	0,0833	0,0900	0,0850	0,0950	0,1150	0,1350
5	Acc (%)	93,50	90,83	91,17	88,00	88,83	84,00
	Loss	0,0650	0,0917	0,0883	0,1200	0,1117	0,1600

Na Figura 4.7 podemos visualizar as curvas da *accuracy* e do *loss* para o treinamento com a arquitetura VGG19. Na Figura 4.7a observamos que a curva dos valores obtidos no conjunto de validação convergem à curva do treinamento na época 11 e ai os valores são quase similares, assim mesmo podemos visualizar que os dados da validação comecem com valores altos a diferença dos dados do treinamento, isso é dado pelo uso da transferência do aprendizado.

Na Figura 4.7b, as duas curvas mostram um decaimento contínuo tanto na validação como no treinamento, porém a aprendizagem em torno da época 22 foi finalizado devido a que a partir daqui observamos um aumento do *loss*.

### 4.2.4 Comparativa das arquiteturas com o treinamento do Modelo C

Uma vez que o Modelo C tem uma maior quantidade de imagens, vamos ter melhores resultados de classificação do nosso conjunto de dados, já que no processo percebemos como a *accuracy* vai aumentando quando há maior variabilidade dos dados e como é natural o *loss* vai diminuir até atingir os melhores valores de aprendizagem.

A Figura 4.8 mostra os melhores valores que foram encontrados no processo de aprendizagem do conjunto de dados do Modelo C, onde podemos visualizar que a arquitetura AlexNet conjuntamente com o otimizador Adam atingiram as *acurracys* mais

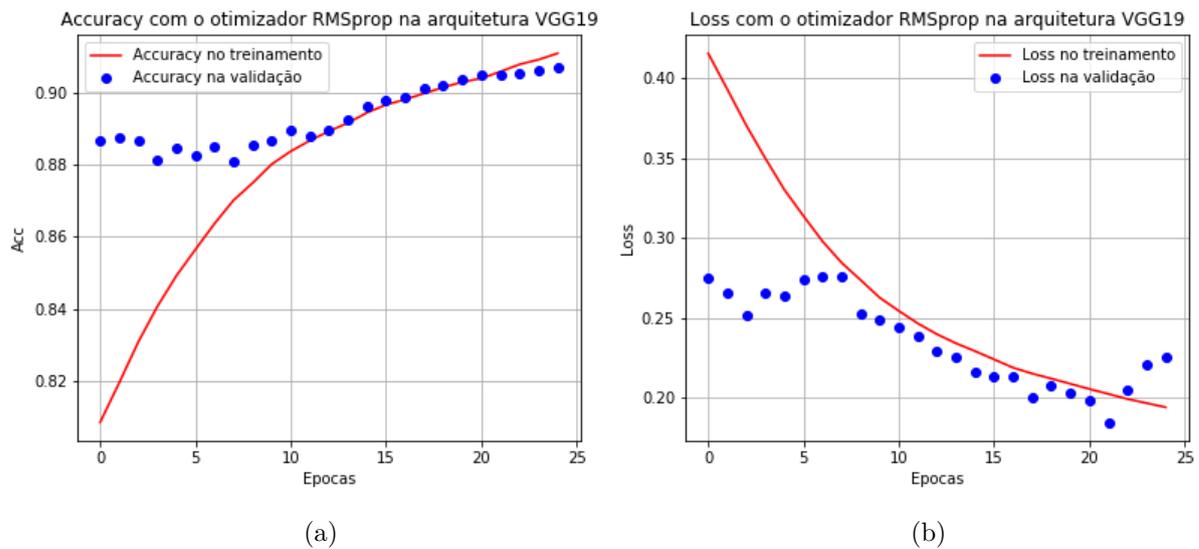


Figura 4.7 – Desempenho do treinamento no Modelo C com o otimizador RMSprop na arquitetura VGG19 (a) *Accuracy*, e (b) *Loss*.

altas em comparação as arquiteturas VGG16 e VGG19 tem uma *accuracy* de 93,17% para o conjunto de validação e 92,50% para o conjunto de teste.

Em referência as outras arquiteturas, elas tem uma melhora de 1,50% para a arquitetura VGG16 com o otimizador RMSprop e de 1,33% o VGG19 com o otimizador RMSprop.

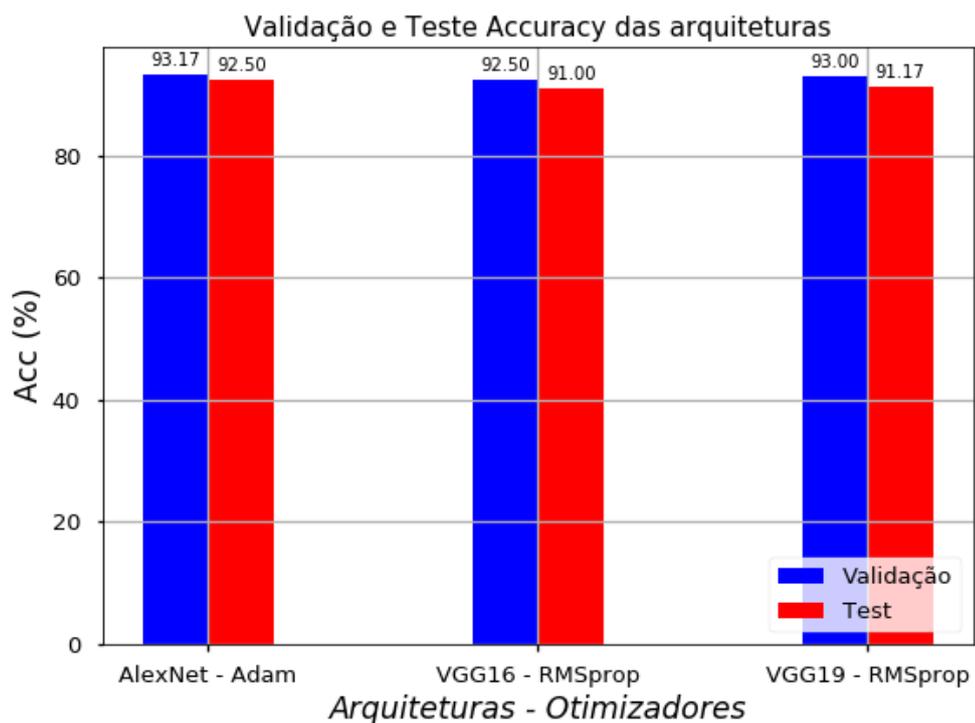


Figura 4.8 – Comparativa da *accuracy* das três arquiteturas no conjunto de validação e teste do Modelo C.

Depois de avaliar os valores de *accuracy* na aprendizagem é necessário avaliar através da matriz de confusão, pois em problemas de imagens médicas é preciso obter bons resultados tanto na precisão como na sensibilidade do algoritmo o qual é possível ao usar os valores da matriz de confusão o qual foi apresentado na Seção 3.3.3.

#### 4.2.4.1 Matriz de confusão com a arquitetura AlexNet

No momento de avaliar a aprendizagem feito com a arquitetura AlexNet sobre o conjunto de imagens para o teste, onde ela contém 100 imagens da classe melanoma e 100 imagens da classe não-melanoma, os resultados podem ser visualizados na Figura 4.9.

A diagonal principal da matriz apresenta as imagens que foram classificadas acertadamente das 100 da classe melanoma, onde 91 foram estimadas corretamente e 9 estão erradas. Por outro lado, para a classe não-melanoma 96 imagens foram estimadas acertadamente e só 4 erroneamente.

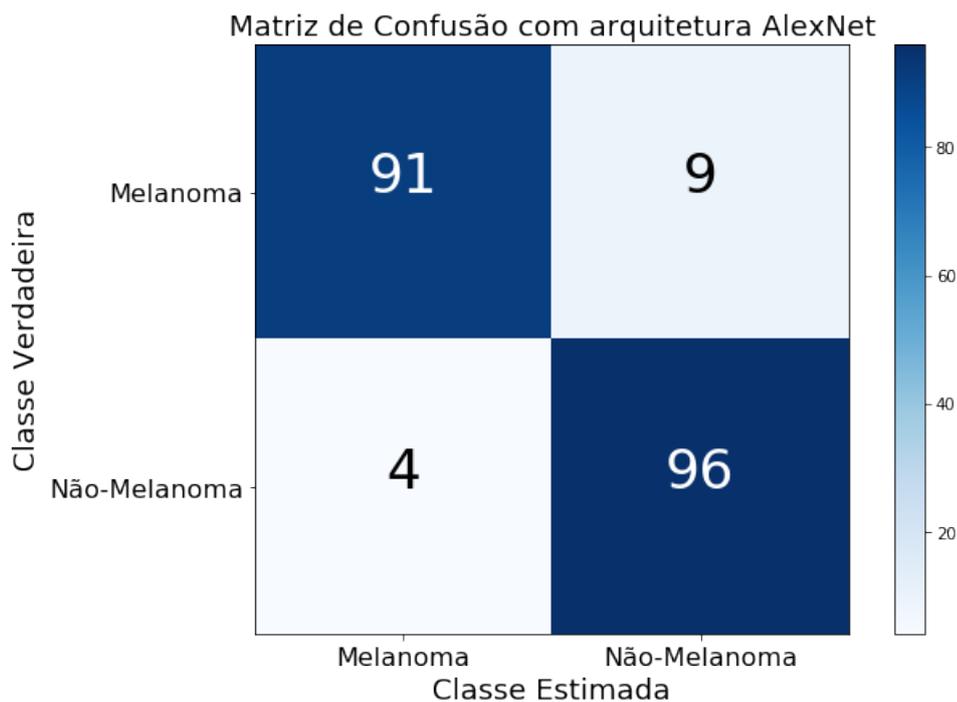


Figura 4.9 – Matriz de confusão da arquitetura AlexNet e o otimizador Adam do Modelo C.

Das equações mostradas na Seção 3.3.3, obtivemos os seguintes resultados:

- **Precisão** =  $0,9578 = 95,78\%$
- **Sensibilidade** =  $0,9100 = 91,00\%$
- **Especificidade** =  $0,9600 = 96,00\%$

#### 4.2.4.2 Matriz de confusão com a arquitetura VGG16

Na arquitetura VGG16 pré-treinada com as imagens da Imagenet, podemos apreciar na Figura 4.10, houve a aquisição de bons resultados de classificação mostrados na diagonal principal da matriz.

Fazendo referência aos resultados obtidos anteriormente esta arquitetura erro em torno a uma imagem a mais para cada classe.

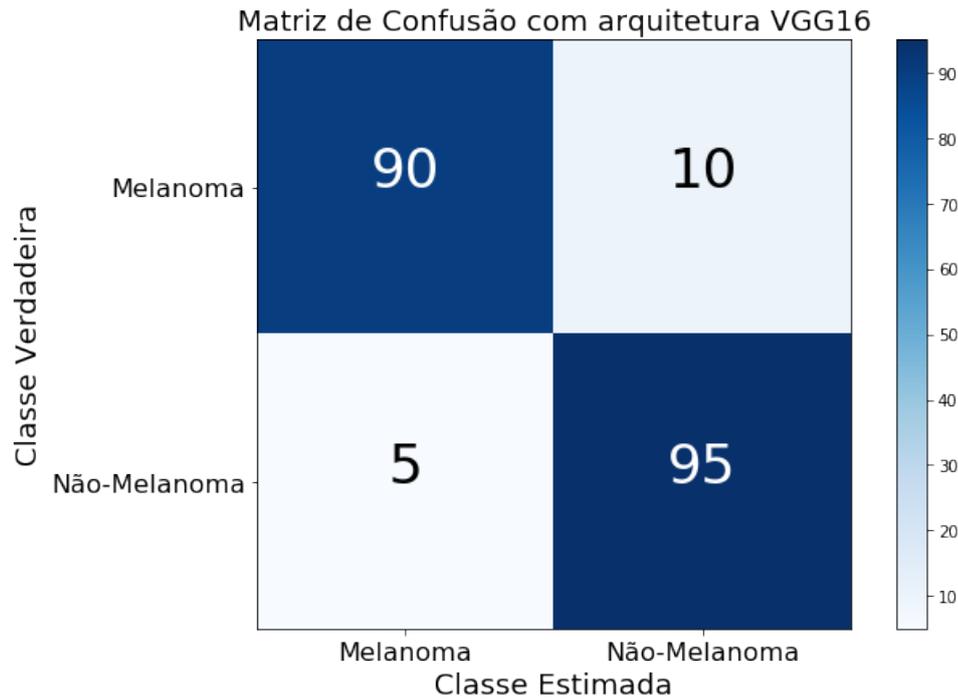


Figura 4.10 – Matriz de confusão da arquitetura VGG16 e o otimizador RMSprop do Modelo C.

Das equações mostradas na Seção 3.3.3, obtivemos os seguintes resultados:

- **Precisão** =  $0,9474 = 94,74\%$
- **Sensibilidade** =  $0,9000 = 90,00\%$
- **Especificidade** =  $0,9500 = 95,00\%$

#### 4.2.4.3 Matriz de confusão com a arquitetura VGG19

Finalmente, na arquitetura VGG19 podemos observar os resultados da classificação na Figura 4.11, onde notamos que a VGG19 obteve melhor resultado no momento de classificar corretamente os melanomas, sendo 96 das 100 imagens. Porém, devemos prestar atenção que na classe não-melanoma 17 das 100 imagens foram classificadas erroneamente estimando que pertencem à classe melanoma.

Podemos concluir que isso levaria a considerar que os pacientes foram submetidos a procedimentos cirúrgicos para a retirada da lesão ou foram submetidos a mais testes para descartar um possível melanoma.

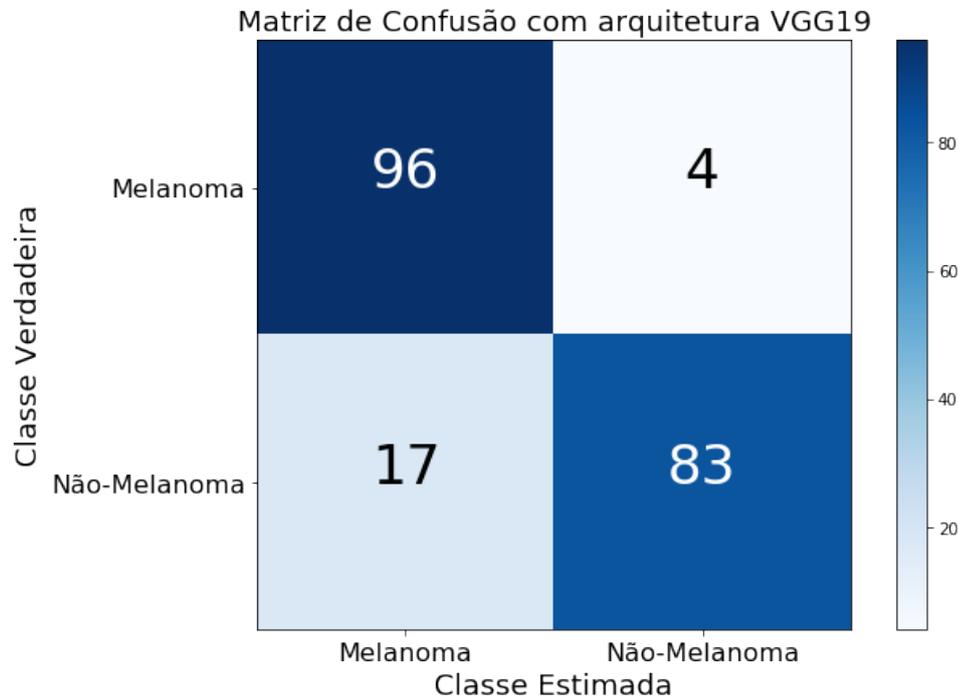


Figura 4.11 – Matriz de confusão da arquitetura VGG19 e o otimizador RMSprop do Modelo C.

Das equações mostradas na Seção 3.3.3, obtivemos os seguintes resultados:

- **Precisão** =  $0,8496 = 84,96\%$
- **Sensibilidade** =  $0,9600 = 96,00\%$
- **Especificidade** =  $0,8300 = 83,00\%$

Com esta arquitetura temos um classificador com desempenho alto no momento de prever o melanoma com uma sensibilidade de  $96,00\%$ , mas a especificidade de  $83,00\%$  e a precisão de  $84,96\%$  é baixa em comparação com as outras duas arquiteturas.

A Figura 4.12 mostra o resumo de quantas imagens foram classificadas corretamente e quantas foram classificadas de forma errada das três arquiteturas testadas. Podemos salientar que a AlexNet estimou melhor as imagens da classe não-melanoma classificando acertadamente 96 das 100 imagens.

A arquitetura VGG19 obteve melhor desempenho no momento de prever as imagens da classe melanoma acertando 96 das 100 imagens.

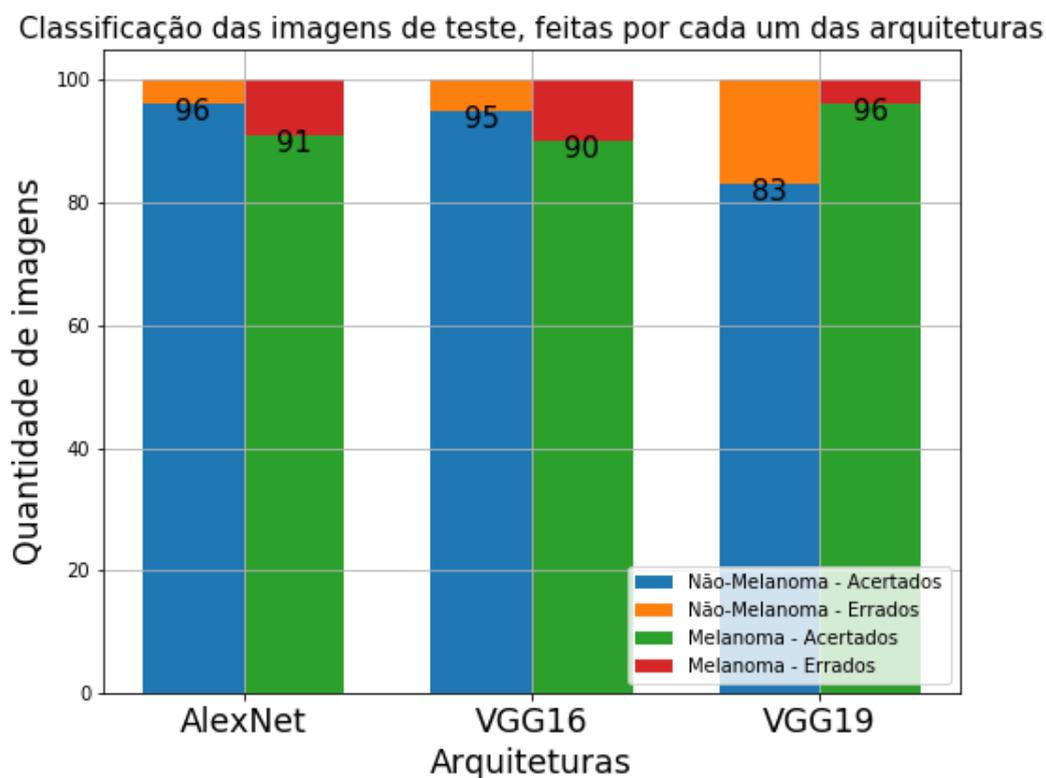


Figura 4.12 – Resumo das imagens de teste com cada uma das arquiteturas treinadas com o Modelo C.

Porém em classificação de imagens médicas há muitas considerações a serem levadas em conta, uma delas é a sensibilidade com a qual vamos prever as doenças, já que é um tópico sensível dizer qual é o problema ao que esta enfrentando-se o paciente.

Nossa proposta é prever com antecedência, eficácia, e a maior rapidez possível qual tipo de lesões de pele o paciente tem, a fim de dar uma melhor perspectiva ao médico para ter um diagnóstico e tomar as ações necessárias.

#### 4.2.4.4 Curvas ROC e a área sob a curva AUC

A curva ROC da Figura 4.13 mostra o desempenho das três arquiteturas avaliadas em relação ao problema de classificação de imagens, onde podemos avaliar o comportamento delas quanto a sensibilidade e a especificidade.

Aqui podemos conferir que a AlexNet é aquela que apresenta os melhores valores no momento de classificar com respeito as outras arquiteturas avaliadas já que a curva dela (em cor roxo) encontra-se quase mais próximo do valor ideal 1. As outras arquiteturas também tem valores aceitáveis no momento de classificar.

A Tabela 4.6 mostra os valores da área sob a curva, onde como ja tínhamos mencionado a arquitetura AlexNet foi aquela que apresentou melhor desempenho sendo

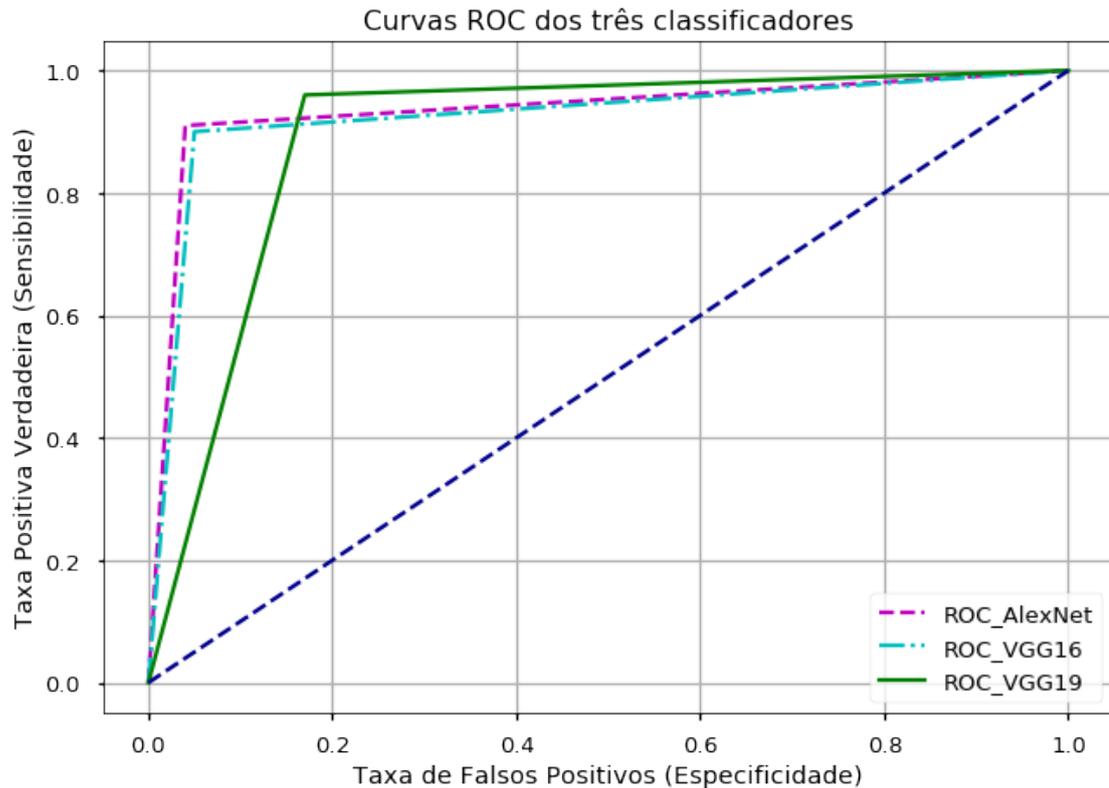


Figura 4.13 – ROC dos três classificadores.

de  $AUC = 0,935$  no momento de classificar mas é primordial melhorar o resultado para obter valores melhores e assim desenvolver uma ferramenta mais viável no momento de reconhecer o melanoma para que o médico possa decidir um diagnóstico melhor, já que antecipando os reconhecimentos dos melanomas em suas fases iniciais vai evitar o espalhamento pelo corpo (metástase).

Tabela 4.6 – Medidas da área sob a curva AUC, das três arquiteturas.

	AlexNet	VGG16	VGG19
AUC	<b>0,935</b>	0,925	0,895

(VIJAYAM, 2015) e (JØRGENSEN *et al.*, 2008) empregam técnicas de *machine learning* com a finalidade de melhorar o desempenho no reconhecimento dos não-melanomas com o objetivo principal de diminuir as cirurgias dos pacientes que não tiverem nenhum risco de evoluir a lesão, já que na maioria dos casos o melhor jeito de evitar qualquer problema de câncer é tirando a lesão.

Segundo os resultados dados por (ROGERS *et al.*, 2015), mostra o total de pacientes tratados por casos de não-melanoma nos Estados Unidos. (Serener; Serte, 2019) apresenta a importância na classificação dos não-melanomas com o uso de *deep learning*.

## 5 Conclusões e Trabalhos Futuros

Neste capítulo são apresentadas as conclusões obtidas ao longo da pesquisa com cada um dos modelos propostos e as arquiteturas de aprendizagens treinadas levando em consideração os valores obtidos tanto na *accuracy*, quanto no *loss*, assim como as métricas da matriz de confusão obtida a partir do conjunto de teste para cada uma das arquiteturas. Além disso, são apresentadas algumas propostas sob trabalhos futuros que podem surgir a partir desta pesquisa.

### 5.1 Conclusões

Nossa proposta de classificação binária ajuda na detecção precoce das neoplasias cutâneas conhecidas como melanoma, nos dois primeiros modelos propostos o A e o B foram treinados com uma única arquitetura que foi a AlexNet a qual mostrou bons resultados de *accuracy* e *loss* sob os três conjuntos de dados (treinamento, validação e teste). Com o Modelo A o qual possui 2400 imagens para o treinamento, o otimizador SGD é o melhor atingindo uma *accuracy* de 97,66%, no conjunto de validação obteve 78,00% e no teste foi de 70,50%. Neste contexto, notamos a melhoria da aprendizagem com o aumento na quantidade de imagens no conjunto de treinamento.

De forma similar, para o Modelo B os testes foram feitos nos três conjuntos (treinamento, validação e teste), da mesma forma o melhor otimizador foi o SGD, o qual atingiu uma *accuracy* de 99,79% no conjunto de treinamento, 81,50% no conjunto de validação (1000 imagens) e de 79,17% no conjunto de teste (600 imagens). Todos estes resultados foram obtidos no treinamento com o Modelo B que foi com 4000 imagens. É importante atentar nos resultados avaliados pelo *loss*, já que com esta métrica nós podemos observar se a arquitetura do modelo está generalizando os dados ou se ele está tendendo para um sobre-ajuste ou *overfitting*.

Ao observar que existe uma melhora nos resultados conforme há um aumento dos dados o que representa maior variabilidade, procuramos encontrar mais quantidades de imagens com o fim de melhorar o nosso processo de aprendizagem. Assim, o Modelo C contém 8244 imagens para o treinamento. Nesta metodologia avaliamos as três arquiteturas propostas, onde a AlexNet conjuntamente com o otimizador Adam atingiram as *accuracies* mais altas tanto no conjunto de validação e teste sendo de 93,17% e 92,50% em comparação com as outras duas arquiteturas VGG16 e VGG19. Ao comparar as figuras do desempenho das três arquiteturas a VGG19 mostra uma melhor generalização dos dados,

já que a curva do *loss* não entra no *overfitting*. No entanto o aumento de camadas nas arquiteturas não mostrou uma melhor aprendizagem na tarefa de classificação.

Em conclusão, devido aos bons resultados promisorios dados pela arquitetura AlexNet na tarefa de classificação obtendo uma AUC de 0,935 com um conjunto de dados relativamente pequeno com respeito ao outras pesquisas feitas com o uso de *deep learning*, podemos concluir que uma ótima classificação é possível com ajuda de outras ferramentas como processamento de imagens, segmentação de imagens, entre outras, permitindo um possível desenvolvimento de uma ferramenta útil para um ambiente clínico que possa ajudar aos médicos na tomada de decisões.

## 5.2 Trabalhos Futuros

Considerando os bons resultados nos nossos experimentos de classificação binária e a crescente necessidade da otimização de processos nas áreas médicas com a criação de sistemas de classificação, detecção, entre outras seria benéfico expandir a pesquisa considerando as seguintes propostas:

- Devido a que o *ISIC archive* é uns dos únicos *datasets* de acesso público com diversas anotações em cada uma das suas imagens, se poderia testar modelos de arquiteturas GAN ("*Generative Adversial Networks*") para a criação de imagens sintéticas baseadas nas imagens originais, já que os avances de pesquisa no uso de GANs poderiam produzir as imagens sintéticas o mais realistas possível permitindo um melhor desempenho na aprendizagem.
- Incluir mais classes para permitir ter uma melhor abrangência no diagnostico em lesões de pele.
- Com mais quantidades de dados poderíamos testar modelos de *deep learning* mais profundos adicionando camadas ou empregando redes residuais, da mesma forma fazer *fine-tuning* para complementar alguns modelos de aprendizagem com outros misturando diversas arquiteturas.
- Adicionar blocos no modelo do sistema já seja na entrada com um pre-processamento das imagens ou na saída com algoritmos de *machine learning* para a classificação dos valores dos vetores apresentados na saída do aprendizagem feito como o *deep learning*.
- Classificar as imagens médicas adquiridas no Equador as quais estão sem rótulos e contém uma diversa quantidade de lesões de pele, para atingir dito problema verificaremos que tão benéfico é treinar arquiteturas com um certo *dataset*, e

provar dito aprendizagem em nossas imagens. Conseguindo obter uma comparativa entre dois modelos diferentes de imagens.

## Referências

- ALBERT, M. R.; WEINSTOCK, M. A. Keratinocyte carcinoma. *CA: A Cancer Journal for Clinicians*, v. 53, n. 5, p. 292–302, 2003. Disponível em: <<https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/canjclin.53.5.292>>. Citado na página 31.
- ARGENZIANO, G.; CATRICALÀ, C.; ARDIGO, M.; BUCCINI, P.; SIMONE, P. D.; EIBENSCHUTZ, L.; FERRARI, A.; MARIANI, G.; SILIPO, V.; SPERDUTI, I.; ZALAUDEK, I. Seven-point checklist of dermoscopy revisited. *British Journal of Dermatology*, v. 164, n. 4, p. 785–790, 2011. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2133.2010.10194.x>>. Citado na página 18.
- BRINKER, T. J.; HEKLER, A.; HAUSCHILD, A.; BERKING, C.; SCHILLING, B.; ENK, A. H.; HAFERKAMP, S.; KAROGLAN, A.; KALLE, C. [von; WEICHENTHAL, M.; SATTLER, E.; SCHADENDORF, D.; GAISER, M. R.; KLODE, J.; UTIKAL, J. S. Comparing artificial intelligence algorithms to 157 german dermatologists: the melanoma classification benchmark. *European Journal of Cancer*, v. 111, p. 30 – 37, 2019. ISSN 0959-8049. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0959804918315624>>. Citado na página 18.
- BUDUMA, N.; LOCASCIO, N. *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491925612. Citado na página 20.
- CASTRO, F. d. *Inteligência artificial vence dermatologistas no diagnóstico de câncer de pele - Saúde*. Estadão, 2018. Disponível em: <<https://saude.estadao.com.br/noticias/geral,inteligencia-artificial-vence-dermatologistas-no-diagnostico-de-cancer-de-pele,70002328295>>. Citado na página 17.
- CESTARI, S. *Dermatologia Pediátrica*. 1st. ed. [S.l.]: Editora Atheneu, 2012. ISBN 978-8538802983. Citado 2 vezes nas páginas 29 e 30.
- CHOLLET, F. *Deep learning with Python*. [S.l.]: Manning Publications Co., 2018. Citado 2 vezes nas páginas 17 e 28.
- CHOLLET, F. *et al. Keras*. 2015. Disponível em: <<https://keras.io>>. Citado 3 vezes nas páginas 36, 39 e 45.
- CURIEL-LEWANDROWSKI, C.; NOVOA, R.; BERRY, E.; CELEBI, M. E.; CODELLA, N.; GIUSTE, F.; GUTMAN, D.; HALPERN, A.; LEACHMAN, S.; LIU, Y.; LIU, Y.; REITER, O.; TSCHANDL, P. Artificial intelligence approach in melanoma. In: \_\_\_\_\_. [S.l.: s.n.], 2019. p. 1–31. ISBN 978-1-4614-7322-0. Citado na página 17.
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009. Citado 3 vezes nas páginas 18, 42 e 51.
- DUMOULIN, V.; VISIN, F. *A guide to convolution arithmetic for deep learning*. 2016. Citado 2 vezes nas páginas 26 e 28.

- ESTEVA, A. *Artificial intelligence in healthcare and medicine : enhancing the expert*. 2018. Disponível em: <<http://purl.stanford.edu/xg779kd4737>>. Citado na página 21.
- ESTEVA, A.; KUPREL, B.; NOVOA, R. A.; KO, J.; SWETTER, S. M.; BLAU, H. M.; THRUN, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, Macmillan Publishers Limited, part of Springer Nature. All rights reserved., v. 542, p. 115–, jan. 2017. Disponível em: <<http://dx.doi.org/10.1038/nature21056>>. Citado 2 vezes nas páginas 17 e 18.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Pearson, 2018. ISBN 9780133356724. Citado 2 vezes nas páginas 32 e 33.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado 3 vezes nas páginas 20, 22 e 26.
- GORDON, R. Skin cancer: An overview of epidemiology and risk factors. *Seminars in Oncology Nursing*, v. 29, n. 3, p. 160 – 169, 2013. ISSN 0749-2081. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0749208113000326>>. Citado na página 31.
- GÉRON, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491962291, 9781491962299. Citado 2 vezes nas páginas 41 e 43.
- HAN, S.; KIM, M.; LIM, W.; PARK, G.; PARK, I.; CHANG, S. Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. *Journal of Investigative Dermatology*, v. 138, 02 2018. Citado na página 18.
- HENNING, J. S.; DUSZA, S. W.; WANG, S. Q.; MARGHOOB, A. A.; RABINOVITZ, H. S.; POLSKY, D.; KOPF, A. W. The cash (color, architecture, symmetry, and homogeneity) algorithm for dermoscopy. *Journal of the American Academy of Dermatology*, v. 56, n. 1, p. 45 – 52, 2007. ISSN 0190-9622. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0190962206025278>>. Citado na página 18.
- HOSSIN, M.; M.N, S. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, v. 5, p. 01–11, 03 2015. Citado na página 36.
- HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 39.
- INCA. *Estimativa 2018 - Síntese de Resultados e Comentários*. 2016. Accessed = 2019-09-11. Disponível em: <<http://www1.inca.gov.br/estimativa/2018/sintese-de-resultados-comentarios.asp>>. Citado 2 vezes nas páginas 16 e 31.
- INCA. *Câncer de pele melanoma*. 2018. Accessed = 2019-09-13. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>>. Citado na página 16.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. Disponível em: <<http://arxiv.org/abs/1502.03167>>. Citado na página 41.

- JØRGENSEN, T. M.; TYCHO, A.; MOGENSEN, M.; BJERRING, P.; JEMEC, G. B. Machine-learning classification of non-melanoma skin cancers from image features obtained by optical coherence tomography. *Skin Research and Technology*, v. 14, n. 3, p. 364–369, 2008. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0846.2008.00304.x>>. Citado na página 59.
- JUNQUEIRA, L.; CARNEIRO, J. *Histologia básica*. 12. ed. [S.l.]: Rio de Janeiro: Guanabara Koogan, 2013. ISBN 978-85-277-2311-4. Citado na página 29.
- KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2014. Cite arxiv:1412.6980 Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. Disponível em: <<http://arxiv.org/abs/1412.6980>>. Citado na página 44.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, v. 25, 01 2012. Citado na página 40.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. Disponível em: <<https://doi.org/10.1038/nature14539>>. Citado 3 vezes nas páginas 23, 24 e 26.
- LITJENS, G. J. S.; KOOI, T.; BEJNORDI, B. E.; SETIO, A. A. A.; CIOMPI, F.; GHAFORIAN, M.; LAAK, J. A. W. M. van der; GINNEKEN, B. van; SÁNCHEZ, C. I. A survey on deep learning in medical image analysis. *Medical Image Anal.*, v. 42, p. 60–88, 2017. Disponível em: <<http://dblp.uni-trier.de/db/journals/mia/mia42.html#LitjensKBSCGLGS17>>. Citado na página 31.
- MAHAJAN, D.; GIRSHICK, R.; RAMANATHAN, V.; HE, K.; PALURI, M.; LI, Y.; BHARAMBE, A.; MAATEN, L. van der. Exploring the limits of weakly supervised pretraining. In: *The European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. Citado na página 17.
- MAZUROWSKI, M. A.; HABAS, P. A.; ZURADA, J. M.; LO, J. Y.; BAKER, J. A.; TOURASSI, G. D. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, v. 21, n. 2, p. 427 – 436, 2008. ISSN 0893-6080. Advances in Neural Networks Research: IJCNN 07. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608007002407>>. Citado na página 38.
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115–133, 1943. Disponível em: <<https://doi.org/10.1007/BF02478259>>. Citado 2 vezes nas páginas 24 e 25.
- MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado na página 20.
- NACHBAR, F.; STOLZ, W.; MERKLE, T.; COGNETTA, A. B.; VOGT, T.; LANDTHALER, M.; BILEK, P.; BRAUN-FALCO, O.; PLEWIG, G. The abcd rule of dermatoscopy: High prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, v. 30, n. 4, p. 551 – 559, 1994. ISSN 0190-9622. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0190962294700613>>. Citado na página 18.

NIELSEN, M. A. *Neural Networks and Deep Learning*. [S.l.]: Determination Press, 2015. Citado na página 28.

PATTERSON, J.; GIBSON, A. *Deep Learning: A Practitioner's Approach*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491914254. Citado na página 23.

PROJECT, I. *Public skin lesion image archive*. 2018. Disponível em: <<https://www.isic-archive.com/#!//topWithHeader/onlyHeaderTop/gallery>>. Citado na página 34.

RAMPAZZO, W. *Aprendizado profundo e interfaces cérebro-computador*. Tese (Doutorado), 2019. Disponível em: <<http://repositorio.unicamp.br/jspui/handle/REPOSIP/335555>>. Citado na página 26.

ROGERS, H. W.; WEINSTOCK, M. A.; FELDMAN, S. R.; COLDIRON, B. M. Incidence Estimate of Nonmelanoma Skin Cancer (Keratinocyte Carcinomas) in the US Population, 2012. *JAMA Dermatology*, v. 151, n. 10, p. 1081–1086, 10 2015. ISSN 2168-6068. Disponível em: <<https://doi.org/10.1001/jamadermatol.2015.1187>>. Citado na página 59.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958. Citado na página 24.

RUDER, S. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016. Disponível em: <<http://arxiv.org/abs/1609.04747>>. Citado na página 44.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. In: \_\_\_\_\_. *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988. p. 696–699. ISBN 0262010976. Citado na página 26.

SBD. *Câncer da pele*. 2017. Disponível em: <<https://www.sbd.org.br/dermatologia/pele/doencas-e-problemas/cancer-da-pele/64/>>. Citado na página 31.

Serener, A.; Serte, S. Keratinocyte carcinoma detection via convolutional neural networks. In: *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. [S.l.: s.n.], 2019. p. 1–5. Citado na página 59.

SHI, Z.; HE, L.; NAKAMURA, T.; SUZUKI, K.; ITOH, H. Survey on neural networks used for medical image processing. *International journal of computational science*, v. 3, 02 2009. Citado na página 31.

SIMONYAN, K.; ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. Citado na página 41.

SOCIETY, A. C. *Melanoma Skin Cancer Risk Factors: Melanoma Risk Factors*. 2017. Accessed = 2019-09-13. Disponível em: <<https://www.cancer.org/cancer/melanoma-skin-cancer/causes-risks-prevention/risk-factors.html>>. Citado na página 16.

SOCIETY, A. C. *Ultraviolet (UV) Radiation*. 2017. Accessed = 2019-09-11. Disponível em: <<https://www.cancer.org/cancer/cancer-causes/radiation-exposure/uv-radiation.html>>. Citado na página 16.

- SOCIETY, A. C. *What Is Melanoma Skin Cancer?: What Is Melanoma?* 2017. Accessed = 2019-09-11. Disponível em: <<https://www.cancer.org/cancer/melanoma-skin-cancer/about/what-is-melanoma.html>>. Citado na página 16.
- SOCIETY, A. C. *Key Statistics for Melanoma Skin Cancer*. 2019. Accessed = 2019-09-11. Disponível em: <<https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html>>. Citado na página 16.
- SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, v. 45, n. 4, p. 427 – 437, 2009. ISSN 0306-4573. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0306457309000259>>. Citado na página 38.
- SOLA, J.; SEVILLA, J. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, v. 44, n. 3, p. 1464–1468, June 1997. Citado na página 40.
- SOYER, P.; ARGENZIANO, G.; ZALAUDEK, I.; CORONA, R.; SERA, F.; TALAMINI, R.; BARBATO, F.; BARONI, A.; CICALÉ, L.; STEFANI, A. D.; FARRO, P.; ROSSIELLO, L.; RUOCCO, E.; CHIMENTI, S. Three-point checklist of dermoscopy. *Dermatology (Basel, Switzerland)*, v. 208, p. 27–31, 02 2004. Citado na página 18.
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: *Computer Vision and Pattern Recognition (CVPR)*. [s.n.], 2015. Disponível em: <<http://arxiv.org/abs/1409.4842>>. Citado na página 18.
- VANHEULE, R.; PIŽURICA, A. *Deep Learning in Skin Lesion Classification Tasks*. 2018. Disponível em: <<http://lib.ugent.be/catalog/rug01:002495010>>. Citado 3 vezes nas páginas 21, 23 e 39.
- VIJAYAM, S. Categorization of non-melanoma skin lesion diseases using support vector machine and its variants. In: . [S.l.: s.n.], 2015. Citado na página 59.
- WANG, Y.; WEI, G.; BROOKS, D. Benchmarking tpu, gpu, and CPU platforms for deep learning. *CoRR*, abs/1907.10701, 2019. Disponível em: <<http://arxiv.org/abs/1907.10701>>. Citado na página 39.
- WU, H.; GU, X. Max-pooling dropout for regularization of convolutional neural networks. *CoRR*, abs/1512.01400, 2015. Disponível em: <<http://arxiv.org/abs/1512.01400>>. Citado 2 vezes nas páginas 41 e 43.

# APÊNDICE A – Listado de Publicações

## Artigos aceitos em Periódicos e Anais de Congressos

- **P. Minango**, Y. Iano, A.C.B. Monteiro, R.P. França and G.G. Oliveira; "**Automatic Skin Lesions Classification from Dermoscopic Images Employing Deep Learning**", in SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING, SET IJBE, December 2019. ISSN 2446-9432 / ISSN 2446-9246 (**APROVADO**)
- **P. Minango**, Y. Iano, A.C.B. Monteiro, R.P. França, G.G. Oliveira and D. Pajuelo; "**Classification of Skin Dermoscopy Images with the Application of Deep Learning Techniques**", Proceedings of the 5th Brazilian Technology Symposium Emerging Trends and Challenges in Technology: Springer Nature, 2020 (**APROVADO**)
- J. Minango, Y. Iano, **P. Minango**, D.A. Ibarra, A. Flores, and G.G. Oliveira; "Synchronization Reduction of the Conjugate Gradient Detector Used in Massive MIMO Uplink", Proceedings of the 5th Brazilian Technology Symposium Emerging Trends and Challenges in Technology: Springer Nature, 2020 (**APROVADO**)
- G.G. Oliveira, Y. Iano, A.C.B. Monteiro, R.P. França, **P. Minango** and D. Pajuelo; "The historical development of World Broadcasting and its impact reflected in Brazil", in SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING, SET IJBE, December 2019. ISSN 2446-9432 / ISSN 2446-9246 (**APROVADO**)
- D. Pajuelo, Y. Iano, **P. Minango**, G.G. Oliveira, A.C.B. Monteiro and R.P. França; "Video Content Delivery Schemes: Approaches and Directions", in SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING, SET IJBE, December 2019. ISSN 2446-9432 / ISSN 2446-9246 (**APROVADO**)