



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

JÉSSICA BARBOSA DINIZ

DESENVOLVIMENTO DE UMA GRAMÁTICA
PARA CONSTRUÇÃO AUTOMÁTICA DE
ARQUITETURAS DE DEEP LEARNING
APLICADO À CLASSIFICAÇÃO DE IMAGENS
DERMATOSCÓPICAS

RECIFE-PE

2019

JÉSSICA BARBOSA DINIZ

**DESENVOLVIMENTO DE UMA GRAMÁTICA
PARA CONSTRUÇÃO AUTOMÁTICA DE
ARQUITETURAS DE DEEP LEARNING
APLICADO À CLASSIFICAÇÃO DE IMAGENS
DERMATOSCÓPICAS**

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Computação da Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

ORIENTADOR: Filipe Rolim Cordeiro

CO-ORIENTADOR: Péricles Barbosa Cunha de Miranda

RECIFE-PE

2019

JÉSSICA BARBOSA DINIZ

DESENVOLVIMENTO DE UMA GRAMÁTICA
PARA CONSTRUÇÃO AUTOMÁTICA DE
ARQUITETURAS DE DEEP LEARNING
APLICADO À CLASSIFICAÇÃO DE IMAGENS
DERMATOSCÓPICAS

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Computação da Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

Aprovada em: 24 de Maio de 2019.

BANCA EXAMINADORA

Filipe Rolim Cordeiro (Orientador)
Universidade Federal Rural de Pernambuco
Departamento de Computação - DC

Valmir Macário
Universidade Federal Rural de Pernambuco
Departamento de Computação - DC

Wellington Pinheiro dos Santos
Universidade Federal de Pernambuco
Departamento de Engenharia Biomédica

À minha mãe e a todos aqueles acreditam no
uso da Ciência da Computação para trazer
melhorias à sociedade.

Agradecimentos

Agradeço de forma especial aos meus pais, José Lopes e Juanita Merencia, pela determinação, luta e por não medirem esforços para que meus irmãos e eu pudéssemos levar os estudos adiante, sempre permitindo que seguíssemos nossos sonhos.

Agradeço aos meus irmãos que por mais difícil que fossem as circunstâncias, sempre tiveram paciência e confiança.

Ao Prof. Filipe Rolim, obrigada pela orientação no amadurecimento dos meus conhecimentos e conceitos que levaram a execução desta dissertação, pelo apoio constante nos momentos de incertezas e pela atenção durante todo o período de construção deste trabalho, sempre mantendo o foco mesmo nos momentos que achei mais difíceis. Agradeço também ao Prof. Péricles Miranda por ter sido de singular importância especialmente no entendimento inicial de importantes conceitos que guiaram esse estudo.

Aos colegas de aula, de trabalho, professores e todos com os quais tive oportunidade de conviver durante o mestrado e às demais pessoas que me proporcionaram momentos de apoio, lazer e reflexão tão importantes para o bom andamento das atividades e para superação das dificuldades do dia a dia.

Finalmente agradeço a Deus, por proporcionar estes agradecimentos a todos que tornaram minha vida mais afetuosa, além de ter me dado uma família maravilhosa e amigos sinceros. Viver é o meu modo de agradecer sempre. Sou extremamente grata a todos vocês!

“Não importa o que aconteça,
continue a nadar.”

(Graham Walters; Procurando Nemo, 2003)

Resumo

O melanoma é um tumor maligno originário nas células produtoras de pigmentos naturais da pele, a melanina. Causado pela produção excessiva da melanina, costuma ter forma de lesões pigmentadas sobre a pele. Trata-se de um dos mais perigosos tipos de câncer de pele. O número crescente de incidentes e causas de mortes fez com que surgisse uma maior busca por soluções de detecção da doença especialmente em seus estágios iniciais, onde ela frequentemente se mostra falha, principalmente quando o profissional que analisa a lesão não é experiente. Por essa razão, é de grande importância o uso de ferramentas computacionais para o auxílio na análise das lesões, de forma que os resultados se tornem mais precisos e confiáveis, viabilizando, assim, a correta identificação do tumor. Outro aspecto importante é a democratização do diagnóstico disponibilizando conhecimento anteriormente limitado a especialistas. O *deep learning* é uma área de pesquisa em alta nos últimos anos devido à sua aplicação bem-sucedida em diversos domínios, como visão computacional e reconhecimento de imagem. Sua técnica mais proeminente é a rede neural convolutiva, que permite a rede o aprendizado de representações automaticamente. No entanto, a técnica tem algumas limitações, projetar essas redes não é uma tarefa fácil e o domínio requer bastante conhecimento. Neste trabalho, apresentamos o uso do algoritmo genético associado à programação genética baseada em gramática para otimizar arquiteturas de redes neurais convolutivas. É apresentada uma gramática capaz de explorar o espaço de busca do problema de forma a viabilizar a criação de indivíduos válidos capazes de serem qualitativamente analisados. Para avaliação da abordagem proposta, foi utilizado um conjunto de imagens do *International Skin Imaging Collaboration* (ISIC) durante a validação da evolução das arquiteturas produzidas, usando a métrica da acurácia para avaliar o desempenho de classificação no conjunto de dados de teste. Os resultados demonstram que o método usando programação genética baseada em gramática pode facilmente produzir arquiteturas CNN otimizadas que são competitivas e alcançam resultados de alta precisão, se mostrando superior em comparação com técnicas do estado da arte.

Palavras-chave: *Deep Learning*. Redes Neurais Convolutivas. Algoritmo Genético. Programação Genética baseada em Gramática. Melanoma

Abstract

Melanoma is a type of cancer originated in the pigment-producing melanocytes of the skin and caused by over production of melanin pigments by melanocytes. It is also known as malignant melanoma and often resembles moles or pigmented skin lesions. Melanoma is the most dangerous type of skin cancer. The continuous increase in incidence rates and melanoma mortality have guided searches aiming to help specially on early detections, where physicists usually fail on detecting and the diagnosis is most efficient. Therefore, the use of computational tools helping the analysis of skin lesions is very important. These tools can increase the accuracy and precision of the correct diagnosis of the lesion. Another important aspect that should be considered is the democratization of the disease diagnosis sharing knowledge that used to be limited to specialized people. Deep Learning is a research area under the spotlight in recent years due to its successful application to many domains, such as computer vision and image recognition. The most prominent technique derived from Deep Learning is Convolutional Neural Network, which allows the network to learn representations needed for detection or classification tasks automatically. Deep learning convolutional neural networks (CNN) may facilitate melanoma detection. However, CNN have some limitations, as designing these networks are not easy to master and require expertise and insight. In this work, we present the use of genetic algorithm associated to grammar-based genetic programming to optimize convolution neural network architectures. It is presented a grammar capable of exploring a search space from the problem. The search using the grammar ensure the criation of valid individuos that can be qualitatively analyzed. To evaluate our proposed approach, we adopted images from ISIC dataset to validate the evolution of the generated architectures, using the metric of accuracy to evaluate its classification performance in the test dataset. The results demonstrate that our method using grammar-based genetic programming can easily produce optimized CNN architectures that are competitive and achieve high accuracy results, showing superior results in comparison with the state-of-the-art methods.

Keywords: Deep Learning. Convolutional Neural Network. Genetic Algorithm. Grammar based Genetic Programming. Melanoma

Lista de Figuras

Figura 1 – Exemplos de lesões de pele.	17
Figura 2 – Exemplo de como lesões de pele são visualizadas através do dermatoscópio; as imagens na primeira linha são fotografias macroscópicas e na segunda linha são as respectivas imagens vistas por um aparelho dermatoscópico com capacidade de zoom de 10x. . . .	18
Figura 3 – Artefatos encontrados em imagens dermatoscópicas que atrapalham a automação da detecção da lesão.	19
Figura 4 – Exemplo de visualização de uma lesão de pele. a) Imagem corporal da localização da lesão; b) Visualização macroscópica a “olho nu” da lesão; c) Visualização dermatoscópica da lesão.	28
Figura 5 – Técnica ABCDE para diagnóstico de melanoma.	29
Figura 6 – Fluxograma da técnica de Análise de Padrões.	30
Figura 7 – Representação de imagem digital em <i>pixels</i>	31
Figura 8 – Representação da relação de <i>pixels</i> com as informações que cada um contém. a) Imagem digital; b) <i>Zoom</i> em grupo de 9×9 <i>pixels</i> da imagem (a); c) Valores de cada um dos <i>pixels</i>	32
Figura 9 – Exemplo de uma mesma imagem digital representada em diferentes resoluções.	32
Figura 10 – Exemplo de composição de uma imagem do tipo RGB.	33
Figura 11 – Estrutura básica de um neurônio biológico e sua representação em modelo computacional.	36
Figura 12 – Arquitetura básica de um rede neural.	38
Figura 13 – Acurácia do ImageNet ao longo dos anos. Marcações azul correspondem a abordagens tradicionais de visão computacional e marcações em verde representam técnicas de <i>deep learning</i>	39
Figura 14 – Sequência de camadas do <i>deep learning</i> com o reconhecimento de características.	40
Figura 15 – Exemplo de aplicação da técnica de <i>data augmentation</i> na imagem original de um cachorrinho.	41

Figura 16 – Exemplo da aplicação de um filtro de tamanho 2×2 que calcula a média com um <i>stride</i> de tamanho 2.	42
Figura 17 – Sequência de camadas da CNN disposta em função de três dimensões.	43
Figura 18 – Exemplo de aplicação de um <i>MaxPooling</i>	44
Figura 19 – Funcionamento de uma rede CNN.	45
Figura 20 – Arquitetura do LeNet-5.	46
Figura 21 – Módulos de <i>inception</i> do GoogLeNet.	47
Figura 22 – Arquitetura do GoogLeNet.	48
Figura 23 – Arquitetura do VGGNet com 16 camadas.	49
Figura 24 – Bloco residual utilizados na composição do ResNet.	50
Figura 25 – Indivíduos de uma população com seus devidos valores de <i>fitness</i> e a representação da população na roleta.	55
Figura 26 – Operação de reprodução de dois pontos, ou <i>two-point crossover</i>	56
Figura 27 – Exemplo de uma operação de mutação.	56
Figura 28 – Forma de representação utilizada na programação genética.	58
Figura 29 – Fluxo completo da programação genética.	59
Figura 30 – Cruzamento na programação genética.	60
Figura 31 – Mutação na programação genética.	60
Figura 32 – Fluxo da programação genética baseada em gramática.	61
Figura 33 – Exemplo de uma árvore sintática para representação do indivíduo $x+(x*x)$	63
Figura 34 – Exemplo de uma operação de cruzamento em um ponto no cromossomo.	66
Figura 35 – Fluxograma do método proposto.	75
Figura 36 – Primeira gramática utilizada pelo estudo.	76
Figura 37 – Representação da melhor arquitetura da gramática apresentada em Diniz et al. (2018) para a base de dados CIFAR-10.	77
Figura 38 – Evolução da gramática inicialmente estudada.	78
Figura 39 – Gramática proposta no estudo.	80
Figura 40 – Árvore de derivação da gramática proposta.	81
Figura 41 – Representação de um indivíduo produzido pela gramática.	82
Figura 42 – Representação de um indivíduo produzido pela gramática com arquitetura maior e mais profunda.	83

Figura 43 – Exemplo de uma rede neural convolutiva gerada para um indivíduo produzido pela gramática.	83
Figura 44 – Gráfico de representação dos resultados por gerações.	92
Figura 45 – Gráfico do desvio padrão da média da acurácia por geração.	93
Figura 46 – Exemplos de CNNs da primeira geração.	94
Figura 47 – Exemplos de CNNs da última geração.	95
Figura 48 – Resultados da acurácia em função do tempo de execução de CNNs da primeira (azul) e da última geração (verde) com visualização da quantidade de parâmetros de cada CNN.	96
Figura 49 – Representação do melhor indivíduo produzido pela gramática.	97
Figura 50 – Arquitetura do indivíduo que apresentou maior performance.	97
Figura 51 – Gráfico com os valores de acurácia do treinamento (<i>acc</i>), acurácia da validação (<i>val_acc</i>), <i>loss</i> do treinamento (<i>loss</i>) e <i>loss</i> da validação (<i>val_loss</i>) ao longo das época referente as três execuções do melhor indivíduo.	98
Figura 52 – Gráfico de representação dos resultados dos algoritmos do estado da arte. Todos os resultados apresentam o desvio padrão e o sufixo ‘-trans’ se refere a aplicação da rede utilizando a técnica de transferência de aprendizado.	99
Figura 53 – Resultados em função do tempo de execução pela acurácia com visualização da quantidade de parâmetros de cada CNN. O sufixo ‘-trans’ se refere a aplicação da rede utilizando a técnica de transferência de aprendizado. O segundo gráfico se refere a ampliação do gráfico principal na parte mais aglomerada das bolhas.	100

Lista de tabelas

Tabela 1 – Estudos analisados no estado da arte.	72
Tabela 2 – Exemplo da matriz de confusão.	87
Tabela 3 – Parâmetros das execuções da programação genética.	90
Tabela 4 – Matriz de confusão.	98

Lista de Algoritmos

1	Pseudocódigo de um algoritmo evolucionário.	54
---	---	----

Lista de Siglas

ANN	<i>Artificial Neural Network</i>
BNF	<i>Backus–Naur form</i> ou <i>Backus Normal Form</i>
CAD	<i>Computer-Aided Diagnosis</i>
CDS	<i>Clinical Diagnosis Support</i>
CE	Computação Evolucionária
CNMD	<i>Consensus Net Meeting on Dermoscopy</i>
CNN	<i>Convolutional Neural Network</i>
ConvNet	<i>Convolutional Neural Network</i>
FC	<i>Fully Connected</i>
GGGP	<i>Grammar-guided Genetic Programming</i>
GLC	Gramática Livre de Contexto
GPU	<i>Graphics Processing Unit</i>
HSV	<i>Hue, Saturation, Value</i>
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
ISIC	<i>International Skin Imaging Collaboration</i>
LBP	<i>Local Binary Patterns</i>
PG	Programação Genética
PGOG	Programação Genética Orientada a Gramática
PIGrow	<i>Position Independent Grow</i>
ReLu	<i>Rectified Linear Unit</i>
RGB	<i>Red-Green-Blue</i>
TPU	<i>Tensor Processing Unit</i>

Sumário

1	Introdução	16
1.1	Motivação	16
1.2	Problema de Pesquisa	22
1.3	Objetivos	25
1.3.1	Objetivo Geral	25
1.3.2	Objetivos Específicos	25
1.4	Estrutura do Trabalho	25
2	Fundamentação Teórica	27
2.1	Fundamentos Médicos	27
2.1.1	Dermatoscopia	27
2.1.2	Principais Técnicas para Detecção Clínica do Melanoma	28
2.2	Fundamentos de Imagem Digital	30
2.2.1	<i>Pixel</i>	30
2.2.2	Representação de Imagem Digital	31
2.2.3	Sistema de Cores RGB	31
2.3	Aprendizagem de Máquina	33
2.3.1	Redes Neurais Artificiais	35
2.3.2	<i>Deep Learning</i>	38
2.3.3	Redes Neurais Convolutivas	41
2.3.3.1	LeNet	45
2.3.3.2	GoogLeNet	46
2.3.3.3	VGGNet	48
2.3.3.4	ResNet	49
2.3.3.5	Transferência de Conhecimento	50
2.4	Algoritmos Evolucionários	51
2.4.1	Princípios da Evolução Natural	51
2.4.2	Computação Evolutiva	52
2.4.2.1	Programação Genética	57
3	Trabalhos Relacionados	67
3.1	Estado da Arte	67

3.2	Resumo do Estado da Arte e Contribuições do Trabalho Proposto	71
4	Método Proposto	74
4.1	Análises Prévias	74
4.2	Fluxograma do Método Proposto	74
4.3	Técnica Proposta	75
4.3.1	Evolução da Gramática Proposta	76
4.3.2	Gramática Proposta	79
4.4	Considerações do Capítulo	84
5	Metodologia	85
5.1	Base de Dados	85
5.2	Ambiente Experimental	86
5.2.1	Linguagem de Desenvolvimento	86
5.2.2	Ambiente de Desenvolvimento	86
5.3	Acurácia da Classificação	86
5.4	Técnicas Comparadas	87
5.5	Detalhes das Execuções	88
6	Resultados	91
6.1	Resultados Experimentais	91
6.2	Comparação dos Resultados com o Estado da Arte	97
6.3	Discussão dos Resultados	101
7	Conclusão	103
7.1	Considerações Finais e Contribuições do Trabalho	103
7.2	Trabalhos Futuros	104
	Referências	105
	GLOSSÁRIO	117

1 Introdução

1.1 Motivação

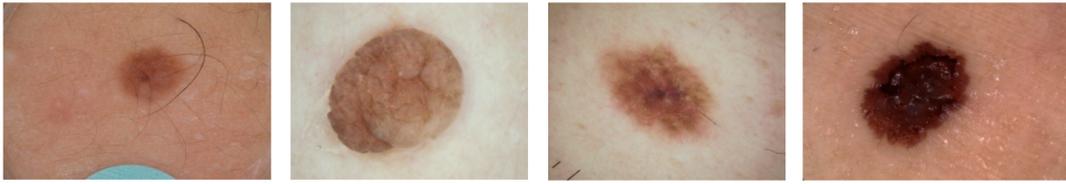
O melanoma é um tumor maligno originário nas células chamadas melanócitos; essas células são as produtoras de pigmentos naturais da pele, a melanina. Esse é um dos mais perigosos tipos de câncer de pele e pode se desenvolver também em partes do corpo como olhos, boca, área genital e anal ([American Cancer Society, 2016](#)). Em 2018, foram identificados aproximadamente 287.723 novos casos no mundo ([World Cancer Research Fund International, 2018](#) apud [BRAY et al., 2018](#)) e estima-se que em 2019, apenas nos Estados Unidos, 96.480 novos casos de melanomas serão diagnosticados (em média 57.220 em homens e 39.260 em mulheres) e 7.230 pessoas morrerão por conta da doença ([American Cancer Society, 2018](#)). No Brasil, os números também são altos, correspondendo a 30% dos tipos de câncer de pele ([Instituto Nacional de Câncer - INCA, 2018a](#)) e em 2018 foram detectados 6.260 novos casos ([Instituto Nacional de Câncer - INCA, 2018b](#)).

É reconhecido que a principal causa da doença é a exposição intensa aos raios ultravioleta (UV) ([AUTIER et al., 1998](#) apud [International Agency for Research on Cancer, 1992](#)) – normalmente os raios solares ou os provenientes de câmaras de bronzamento artificial – especialmente em pessoas que já possuem predisposição à doença. O problema surge quando células de DNA da pele são danificadas, fazendo com que sofram mutações genéticas levando-as a se multiplicar rapidamente e formar tumores malignos ([Skin Cancer Foundation, 2019](#)).

Por se originar a partir das células responsáveis pela produção da melanina, o melanoma costuma ter forma de lesões pigmentadas sobre a pele ([Skin Cancer Foundation, 2019](#)). A lesão de pele consiste em uma região da pele que é frequentemente possível de distinguir da pele normal, pois ela possui diferença de cor ou textura ([MISHRA; CELEBI, 2016](#)), como mostra a Figura 1. Essa característica possibilita a análise e o diagnóstico das lesões por meio de exame físico.

Em geral, o tumor tem um crescimento lento ([ARGENZIANO et al., 2010](#)), fator que acaba levando pacientes a procurarem ajuda médica quando a doença já tem até anos de existência. Apesar de não podermos assumir que toda pequena lesão de melanoma na pele irá evoluir para o estado mais avançado e grave da doença, as lesões metastáticas, a

Figura 1 – Exemplos de lesões de pele.



Fonte: [International Skin Imaging Collaboration \(2019\)](#).

detecção do melanoma nos seus primeiros estágios tem grande impacto no tratamento e posterior sucesso do mesmo ([HALPERN; LIEB, 2007](#); [KIRSNER et al., 2005](#)).

Infelizmente, a examinação clínica da pele pode apresentar elevados níveis de diagnósticos falso-positivos e em muitos casos lesões benignas são classificadas como malignas e acabam levando a cirurgias desnecessárias, enquanto outras malignas não são identificadas em seus estágios iniciais ([ARGENZIANO; SOYER, 2001](#); [CARLI et al., 2003](#)).

Nesse contexto, procedimentos médicos, como a dermatoscopia, se mostram de grande importância no processo de detecção da doença, especialmente quando comparado à inspeção clínica da pele sem auxílio de nenhuma ferramenta, o exame a “olho nu” ([VESTERGAARD et al., 2008](#)). A dermatoscopia trata-se de uma técnica não invasiva de diagnóstico a partir da observação da pele utilizando um aparelho chamado dermatoscópio. O equipamento possibilita a visualização de estruturas e cores da epiderme e derme não visíveis a “olho nu”, permitindo uma análise mais detalhada da lesão ([Dermoscopy, 2016](#); [ARGENZIANO; SOYER, 2001](#)), como mostrado na Figura 2. Apesar do uso do dermatoscópio aumentar a precisão de diagnósticos ([MAYER, 1997](#)), foi demonstrado que o aparelho nas mãos de profissionais inexperientes ocasiona a diminuição dessa precisão ([CELEBI et al., 2015](#)). Sendo assim, a análise das lesões continua dependendo da subjetiva experiência do médico ([KITTLER et al., 2002](#)), o que gera variações de diagnóstico inclusive entre os próprios dermatologistas se comparados com os resultados de um médico júnior¹ com os de um experiente².

Antes de 1980, normalmente o melanoma só era detectado quando a lesão de pele era grande, com presença de ulceração e sangramento ([RIGEL et al., 2010](#)). As abordagens de análise foram evoluindo e começaram a ser observadas características em comum que as

¹ Considerar experiência júnior os profissionais com menos de um ano de experiência.

² Considerar experiência experiente os profissionais com mais de um ano de experiência especificamente em dermatoscopia.

Figura 2 – Exemplo de como lesões de pele são visualizadas através do dermatoscópio; as imagens na primeira linha são fotografias macroscópicas e na segunda linha são as respectivas imagens vistas por um aparelho dermatoscópico com capacidade de zoom de 10x.



Fonte: Adaptado de (REZZE et al., 2006).

lesões malignas compartilhavam entre si, surgindo assim técnicas de auxílio ao diagnóstico. Ainda hoje, mesmo com toda a evolução na área, o reconhecimento, detecção e rápido tratamento do melanoma continuam sendo cruciais para a saúde do paciente.

Atualmente existem alguns métodos conhecidos na literatura que podem ser utilizados no diagnóstico do melanoma, dentre eles quatro, propostos pelo *Consensus Net Meeting on Dermoscopy* (CNMD)³, são amplamente aplicados na prática clínica para o diagnóstico do melanoma (SÁEZ et al., 2014): a Regra do ABCD, a Análise de Padrões, o Método Menzies e a Regra dos 7 Pontos. Todos explicados na Sessão 2.1.2.

A partir desses métodos clínicos, dermatologistas buscam alcançar ao máximo corretamente o objetivo em comum de cada um deles: a detecção da presença do melanoma. No entanto, mesmo havendo maior variedades de técnicas ao qual o médico pode escolher aplicar, o resultado ainda depende da seu entendimento subjetivo.

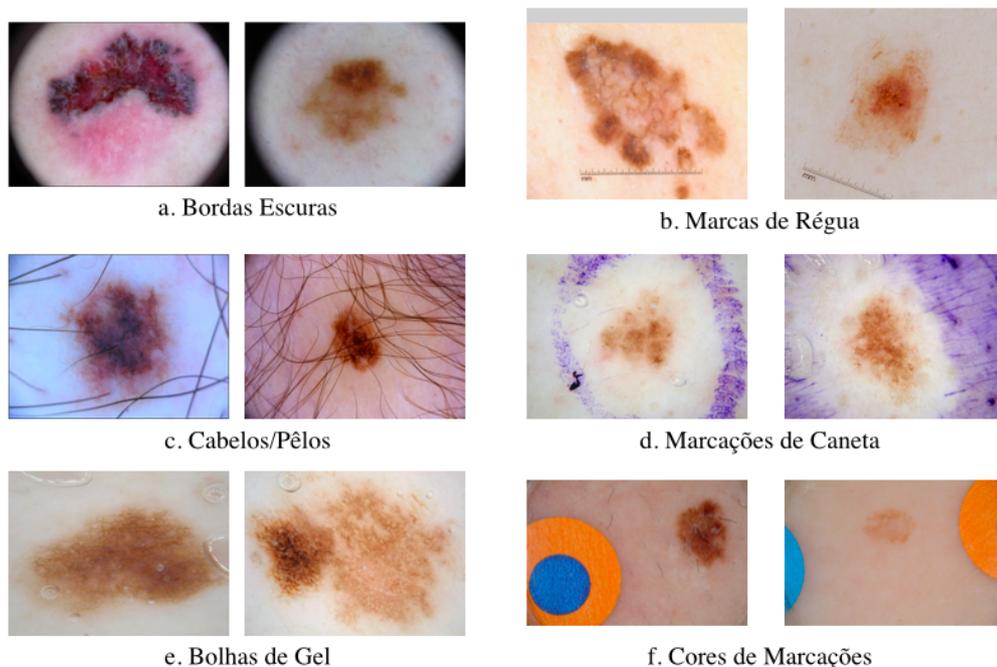
Em vista disso, e ciente de que relatórios já mostraram que vem surgindo uma escassez crescente de dermatologistas (KIMBALL; JR, 2008), no que concerne a análise das lesões de pele, aplicações computacionais visando auxiliar médicos no processamento das informações de maneira mais rápida e com resultados mais confiáveis aparecem na literatura desde 1987 (CASCINELLI et al., 1987). Com a adequada aplicação, os computadores são capazes de capturar detalhes nas lesões que podem tornar os diagnósticos mais precisos.

A automação do diagnóstico é realizada a partir da análise de imagens de exames

³ *Consensus Net Meeting on Dermoscopy* foi uma reunião virtual de consenso sobre a análise de imagens dermatoscópicas realizada no ano de 2000 que contou com a participação de 40 experientes médicos que possuíam reconhecimento demonstrado por publicações ou palestras na área de dermatoscopia de vários países (ARGENZIANO et al., 2003).

dermoscópicas da lesão de pele; essas imagens podem ser capturadas de diversas formas, pelo próprio aparelho no caso de aparelhos mais modernos, ou podem ser fotografias, inclusive utilizando *smartphones*, do que é observado a partir da lente do dermatoscópio. Ainda assim, ao mesmo tempo que isso flexibiliza a forma de captura das imagens, também faz com que não exista um padrão sobre como elas estarão disponíveis para análise, como existe, por exemplo, em outros exames como o raio-x. Devido a essa falta de padrão, as imagens podem ter os mais diferentes aspectos, ter resoluções diferentes, zoom, iluminação e na maiorias das vezes, possuem artefatos que atrapalham a automação imediata da sua avaliação, como mostrado na Figura 3.

Figura 3 – Artefatos encontrados em imagens dermatoscópicas que atrapalham a automação da detecção da lesão.



Fonte: [International Skin Imaging Collaboration \(2019\)](#).

Assim sendo, visando obter resultados mais precisos e minimizar os erros de diagnósticos causados pela dificuldade e subjetiva interpretação médica no diagnóstico do melanoma, um grande interesse na automação da análise de imagens dermatoscópicas vem se desenvolvendo nas últimas décadas ([KOROTKOV; GARCIA, 2012](#); [MASOOD; AL-JUMAILY, 2013](#); [CELEBI et al., 2015](#)). Várias ferramentas de diagnóstico computadorizado (*Computer-Aided Diagnosis - CAD*⁴) vêm sendo apresentadas na

⁴ *Computer-Aided Diagnosis (CAD)* ou *Clinical Diagnosis Support (CDS)* são ferramentas de apoio

literatura, onde são usadas diferentes abordagens de detecção de borda, seleção e extração de características, assim como algoritmos de classificação das lesões. As ferramentas de automação de diagnósticos (CAD), no que se trata do melanoma, objetivam encontrar em dada imagem de lesão de pele, a localização da lesão e estimar a probabilidade delas serem a doença (MASOOD; AL-JUMAILY, 2013).

Deep Learning é uma técnica que utiliza o conceito de aprendizado fazendo com que máquinas “aprendam” diferentes tipos de representações a partir de dados brutos. Abordagens de *deep learning* vêm mostrando resultados promissores em diversas aplicações de visão computacional, como o rastreamento e detecção de objetos e classificação de imagens. O método vêm mostrando bons resultados também em aplicações de análise de imagens médicas em diversas abordagens (SUK; SHEN, 2013; LIU et al., 2014; DHUNGEL et al., 2015), inclusive com estudos demonstrando boa performance com o uso da técnica de *deep learning* conhecida como redes neurais convolutivas (SPANHOL et al., 2016; HALICEK et al., 2017; REDA et al., 2018).

Redes neurais convolutivas, do inglês “*Convolutional Neural Networks*” (CNNs), são um tipo de abordagem de *deep learning* aplicado a imagens inspiradas no sistema visual humano que buscam aprender as características dos objetos através de uma hierarquia composta por camadas que começam aprendendo bordas simples que compõem o objeto de análise até chegar a formas mais complexas. Nela filtros treinados são aplicadas nos dados a serem analisadas extraindo conjuntos complexos de características automaticamente; a técnica surgiu nos últimos anos apresentados bom potencial de resultados em tarefas de diversas categorias (RUSSAKOVSKY et al., 2015; HE et al., 2016), inclusive no campo médico (LI et al., 2014; KAYALIBAY et al., 2017; KHVOSTIKOV et al., 2018).

CNNs têm se mostrado úteis para análises de imagens médicas de diversos tipos. No entanto, apesar de sua recente popularidade, sua projeção envolve uma grande quantidade de escolhas, muitas delas normalmente feitas por especialistas na área, que impactam fortemente no treinamento e na performance das redes. Decisões precisam ser tomadas com relação ao número e tipos de camadas, a quantidade ou tamanho de filtros, tamanho dos *batches* de execução, quantidade de épocas, taxa de aprendizagem, método de otimização e normalização, dentre outros.

médico desenvolvidas para diagnóstico, ou a previsão dele, e interpretação de imagens médicas. No caso do câncer de pele, o objetivo de desenvolver ferramentas CAD é para localizar lesões em imagens e estimar a probabilidade dela ser um melanoma (MASOOD; AL-JUMAILY, 2013).

Conhecidas essas restrições, foram buscadas na literatura algumas técnicas com potencial para aborda-las e facilitar a implementação das redes. Um desses métodos são os algoritmos evolucionários que tratam-se de estratégias baseadas no conceito de evolução natural proposto por Darwin (1859), onde os processos naturais ao qual os seres vivos são submetidos, como reprodução, mutação, competição e seleção, são vistos como responsáveis pela perpetuação da espécie, fazendo com que os seres menos adaptados não sobrevivam ou não procriem e os mais adaptados resistam e reproduzam-se. Esse conceito deu origem a técnicas de otimização conhecida como computação evolutiva e, posteriormente, surgiram estudos propondo a aplicação do método de diferentes formas, uma delas é a programação genética (PG).

Na PG, os indivíduos são vistos como programas computacionais sendo cada um deles uma possível solução para o problema analisado (KOZA, 1994). A abordagem também tem como característica a procura pelas soluções mais promissoras em todo o espaço de busca do problema. Contudo, essa busca pode acabar gerando opções de indivíduos inválidas ou, em outras palavras, incapazes de passarem pelo processo de verificação de adaptação ao ambiente produzindo um valor de aptidão (*fitness*), e, ainda assim, as formas de criação do indivíduo pode continuar estando dentro do espaço de possibilidades do problema. Esse tipo de situação tem como consequência a invalidação do indivíduo, prejudicando ou retardando a procura pela melhor possibilidade.

Um paradigma que se destaca abordando esse problema é a Programação Genética Orientada a Gramática (PGOG). A técnica introduz a utilização de gramática para que a geração de indivíduos produza apenas possibilidades válidas através da incorporação de regras na formação dos programas candidatos e na aplicação de operações genéticas a fim de produzir seres válidos no domínio desejado.

Devido a problemática envolvendo o diagnóstico do melanoma e tendo em vista que a literatura apresenta técnicas do estado da arte para classificação de imagens, buscaremos no uso de CNNs alternativas para a detecção da doença em imagens dermatoscópicas. No entanto, considerando ainda aos fatores mencionados acima quanto a grande quantidade de parâmetros para a tomada de decisões arquiteturais das CNNs, uma alternativa para melhorar esse processo e por consequência sua performance, é explorar o uso de abordagens de PG para produzir CNNs automaticamente e otimizá-las.

Neste trabalho, explora-se o uso de programação genética orientada a gramática

como forma de otimização de redes neurais convolutivas aplicadas à classificação de imagens dermatoscópicas. A eficiência da abordagem é comparada com os resultados presentes na literatura e técnicas padrão de CNNs do estado da arte. Os resultados obtidos até então se mostram competitivos com as análises de imagens dermatoscópicas feitas por dermatologistas especialistas na detecção de melanoma e melhores resultados de que as propostas conhecidas na literatura.

1.2 Problema de Pesquisa

A identificação do melanoma em seus estágios iniciais frequentemente se mostra falha, especialmente quando o profissional que analisa a lesão não é experiente (VESTERGAARD et al., 2008). Mesmo seguindo técnicas médicas propostas pela literatura, detalhes importantes da lesão podem passar despercebidos. Por conta disso, são buscados na automação da análise e diagnóstico do melanoma resultados que possam vir a auxiliar médicos e pacientes.

Infelizmente, apesar das várias técnicas para detecção do melanoma nos primeiros estágios através da simples visualização das lesões, muitos pacientes continuam a serem diagnosticados apenas nos estágios mais avançados da doença, o que resulta na necessidade de tratamentos mais severos e causas de mortes todos os anos. A necessidade de melhoria na eficiência, eficácia e precisão do diagnóstico é bastante visível. Os custos psicológicos e monetários da falha na detecção cedo da doença pode custar a vida do paciente. Também vale ressaltar que os diagnósticos falso-positivos, que não chegam a custar a vida do paciente, ainda podem levar as pessoas a serem submetidos a toda uma enorme pressão psicológica e biópsias desnecessárias.

Apesar das técnicas computadorizadas ainda não serem capazes de produzir um diagnóstico definitivo, dispensando a necessidade de presença humana no processo de diagnóstico, elas podem ser utilizadas para melhorar as decisões quanto a necessidade ou não de biópsia, auxiliando também na ampliação de detecção da doença em estágios iniciais.

As redes neurais convolutivas são tipos de arquiteturas de *deep learning*, que, por sua vez, tem origem baseada no funcionamento do cérebro humano, as redes neurais. Em termos de análise de dados, construção da visão, análise de comunicação, dentre outras

tarefas, o cérebro é uma rede neural extremamente complexa com bilhões de neurônios, camadas de ativação e um enorme poder de processamento. Isso exemplifica a complexidade necessária para criar uma rede neural convolutiva efetiva, as várias estratégias precisam ser cuidadosamente pensadas para que o modelo construído produza bons resultados.

Alguns dos parâmetros envolvidos no processo de criação de uma rede que trabalhe bem exatamente com o que é esperado são:

- Criação de um modelo eficiente: a quantidade aleatória de neurônios e camadas podem fazer com que o treinamento leve dias ou mesmo semanas para ser concluído, quando, com boas decisões, o trabalho poderia ser feito em horas ou minutos.
- Escolha da função de otimização: é responsável pela atualização dos pesos e bias da rede, dessa forma, uma má escolha pode resultar em resultados menores e mais demorados.
- Escolha da taxa de aprendizado: a escolha errada da taxa de aprendizado ou da estratégia de decaência (*decay strategy*) pode fazer com que a rede seja treinada muitas vezes mais de que a melhor possível.
- *Data augmentation* correta para o problema: o uso incorreto da técnica pode levar a criação de imagens que não fazem parte do escopo do problema ou podem causar a saturação da rede, *over-fitting*, mostrando eficácia apenas para a classificação dos dados de treinamento.

Para se ter uma ideia acerca das decisões envolvidas no processo de criação das redes, conceitos como *pooling*, amplamente utilizado atualmente, e o próprio uso da convolução no processo são resultados de imensas pesquisas na área de sensoriamento de visão (HUBEL; WIESEL, 1959; HUBEL; WIESEL, 1968; FUKUSHIMA, 1980; HUBEL; WIESEL, 2004; LECUN et al., 2015).

Por toda a complexidade envolvida, uma CNN simples em uma atividade básica pode levar horas no treinamento. Mesmo a VGGNet (SIMONYAN; ZISSERMAN, 2014), rede neural conhecida na literatura composta por apenas 16 camadas, tem 138 milhões de parâmetros que, para serem otimizados, precisam ser iterados por várias vezes, resultando em longas horas de treinamento em bases de dados grandes. O FaceNet (SCHROFF et al., 2015), uma rede de reconhecimento de faces, de acordo com seu artigo levou de 1000 a 2000 horas para ser treinada utilizando um *cluster* de CPUs.

Além do tamanho da base de dados e do modelo da rede, a configuração da máquina

onde a CNN será executada também influencia o tempo de processamento, equipamentos com GPU⁵ ou TPU⁶ e grande memória, são capazes de executar em menos tempo de que processadores padrão.

Mesmo havendo uma forte motivação de saúde pública, o estudo do diagnóstico do melanoma utilizando redes neurais também apresenta dificuldades intrínsecas que dificultam a fácil e rápida criação e execução de uma CNN com parâmetros capazes de produzir resultados eficazes. Existem muitas variáveis, parâmetros e hiper-parâmetros, envolvidas nesse processo e pequenas variações podem fazer com que a qualidade da classificação dos dados seja deteriorada. Por conta disso, a construção da rede adequada ao problema pode exigir conhecimentos de especialistas na área. A maioria dos parâmetros são mais facilmente identificáveis através do conhecimento prático e pesquisadores maduros precisam de centenas de projetos para serem treinados. Este fator motiva o estudo de técnicas que viabilizem a construção de redes sem que haja a necessidade de especialistas e que possibilitem a exploração de CNNs que, por erro ou desconhecimento humano, poderiam ser descartadas antes mesmo de serem construídas ou não serem nem imaginadas como possibilidade de estudo.

Idealmente, precisamos explorar métodos que sejam capazes de investigar várias possibilidades de solução para o problema, sem quem com isso seja necessário grandes quantidades de trabalho humano envolvido no processo, evitando, assim, possíveis falhas e a possibilidade de abrangência de um maior espaço de busca.

Apresentada toda a complexidade envolvida no processo de diagnóstico clínico e automação da classificação de imagens dermatoscópicas, além dos vários caminhos que podem guiar a criação de novos algoritmos, o problema que o presente projeto visa resolver é o seguinte:

1. Dentre as arquiteturas de CNN já conhecidas na literatura, qual delas apresenta os melhores resultados quando aplicada à base de dados dermatoscópicas?
2. Qual a melhor forma de otimização de parâmetros e hiper parâmetros de redes neurais convolutivas para a classificação de imagens dermatoscópicas de forma que

⁵ *Graphics Processing Unit* (GPU) é um processador projetado para acelerar a renderização de gráficos. Normalmente utilizado para a execução de modelos de aprendizagem de máquina, renderização de gráficos e programas que envolvem paralelização de processamento. Pode lidar com dezenas de milhares de operações por ciclo.

⁶ *Tensor Processing Unit* (TPU) é um processador apresentado pelo Google com propósito e arquitetura projetados para acelerar tarefas de *deep learning* desenvolvidas utilizando o *framework* TensorFlow capaz de processar 128 mil operações por ciclo (SATO et al., 2017)

produzam resultados competitivos?

1.3 Objetivos

1.3.1 Objetivo Geral

O trabalho proposto tem por objetivo geral propor uma gramática para a construção de arquiteturas de redes neurais convolutivas para a classificação de melanoma em imagens de lesões de pele utilizando a técnica de computação evolutiva conhecida como programação genética orientada a gramática de forma a encontrar redes com acurácia competitivas com o estado da arte sem a necessidade de interferência humana no processo de classificação. Para atingir esse objetivo geral foram propostos os objetivos específicos a seguir relacionados.

1.3.2 Objetivos Específicos

Visando atingir o objetivo geral foram definidos os seguintes objetivos específicos:

- Implementação de técnicas do estado da arte para execução com a base de dados escolhida e, posteriormente, comparação dos resultados obtidos;
- Análise dos parâmetros aos quais serão utilizados pelas redes neurais;
- Criação de uma gramática capaz de explorar de forma eficaz o espaço de busca do problema analisado;
- Implementação de algoritmo habilitado a interpretar os indivíduos gerados pela gramática, convertê-los em redes neurais e executá-los;
- Análise dos resultados obtidos comparando-os com o estado da arte.

1.4 Estrutura do Trabalho

O presente trabalho estrutura-se de forma que a análise realizada por esse estudo seja totalmente compreendida. À vista disso, informações importantes que compõem a pesquisa são todas descritas previamente à explicação da sua utilização.

O estudo está organizado em 7 capítulos da seguinte forma:

1. O capítulo 1 apresentou a contextualização do problema e a motivação para o estudo, o problema de pesquisa abordado, os objetivos e a estruturação da pesquisa.

2. No Capítulo 2 são abordados conceitos básicos fundamentais que servirão de base para o entendimento do método proposto, ao qual está dividido em quatro grandes sessões: os conceitos médicos, conceitos de imagem digital, aprendizagem de máquina e algoritmos evolucionários. Conhecimento indispensável para a compreensão da pesquisa.
3. Trabalhos relacionados do estado da arte com as contribuições dos últimos anos são apresentados no Capítulo 3.
4. No Capítulo 4 é apresentada a descrição de detalhes do método proposto, incluindo análises prévias, fluxograma do método proposto e a técnica proposta.
5. O Capítulo 5 engloba toda a metodologia utilizada para a definição dos parâmetros aplicados nos testes experimentais, com detalhes da base de dados, do ambiente experimental e a métrica de análise utilizada na verificação de qualidade da proposta.
6. Os resultados da técnica aqui proposta, incluindo comparações com outras abordagens do estado da arte, encontram-se no Capítulo 6.
7. Por fim, temos no Capítulo 7 as considerações finais sobre todo o estudo baseado nos experimentos e comparações de técnicas e os trabalhos futuros.

2 Fundamentação Teórica

Este capítulo contém uma breve descrição sobre tópicos importantes que serão abordados nesse estudo. Aqui é apresentada uma prévia do conhecimento necessário para um completo entendimento da contribuição do trabalho desenvolvido.

O capítulo está dividido sequencialmente em: [2.1](#). Definição conceitos da área médica de grande relevância para o entendimento do trabalho; [2.2](#). Fundamentos de imagens digitais, importante para o entendimento dos tipos de imagens que foram utilizados durante os testes dos algoritmos analisados; [2.3](#). Conceito de aprendizado de máquina onde veremos também as técnicas de redes neurais artificiais, *deep learning*, redes neurais convolutivas e suas arquiteturas mais conhecidas na literatura; [2.4](#). E, por fim, a técnica conhecida como algoritmos evolucionários trazendo desde bases históricas até a explicação de programação genética e programação genética orientada à gramática. Tais técnicas vão auxiliar na adequado entendimento das decisões acerca da seleção da arquitetura da rede que vai classificar as imagens analisadas.

2.1 Fundamentos Médicos

2.1.1 Dermatoscopia

A dermatoscopia é uma técnica não invasiva de diagnóstico a partir da observação da pele. Para isso, é utilizado um aparelho, parecido com um pequeno microscópio, que dá o nome à técnica, chamado dermatoscópio ([BRAUN et al., 2004](#); [Dermoscopy, 2016](#)).

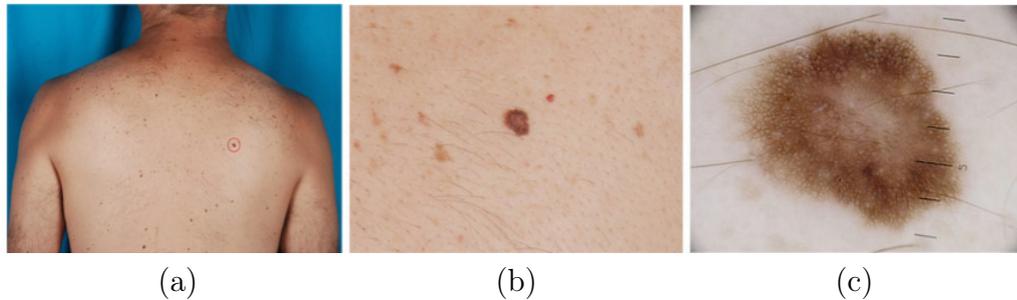
Sem a utilização da técnica, a média de detecção do melanoma é de cerca de 60% ([KITTLER et al., 2002](#)). Com o treinamento adequado, a precisão do diagnóstico pode ser aumentada utilizando a dermatoscopia, passando para cerca de 57% a 84% ([CODELLA et al., 2018](#)).

O procedimento consiste em aplicar sobre a lesão a ser examinada óleo ou fluido (óleo mineral, óleo de imersão, álcool, água) de forma a eliminar o reflexo da luz e tornar a camada mais externa da pele transparente, viabilizando a visualização das cores e estruturas morfológicas em diferentes camadas da pele (epiderme e derme) não visíveis a olho nu, em seguida a pele é observada através do dermatoscópio ([SOYER et al., 2012](#);

[Dermoscopy, 2016](#)).

A Figura 4 mostra uma comparação entre a situação de visualização de uma lesão de pele; a imagem (a) e a imagem (b) demonstram o que é possível observar quando uma lesão é vista sem o auxílio de nenhum equipamento, seria a chamada visualização a olho nu. Nesse tipo de observação não é possível obter muitos detalhes sobre as características da lesão. A imagem (c) mostra a mesma lesão das duas imagens anteriores, no entanto, nela é utilizada a técnica de dermatoscopia, dessa forma, detalhes da lesão que vêm a auxiliar no seu diagnóstico podem ser percebidos.

Figura 4 – Exemplo de visualização de uma lesão de pele. a) Imagem corporal da localização da lesão; b) Visualização macroscópica a “olho nu” da lesão; c) Visualização dermatoscópica da lesão.



Fonte: ([Pelle Digitale, 2016](#))

2.1.2 Principais Técnicas para Detecção Clínica do Melanoma

Existem diversas técnicas clínicas utilizadas para a detecção do melanoma, dentre elas, as mais estudadas e aplicadas são: a Regra do ABCD, a Análise de Padrões, o Método Menzies e a Regra dos 7 Pontos.

A Regra do ABCD foi apresentada em 1985 por [Friedman et al. \(1985\)](#), sendo a primeira técnica criada para auxiliar na diferenciação em lesão benignas e malignas. Nele é proposto a análise de quatro parâmetros na lesão – Assimetria, Borda, Cor e Diâmetro – que são pontuados de acordo com o esperado para cada um dos parâmetros, seguida da aplicação dos valores em um sistema de pontuação que aplica equação linear para determinar a probabilidade da lesão ser maligna. É uma técnica relativamente simples de se aprender e aplicar. Possui critérios simples para avaliação de lesões candidatas a uma melhor investigação, assim é utilizada globalmente na medicina ([TSAO et al., 2015](#)), adequado especialmente para médicos com pouca experiência com dermatoscopia. [Abbasi](#)

et al. (2004) propôs o acréscimo de um quinto parâmetro (evolução ou elevação) ao método, tornando-o, assim, conhecido como Regra do ABCDE (ROBINSON; TURRISI, 2006; RIGEL et al., 2005); ver Figura 5.

Figura 5 – Técnica ABCDE para diagnóstico de melanoma.



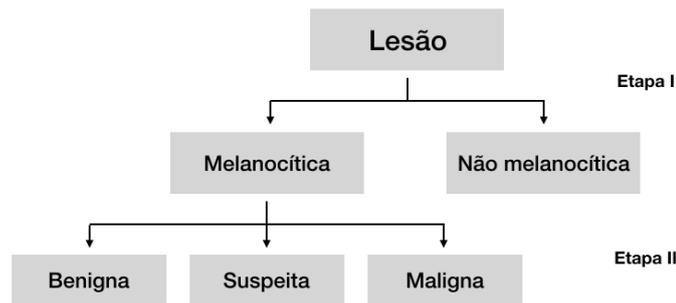
Fonte: Adaptado de Tsao et al. (2015 apud American Academy of Dermatology, 2013).

A Análise de Padrões foi proposta por Pehamberger et al. (1987) em 1987 e mais tarde atualizada na CNMD (SOYER et al., 2001), esta análise define os padrões significantes das lesões de pele (SÁEZ et al., 2014) permitindo a distinção entre padrões de crescimento benignos e malignos (PEHAMBERGER et al., 1987). A técnica é composta por duas etapas distintas, na primeira deve-se identificar se a lesão é melanocítica¹ ou não; a segunda etapa é a distinção entre lesões melanocítica benignas e melanomas (BRAUN et al., 2002). A Figura 6 mostra o fluxograma do processo; é definido ainda padrões bem específicos que devem ser observados em cada uma dessas etapas antes de se chegar a conclusão final.

O Método Menzies foi proposto em 2001 por Menzies (2001); trata-se de uma variação simplificada da Análise de Padrões que buscava diminuir a quantidade de parâmetros a serem analisados. O método é baseado em pontuação com observação

¹ São lesões com presença de rede pigmentada, glóbulos ou pontos e presença de área homogênea azul-acinzentada no nevo azul (REZZE et al., 2006).

Figura 6 – Fluxograma da técnica de Análise de Padrões.



Fonte: Adaptado de [Braun et al. \(2002\)](#).

de 11 características que são classificadas como presentes ou não ([MENZIES, 2005](#)), sendo 9 determinantes de melanoma e 2 de lesões benignas. A determinação de um potencial melanoma é dada pela presença de uma ou mais características da lista de “positivas”; já para a lesão ser considerada não cancerígena é esperado que haja a presença obrigatória de duas características definidas na técnica.

A Regra dos 7 Pontos também é um método baseado em um sistema de pontuação que pontua um conjunto de sete diferentes características na lesão ([WADHAWAN et al., 2011](#)), cada uma delas é classificada como maior ou menor grau; três delas são de maior grau e a presença delas não na lesão pontua acumula 2 pontos por cada uma; as demais características de menor grau pontuam 1 ponto cada uma; o limiar sugerido para a técnica é de 3 pontos, isso significa que lesões com 3 ou mais pontos são potenciais melanomas ([ARGENZIANO et al., 2011](#)).

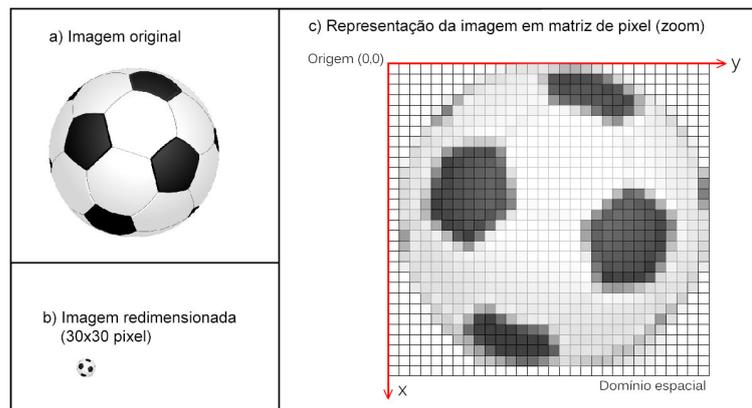
2.2 Fundamentos de Imagem Digital

2.2.1 *Pixel*

Originalmente o termo *pixel* surgiu da contração da expressão da língua inglesa “*picture element*”, sendo “*pix*” uma abreviação para “*picture*”; em tradução livre, seria um “elemento de imagem”. Trata-se do menor elemento de representação de uma imagem digital e, sendo assim, uma imagem digital é composta por uma quantidade finita desses elementos, cada um possuindo uma localização no eixo bi-dimensional x, y , como mostrado na Figura 7 ([GONZALEZ et al., 2002](#)). Esses elementos possuem informações que representam a cor, ou a intensidade da cor, correspondente à sua coordenada na imagem; o tipo de informação

em cada *pixel* varia de acordo com o tipo da imagem. Em imagens em tons de cinza, por exemplo, o *pixel* armazena a intensidade da cor, como mostrado na Figura 8. Nas coloridas do tipo RGB² o que é armazenado são as informações referentes à própria cor nos três canais de cores. *Displays* digitais projetam suas telas utilizando milhões de *pixels* organizados em linhas e colunas tão próximas entre si que chegam a parecer a olho nu que são conectadas. Dessa forma, o *pixel* é uma localização na imagem (MCCONNELL, 2005).

Figura 7 – Representação de imagem digital em *pixels*.



Fonte: França (2016).

2.2.2 Representação de Imagem Digital

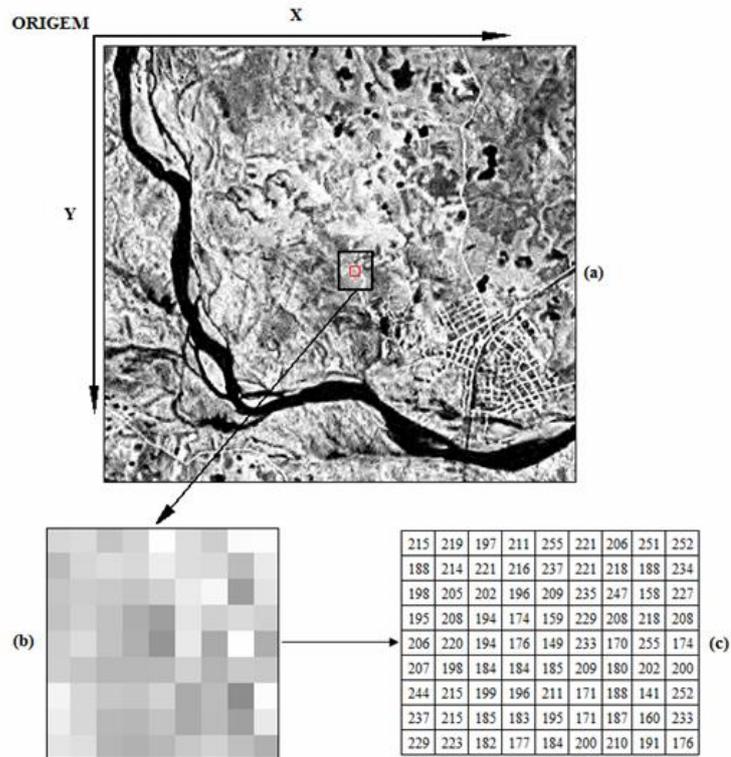
A imagem é composta por um conjunto de *pixels* organizados em forma de matriz, onde cada ponto representa a informação referente à cor daquele ponto. Essa matriz é denominada “*bit-map*” ou “*mapa de bits*”. Ela também é responsável por determinar outra característica, a resolução da imagem; isso é definido pelo número de elementos da linha horizontal da matriz (quantidade de colunas) pelo número de elementos da linha vertical (quantidade de linhas), ver exemplo na Figura 9 (MUSCI, 2016).

2.2.3 Sistema de Cores RGB

Em imagens baseadas no sistema de cores primárias: vermelho, verde e azul (*Red-Green-Blue*), cada *pixel* que compõe a imagem possui um valor para representar cada um

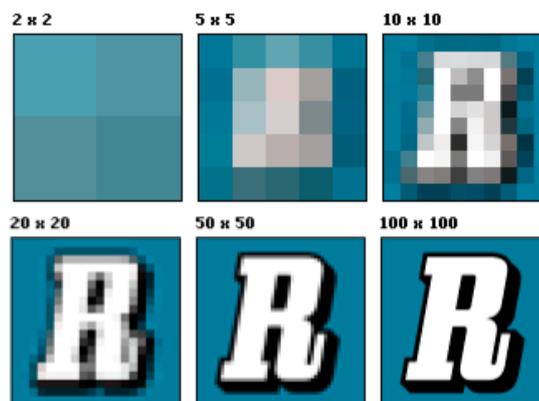
² Sistema de cores primárias vermelho, verde e azul - *Red-Green-Blue* - onde cada canal de cor é representado por um valor de 0 a 255.

Figura 8 – Representação da relação de *pixels* com as informações que cada um contém.
 a) Imagem digital; b) *Zoom* em grupo de 9×9 *pixels* da imagem (a); c) Valores de cada um dos *pixels*.



Fonte: [Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia - CEPSRM - UFRGS \(2016\)](#).

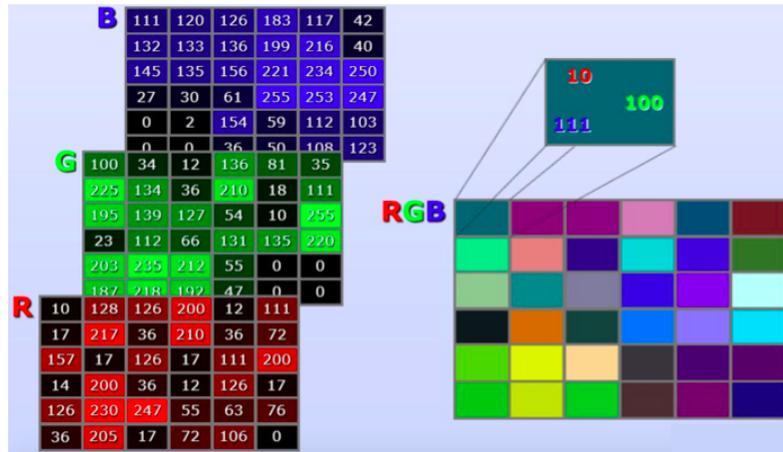
Figura 9 – Exemplo de uma mesma imagem digital representada em diferentes resoluções.



Fonte: Adaptado de [Wikimedia Commons \(2006\)](#).

dos canais de cores. Isso significa que cada um dos *pixels* contém três *bytes* com valores entre 0 (zero) e 255, um referente a cada cor primária. A cor original é formada pela composição das três cores primárias, como mostrado na Figura 10.

Figura 10 – Exemplo de composição de uma imagem do tipo RGB.



Fonte: Adaptado de [Queiroz \(2015\)](#).

2.3 Aprendizagem de Máquina

A habilidade de aprender e adquirir conhecimento é uma das características mais proeminentes da inteligência humana. Aprendizado de máquina, ou *machine learning*, é uma área da inteligência artificial voltada para o desenvolvimento de algoritmos capazes de aprender através da prática ou experiência. É buscado simular a forma humana de aprendizado para se adquirir conhecimento computacional, tornando o sistema capaz de aprender e se adaptar a mudanças no ambiente em que é executado ([MICHALSKI et al., 2013](#)).

Uma definição formal de aprendizagem de máquina amplamente conhecida foi dada por [Mitchell \(1997\)](#): “Diz-se que um programa de computador aprende pela experiência E , com respeito a algum tipo de tarefa T e performance P , se sua performance P nas tarefas em T , na forma medida por P , melhoram com a experiência E ”³ ([Mitchell \(1997\)](#), p. 2). Explicando melhor essa definição, T se refere à tarefa que vai ser executada pelo algoritmo, por exemplo o reconhecimento e classificação de letras em imagens. A medida de performance P é um método quantitativo utilizado para avaliar a capacidade do algoritmo durante o aprendizado da atividade. Isso vai permitir que nas próximas atividades o algoritmo adapte seu comportamento de acordo com sua performance. Ainda no exemplo anterior, a medida de performance poderia ser a porcentagem de letras detectadas corretamente. Por fim, a experiência E se refere à base de dados composta

³ Versão original em inglês: “a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ” ([Mitchell \(1997\)](#), p. 2).

por imagens de letras com suas devidas classificações. Retornando ao exemplo, E são os dados com suas devidas classificações que será a fonte de experiências para o aprendizado do algoritmo.

Aprendizado de máquina pode ser dividido em três categorias de acordo com o processo de aprendizado adotado (RUSSELL; NORVIG, 2009), podendo ser de aprendizado supervisionado, não supervisionado e de reforço. Para este estudo focaremos no aprendizado supervisionado e não supervisionado.

O termo ‘aprendizado supervisionado’, do inglês *supervised learning*, advém do fato de que este tipo de algoritmo funciona utilizando a ideia de ter um instrutor, ou professor, que funciona como um supervisor para o algoritmo. Na prática, o instrutor é uma base de dados com suas devidas características e classificações associadas (GOODFELLOW et al., 2016a) que vai auxiliar no aprendizado de dados futuros de classificações desconhecida pelo algoritmo. Cada instância treinada pode ser descrita em termos de uma tupla de entrada e saída (x_i, y_i) , onde x_i é o vetor de características associado a classe y_i . Em outras palavras, a técnica permite que um dado modelo preveja resultados futuros após a realização de um treinamento com base em dados do passado ou simplesmente dados conhecidos.

Os dados com as classificações auxiliam o sistema a entender características essenciais que definem cada resultado, por exemplo um modelo para identificação de carros entre diversas imagens, vai aprender as características importantes de classificação do carro, como pneus, farol, volante e assim por diante. Da mesma forma, podem ser feitas predições quanto ao preço de um imóvel dado seu tamanho, localização, quantidade de cômodos, dentre outras coisas. Assim, esse tipo de aprendizado inclui classificação, por exemplo identificar qual categoria um dado item pertence, e regressão, predições que resultam em um valor real, como o exemplo do valor do imóvel.

Ao contrário do aprendizado supervisionado, como o próprio nome diz, no aprendizado não supervisionado não é informado ao algoritmo dados de características e classificação de cada instância, cabe ao próprio algoritmo o trabalho de reconhecer sozinho relações, padrões, regularidades e/ou categorias que são importantes serem consideradas nos dados apresentados e codificá-las nas saídas. Dessa forma, possibilita a aplicação em problemas ao qual existe pouco ou nenhum conhecimento sobre o que os resultados devem apresentar (GOODFELLOW et al., 2016a). É interessante salientar que nesse tipo de abordagem não existe a ideia do “professor” que vai dar *feedback* com base nos

resultados da previsão. Exemplos de utilização dessa técnica é o agrupamento (do inglês *clustering*), que agrupa instâncias que compartilham entre si características semelhantes ou relacionadas (DUDA et al., 1973).

Veremos nas próximas sessões algumas abordagens de aprendizado de máquina importantes para este estudo.

2.3.1 Redes Neurais Artificiais

Trata-se de uma abordagem tradicional de aprendizado de máquina com o perfil de aprendizado supervisionado (BASHEER; HAJMEER, 2000). Redes neurais artificiais, ou apenas redes neurais ⁴, são sistemas computacionais inspirados nas redes neurais biológicas que visam solucionar problema da mesma forma que o cérebro humano (HAYKIN, 1994). Esses tipos de sistemas são capazes de gradualmente melhorarem suas performances através da observação de amostras da base de dados. Por exemplo, uma abordagem comum em reconhecimento de imagens é manualmente classificar imagens exemplos, tipo gatos e cachorros, e utilizar essas imagens para identificar outros gatos e cachorros da base de dados que não foram previamente classificados ou tem sua classificação desconhecida pelo algoritmo. O conhecimento para classificar os animais do exemplo não foi adquirido a priori, ou seja, de início o sistema não sabe que gatos e cachorros possuem rabos ou pêlo, dentre outras características; pelo contrário, a rede vai evoluindo seu próprio conjunto de características a partir do próprio aprendizado com sua amostragem.

O cérebro humano é composto por bilhões de neurônios que estão conectados a uma enorme quantidade de sinapses⁵, dentritos⁶ e axônios⁷. É com base na simulação da interação entre esses elementos que são simuladas redes capazes de aprender vários comportamentos (FOGEL, 1995). Basicamente, o neurônio é um tipo de célula responsável pelo recebimento e transmissão do impulso nervoso⁸ do corpo para o cérebro e vice e versa (National Cancer Institute - NIH - PubMed Health Glossary, 2018c). A comunicação entre os neurônios é dada por meio dos axônios e dentritos. Cada célula neural recebe sinais

⁴ Do inglês *Artificial Neural Networks* (ANN).

⁵ Região entre um neurônio e outra célula vizinha onde atuam os neurotransmissores, substâncias químicas produzidas pelos neurônios (National Cancer Institute - NIH - PubMed Health Glossary, 2018d).

⁶ Uma extensão do neurônio que carrega sinais elétricos das sinapses para o corpo da célula neural (National Cancer Institute - NIH - PubMed Health Glossary, 2018b).

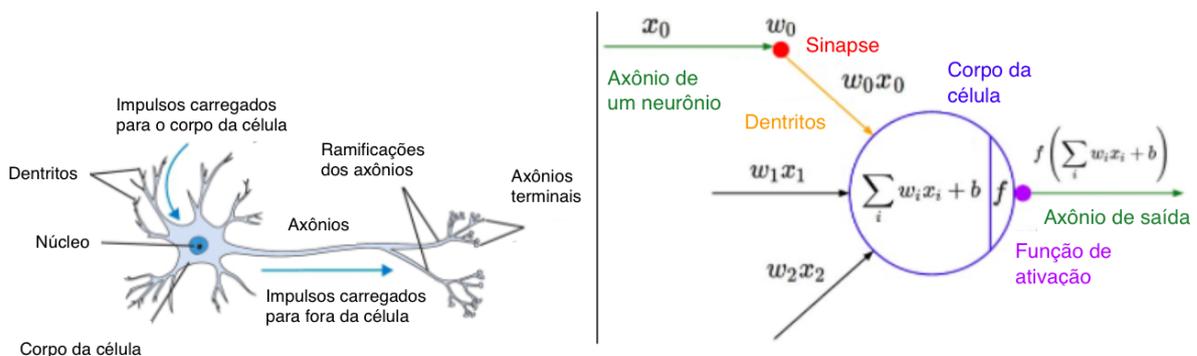
⁷ Ramificações compostas por fibras nervosas capazes de conduzir sinais elétricos para fora da célula neural (National Cancer Institute - NIH - PubMed Health Glossary, 2018a).

⁸ Corrente elétrica fraca responsável pela comunicação entre os neurônios.

de entrada através dos dendritos e produz sinais de saída pelos axônios. Os axônios se conectam por meio da sinapse aos dendritos de outros neurônios.

A Figura 11 faz uma correlação entre um neurônio real (a esquerda) e o que seria um neurônio para a computação (a direita). No modelo computacional, a conexão entre os neurônios são os pesos sinápticos ou pesos da rede e transmitem sinais de uma célula a outra adquirindo, com isso, conhecimento (HAYKIN, 1994). Em outras palavras, o sinal que segue pelo axônio, x_0 por exemplo, interage por meio da multiplicação, $x_0 \times w_0$, com os dendritos de outro neurônio baseado na força sináptica daquela sinapse, w_0 . As forças sinápticas, ou peso w , são ajustadas durante o processo de aprendizagem e seus pesos podem aumentar ou diminuir a influência e a direção – *excitatory* para pesos positivos ou *inhibitory* para pesos negativos – de um sinal na conexão. Basicamente os dendritos carregam o sinal para o corpo da célula onde são somados. Após a soma, o valor da *bias* (b) é adicionado ao resultado. Então, se o resultado for acima de um limiar pré-definido, o neurônio manda o sinal para a próxima conexão. Isso é feito executando uma função de ativação f .

Figura 11 – Estrutura básica de um neurônio biológico e sua representação em modelo computacional.



Fonte: Adaptado de (Stanford University, 2018).

A função de ativação mais simples é a função de limiarização que envia o valor 1 se o sinal for maior de que determinado limiar ou 0 caso contrário (GERSHENSON, 2003). Ainda existem outras conhecidas na literatura baseadas em funções lineares ou não. A função sigmoide é uma exemplo de não linear, seu resultado forma de uma curva em “S” com valores existentes entre 0 e 1 sendo utilizada especialmente em modelos de redes que buscam prever a probabilidade de dados de acordo com o problema analisado (HAN; MORAGA, 1995). A Tanh, ou tangente hiperbólica, também é uma função sigmoide (com

formato de “S”) mas que abrange uma valores na escala de $(-1, 1)$ (KALMAN; KWASNY, 1992), sendo utilizada principalmente para classificações que englobam duas classes. Ainda com características sigmóides, a função SoftMax é normalmente utilizada na camada de saída para a definição da classe do dado de entrada; ela transforma as saídas de cada classe em valores entre 0 e 1 e divide pela soma das saídas, produzindo um valor que pode ser interpretado como a probabilidade de classificação do dado analisado (BRIDLE, 1990a; BRIDLE, 1990b). Outra função que também vem apresentando grande sucesso em tarefas de visão computacional é a *Rectified Linear Unit* (ReLU) (DAHL et al., 2013). A técnica aplica a função não linear de ativação $f(x) = \max(0, x)$, dessa forma, para valores de x menores de que 0 a saída será 0, enquanto que para valores maiores, a saída será o próprio valor de x (NAIR; HINTON, 2010).

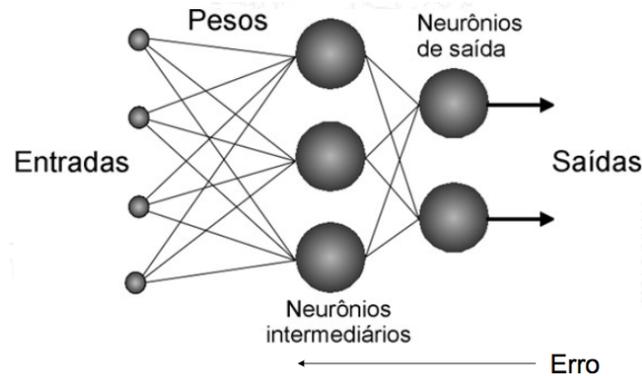
Existe ainda o processo de ajuste dos pesos da rede para minimizar erros na classificação, ele é chamado de *backpropagation* (WERBOS; JOHN, 1974; RUMELHART et al., 1986; LECUN et al., 1989); ele trata-se de um algoritmo de aprendizado supervisionado que ajusta sistematicamente o peso da rede visando diminuir a diferença entre a saída real da camada e saída esperada (RUMELHART et al., 1986). Isso é feito calculando-se inicialmente a diferença entre a saída da rede e a saída esperada e, então, a diferença é propagada de volta na rede para o ajuste dos pesos (ALPAYDIN, 2014).

As redes neurais são modeladas como um conjunto de neurônios que são conectados em um grafo acíclico⁹. Dessa forma, a saída de um neurônio pode se tornar a entrada para outro. Elas são normalmente organizadas em camadas de neurônios, cada uma podendo executar um tipo diferente de transformação nos dados de entrada recebidos, sendo mais comum as camadas totalmente conectadas, do inglês *fully-connected layer* (FC), onde os neurônios entre duas camadas adjacentes são totalmente conectados.

Na Figura 12 é possível observar a arquitetura básica de uma rede neural totalmente conectada. É mostrada uma rede com duas camadas, sendo a camada escondida ou camada intermediária composta por 3 neurônios e a camada de saída por 2 que são responsáveis por determinar o resultado do dado de entrada; para essa rede, há 4 entradas. Na mesma imagem, há ainda uma referência para a função de *backpropagation*, indicada pela seta de “erro” na parte inferior. Como a rede é um grafo acíclico, é possível ver também que há sinapses (conexões) entre as camadas, mas não há entre uma mesma camada.

⁹ Trata-se de um grafo sem ciclos, ou seja, considerando um vértice qualquer v , não existe qualquer ligação com origem e final no próprio v .

Figura 12 – Arquitetura básica de um rede neural.



Fonte: Adaptado de Bastos (2014).

A arquitetura da rede pode ser alterada conforme o problema que se é buscado solucionar e algumas outras condições, como por exemplo a quantidade de dados disponíveis para treina-la.

2.3.2 *Deep Learning*

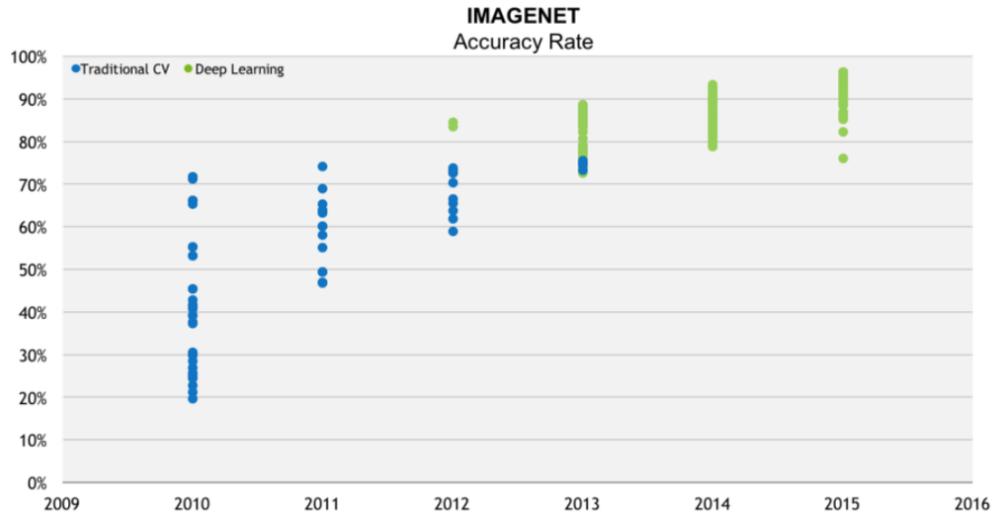
Essa é uma área específica de estudo, parte do aprendizado de máquina, que vem ganhando foco nos últimos anos (BENGIO et al., 2013) e mostrando bons resultados em visão computacional, como mostra a Figura 13. O termo “*deep learning*”, em tradução livre “aprendizado profundo”, se refere à estrutura profunda em que o cérebro é organizado, onde cada área do córtex¹⁰ é responsável por interpretar diferentes estímulos (BENGIO et al., 2015). Redes neurais desse tipo se diferenciam das ANNs devido à sua profundidade, em outras palavras, a quantidade de camadas através do qual os dados são submetidos em um processo de múltiplas etapas de reconhecimento de padrões.

Um estudo conduzido por Hubel e Wiesel (1962) para entender o sistema visual primata, observou que o cérebro inicialmente extrai as bordas, seguindo dos fragmentos, superfícies, objetos e assim por diante (HUBEL; WIESEL, 1962). Portanto, cada nível é um aprendizado de características ou representações em diferentes graus de abstração.

Deep learning busca aprender características hierárquicas construindo conceitos mais complexos partindo dos mais simples (BENGIO et al., 2009; MURPHY, 2012). O método permite que modelos computacionais compostos por múltiplas camadas aprendam representações de dados com múltiplos níveis de abstração (LECUN et al., 2015). A

¹⁰ Camada mais externa do cérebro onde fica o processamento dos neurônios (PubMed Health Glossary, 2018).

Figura 13 – Acurácia do ImageNet ao longo dos anos. Marcações azul correspondem a abordagens tradicionais de visão computacional e marcações em verde representam técnicas de *deep learning*.



Fonte: (BARKER, 2016).

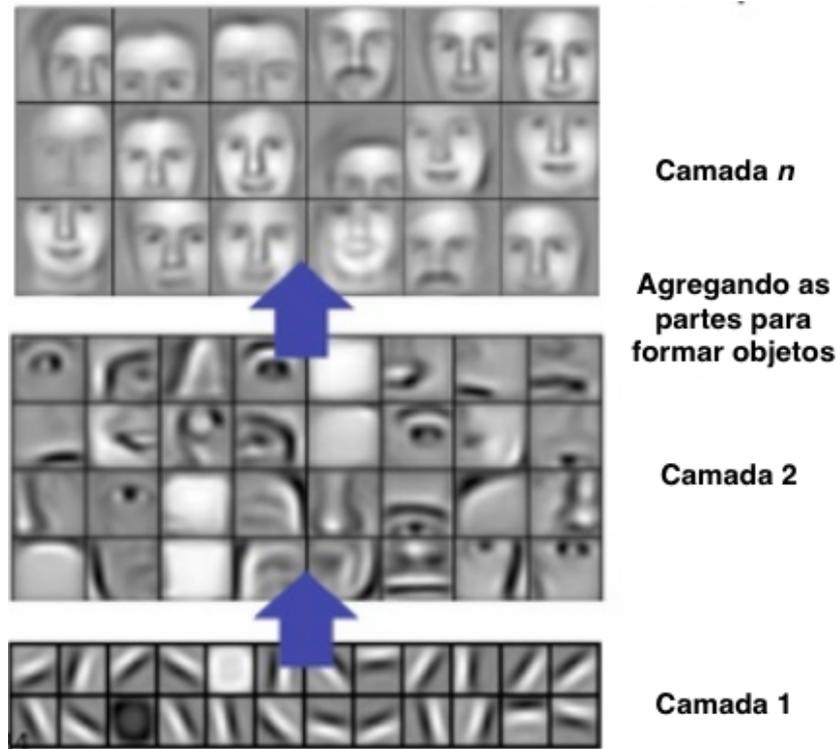
habilidade de aprender automaticamente características de vários níveis de abstração permite que o sistema aprenda funções complexas sem depender totalmente de recursos humanos.

Na técnica, características dos dados de entrada que são relevantes são amplificadas e demais características são descartadas. Cada camada treina um conjunto distinto de características baseadas na camada anterior. Quanto mais se avança na rede, mais complexas são as características que as camadas são capazes de reconhecer posto que elas agregam e recombina características de camadas anteriores. Por exemplo, uma imagem tem o formato de um matriz de valores que representam as cores do *pixel*, as características aprendidas na primeira camada normalmente representa a presença ou não de bordas em uma determinada orientação e localização na imagem; a saída da primeira camada torna-se a entrada da camada seguinte onde se detecta regiões que compartilham características em comum, os *motifs*, e assim por diante; as várias camadas são empilhadas e ao final se é possível ter um conjunto de características que determina a classe da imagem analisada.

Um dos aspectos de *deep learning* é o fato de que as camadas de características não são projetadas pela ação humana, as máquinas são capazes de aprender a partir dos próprios dados do problema (LECUN et al., 2015). A Figura 14 exemplifica como funciona o processo de reconhecimento de padrões nas camadas do *deep learning*.

Um dos problemas enfrentados pela técnica é conseguir fazer um modelo capaz

Figura 14 – Sequência de camadas do *deep learning* com o reconhecimento de características.



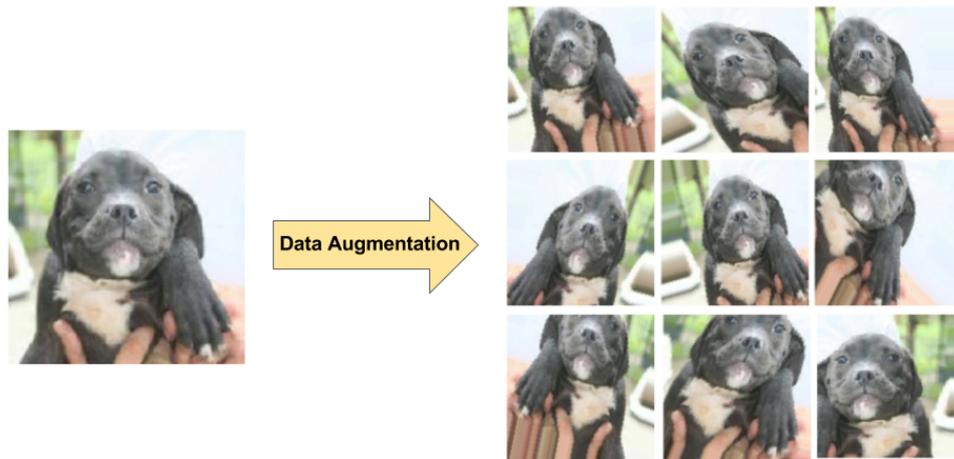
Fonte: Adaptado de (DL4J, 2018).

de apresentar boa performance não apenas nos dados de treinamento, mas também nos dados analisados. Visando abordar esse obstáculo, várias estratégias foram criadas visando a redução da taxa de erro dos dados de teste, esses métodos são conhecidos como regularização. Por definição, segundo posto por Goodfellow et al. (2016a), “regularização é qualquer modificação feita em um algoritmo de aprendizado que tem por objetivo a redução do erro de generalização mas não a taxa de erro de treinamento”.

A melhor maneira de fazer com que um algoritmo de aprendizado generalize melhor é treiná-lo com mais dados, o que vem a ser um fator complicado para muitos problemas do mundo real onde a quantidade de dados é limitada. Uma das formas de atacar essa dificuldade é criar dados falsos e adicioná-los ao conjunto de treinamento, a prática é conhecida como *data augmentation* e se mostra bastante efetiva para problemas de reconhecimento de objetos, onde é mais fácil a criação de falsos dados. Existem várias maneiras possíveis de criar imagem a partir de outra, as mais comuns são a aplicação de combinações de operações sobre a original como translação, rotação, modificação da perspectiva, achatamento, alongamento e distorção de lentes. A Figura 15 mostra um exemplo de aplicação da técnica com diferentes tipos de alterações, todas essas novas

imagens são adicionadas à base de dados de treinamento.

Figura 15 – Exemplo de aplicação da técnica de *data augmentation* na imagem original de um cachorrinho.



Fonte: ([Machine Learning Crash Course - Google Developers, 2019](#)).

A popularidade do *deep learning* levou a vários outros métodos e técnicas dessa área de pesquisa. Uma delas é a rede neural convolutiva, explicada a seguir.

2.3.3 Redes Neurais Convolutivas

Redes neurais convolutivas, em inglês *Convolutional Neural Networks* (CNNs ou ConvNets), surgiram com a evolução das pesquisas na área de *deep learning*. Enquanto redes neurais regulares não são facilmente escaláveis para trabalhar com imagens, essa técnica é especializada no processamento de dados com o formato de múltiplos *arrays*, como, por exemplo, as imagens, áudios e vídeos ou dados volumétricos ([BENGIO et al., 2015](#)). Apenas recentemente, quando esse tipo de estrutura ganhou o *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC)¹¹, que as CNNs tiveram mais destaque ([KRIZHEVSKY et al., 2012](#)). Após isso, CNNs vêm sendo aplicadas em várias áreas e são o estado da arte de diversas aplicações, especialmente em visão computacional.

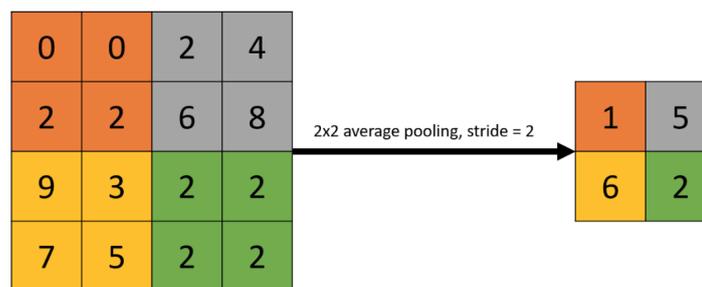
Na CNN, cada neurônio processa os dados que recebe através da aplicação de um filtro que desliza sobre cada um dos elementos que compõem o dado de entrada. O filtro é

¹¹ *ImageNet Large Scale Visual Recognition Challenge* é um concurso que avalia algoritmos de detecção de objetos e classificação de imagens em larga escala ([RUSSAKOVSKY et al., 2015](#)).

representado por uma matriz de valores chamados pesos e eles deslizam sobre a imagem da esquerda para a direita de cima a baixo produzindo valores sobre cada uma das fatias da imagem analisadas. Existem vários tipo de filtros que podem ser aplicados nesse processo e o objetivo principal é adquirir conhecimento sobre o que está sendo analisado.

Nesse contexto, o *Stride* representa o tamanho do passo que o filtro anda sobre os *pixels* do dado de entrada do neurônio; por exemplo, tendo uma imagem de tamanho 4×4 e filtro de tamanho 2×2 , se o *stride* tiver tamanho 2, isso significa que o filtro será aplicado nos *pixels* englobados pela matriz 2×2 , pulará exatamente dois *pixels* e executará o filtro novamente; o processo se repete até que toda a imagem seja percorrida, como mostra a Figura 16.

Figura 16 – Exemplo da aplicação de um filtro de tamanho 2×2 que calcula a média com um *stride* de tamanho 2.



Fonte: (ZAWADZKI, 2018).

A CNN tem seu nome originário do fato de ser utilizado operadores matemáticos de convolução¹² para extração de características da imagem de entrada. Em resumo, a rede utiliza convolução no lugar da matriz de multiplicação em pelo menos uma das camadas da rede (GOODFELLOW et al., 2016a).

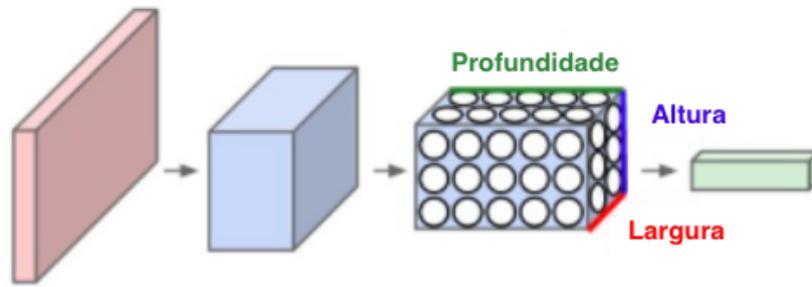
As camadas na CNN tem seus neurônios dispostos em três dimensões: largura, altura e profundidade, por exemplo uma imagem de entrada com tamanho 32×32 e 3 canais de cores possui neurônios dispostos na dimensão $32 \times 32 \times 3$, correspondendo a largura, altura e profundidade respectivamente; exemplificado na Figura 17.

De acordo com LeCun et al. (2015), existem quatro ideias chaves por trás do CNN: conexões locais, pesos compartilhados, *pooling* e o uso de várias camadas.

Ao contrário das rede neurais tradicionais, em que cada entrada é conectada com

¹² Convolução é um operador matemático linear capaz de produzir uma nova função a partir de duas outras, medindo entre as funções a soma do produto ao longo da região subentendida pela superposição do deslocamento existente entre elas (O'NEIL et al., 1963).

Figura 17 – Sequência de camadas da CNN disposta em função de três dimensões.



Fonte: Adaptado de (Stanford University, 2018).

cada um dos neurônios das camadas seguintes, a CNN explora a correlação local através da aplicação de um padrão de conexões locais entre os neurônios das camadas adjacentes. Em outras palavras, cada neurônio das camadas adjacentes só recebem entradas de um pequeno grupo local de *pixels* próximos da imagem de entrada.

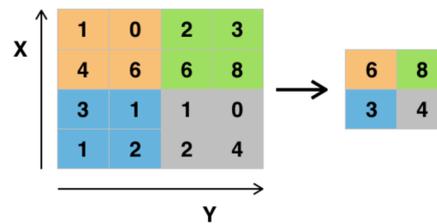
A ideia de pesos compartilhados significa que, ao invés de aprender um conjunto separado de parâmetros para cada área, a rede aprende apenas um conjunto para toda região (GOODFELLOW et al., 2016a).

Dessa forma, a operação de convolução consiste em um filtro K , ou *kernel*, sobre uma imagem I , sendo o tamanho de K menor de que I . Em cada passo, K é sobreposto em uma região de I , calculando o produto interno entre ele. O tamanho do passo do filtro é chamado *stride* e se refere ao tamanho do pulo que K irá fazer entre as regiões de I . Essa camada é chamada de camada convolucional e a sua saída é chamada de mapa de características (*feature map*). Uma camada da rede é composta por múltiplos mapas de características (HAYKIN, 1994).

Camadas de *pooling* são responsáveis por calcular a média ou máxima local e fazer subamostragens dos dados da camada anterior reduzindo a dimensão e selecionando as características mais relevantes na imagem (HAYKIN, 1994). O tipo mais comum de *pooling* utilizado é o *MaxPooling*, que é basicamente um filtro de tamanho quadrado e *stride* escolhidos de acordo com o problema, que tem como resultado de cada sub-seleção da imagem no maior valor dentro da região selecionada; a Figura mostra um exemplo de aplicação com um filtro de tamanho 2×2 e *stride* de valor 2. Outro tipo *pooling* amplamente conhecido na literatura é o de média, do inglês *average pooling*; ele segue a mesma ideia do *MaxPooling*, no entanto, ao invés de resultar no maior valor da área selecionada, ele resulta na media dos valores. CNNs consistem em camadas subsequentes

de convolução e *pooling*, não necessariamente nessa ordem. Ambas camadas compõem a etapa de extração de características.

Figura 18 – Exemplo de aplicação de um *MaxPooling*.

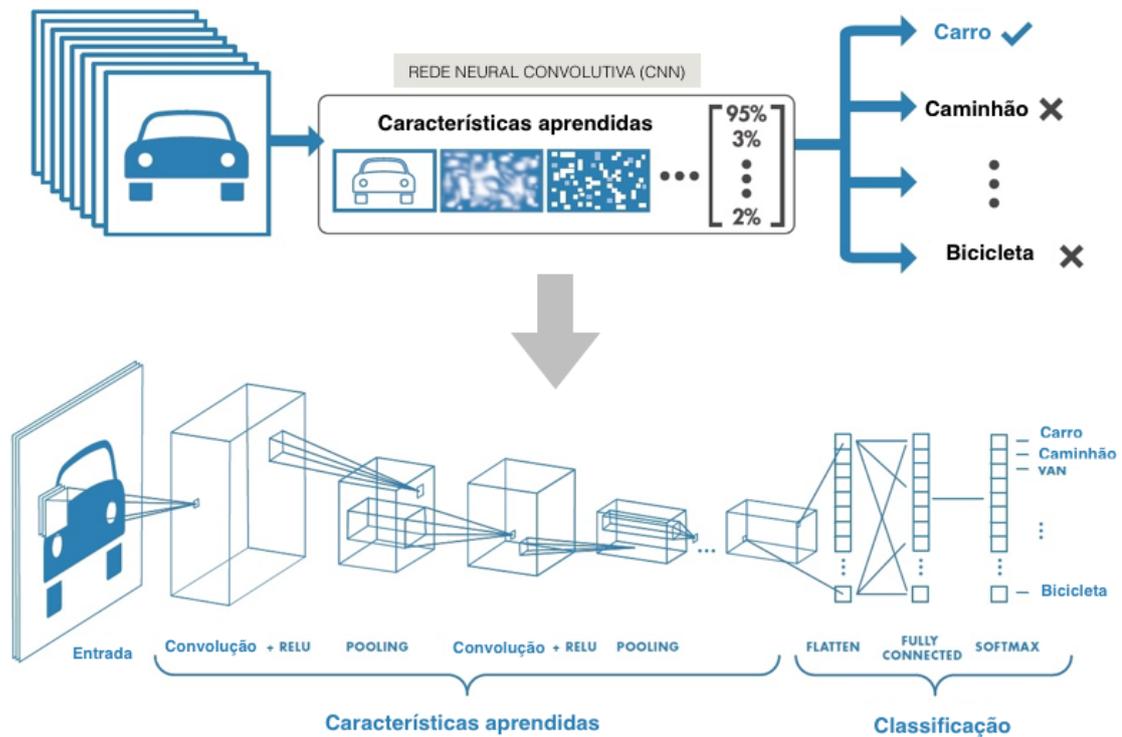


Fonte: [DeepAI \(2019\)](#).

Para explicar o processo da CNN, imaginemos uma imagem – matriz de dados – como entrada para a rede; imagens possuem grupos locais que normalmente apresentam valores correlacionados formando regiões locais distintas. É importante salientar que estatísticas locais e outros sinais são invariantes quanto a localização na imagem; isso significa que uma região referente a um objeto na imagem pode, por exemplo, aparecer em qualquer outra parte com as mesmas características. Dessa forma, essas áreas em diferentes regiões compartilham o mesmo peso e possuem o mesmo padrão em partes diferentes da matriz. A camada convolucional tem a função de detectar características da camada anterior, enquanto uma camada de *pooling* combina características que são semanticamente similares. Dessa forma, combinações locais de borda produzem áreas de interesse (*motifs*), as áreas são combinadas em partes e essas partes formam os objetos ([BENGIO et al., 2015](#)). Em outras palavras, as primeiras camadas da rede são responsáveis por aprender padrões de borda da imagem, as camadas intermediárias aprendem demais representações mais complexas e, por fim, a última camada é responsável pela classificação final do que é analisado. Ver [Figura 19](#).

Com a evolução dos estudos sobre CNN, várias técnicas foram surgindo, cada uma com suas especificidades, no entanto, compartilhando a ideia de aplicar treinamento em conjunto com um algoritmo de *back-propagation*. As características de cada um dos modelos está basicamente na sua arquitetura proposta. Veremos a seguir alguns modelos de arquitetura abordadas nesse estudo.

Figura 19 – Funcionamento de uma rede CNN.



Fonte: Adaptado de [MathWorks® \(2018\)](#).

2.3.3.1 LeNet

LeNet é uma técnica proposta por [LeCun et al. \(1998\)](#) que tinha o objetivo inicial de ser aplicado na automatização da detecção de dígitos escritos a mão em cheques de bancos no Estados Unidos. O algoritmo representou um importante marco nos estudos de reconhecimento numérico, que anteriormente era realizado através de aprendizado de máquina sendo necessário informar as características de cada número, e por se tratar de uma CNN, o LeNet5 consegue aprender sozinho o que é importante em cada imagem.

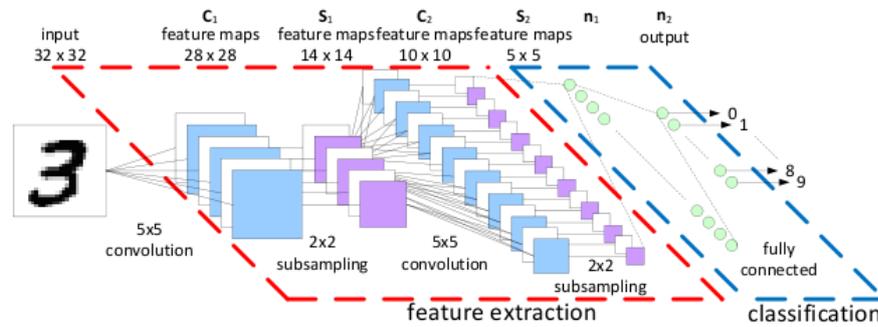
O algoritmo é bem simples, composto por 7 camadas sendo 3 delas convolutivas, 2 de *pooling* e 1 camadas totalmente conectada seguida pela camada de saída. Nas camadas convolutivas é aplicado filtro de 5×5 com *stride* de valor 1; as camadas de *pooling* são do tipo *average* com tamanho 2×2 ; é aplicada a função de ativação sigmóide Tanh ([LECUN et al., 1998](#)). O algoritmo foi capaz de alcançar uma taxa de erro menor que 1% na base de dados MNIST¹³, resultado muito próximo do estado de arte na época. Estudos posteriores foram surgindo e houveram pequenas alterações na técnica inicialmente proposta surgindo variações conhecidas como LeNet-1, LeNet-4, LeNet-5 e Boosted LeNet-4 ([LECUN et al.,](#)

¹³ Base de dados pública amplamente utilizada composta por imagens de números contendo um conjunto com 60 mil imagens de treino e 10 mil de validação ([LECUN et al., 1998](#)).

1995).

A Figura 20 mostra a arquitetura do LeNet-5 quando aplicado em uma base de dados com imagens do tamanho 32×32 , como foi o caso do exemplo usado pelos próprios autores.

Figura 20 – Arquitetura do LeNet-5.



Fonte: [Deep Learning Tutorials](#) (2015 apud [LECUN et al., 1998](#)).

2.3.3.2 GoogLeNet

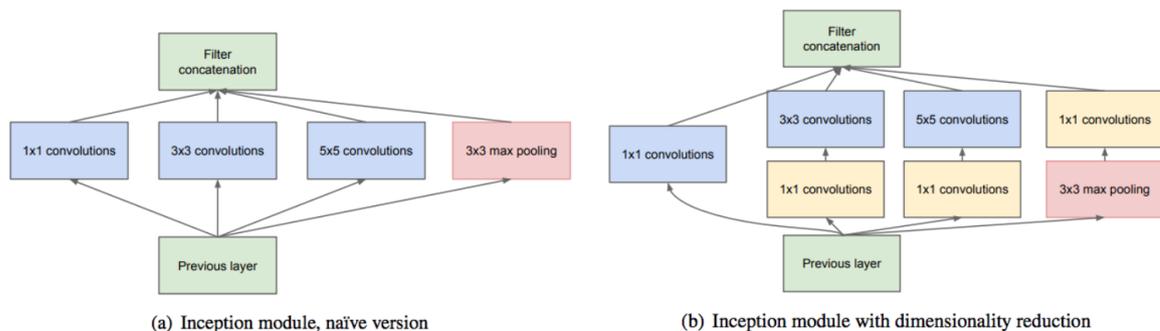
Algoritmo vencedor do ILSVRC 2014 ([RUSSAKOVSKY et al., 2015](#)) na categoria de classificação e também conhecido como *Inception v1* ([SZEGEDY et al., 2015](#)), alcançou uma taxa de erro que ficou entre as 5 melhores demonstrando uma performance de classificação muito próxima da humana. A técnica inspirada no LeNet – sendo inclusive seu nome uma forma de homenagem ao método proposto por [LeCun et al. \(1998\)](#) – traz como diferencial a implementação de um elemento que ficou conhecido como módulo *inception* ou camada de inception.

A GoogLeNet completa é composta por 22 camadas convolutivas baseadas na ideia de utilização de diversas pequenas convoluções visando diminuir drasticamente o número de parâmetros da rede. Se visa aplicar cobertura em uma grande área mantendo também uma boa resolução capaz de coletar pequenas informações nas imagens. A rede aplica convolução em paralelo com tamanhos de filtros diferentes, do mais preciso 1×1 para o mais amplo 5×5 ; se é esperado que essa mistura de filtros consiga lidar melhor com objetos de variadas escalas dado que todos os filtros da *inception* adquirem conhecimento.

A Figura 21 mostra dois módulos propostos pelo GoogLeNet. As convoluções e o *pooling* são aplicados simultaneamente, tendo o mesmo dado de entrada, e se juntam

formando a saída da camada; por conta da quantidade de operações a serem feitas, o custo computacional do módulo naïve (a) é mais alto e para diminuir isso foi proposto o segundo tipo de módulo (b); nele se é adicionado convolutivas com filtro de tamanho 1×1 visando a redução da dimensão dos dados uma vez que sem isso a quantidade de operações aumenta consideravelmente. Basicamente *inception* é um bloco da rede onde são feitas várias convoluções com diferentes filtros; após o treinamento, a rede é capaz de decidir qual tipo de camada da convolução produz o melhor resultado. Isso faz com que não seja necessário escolha humana nas decisões quanto as camadas a serem aplicadas na rede.

Figura 21 – Módulos de *inception* do GoogLeNet.

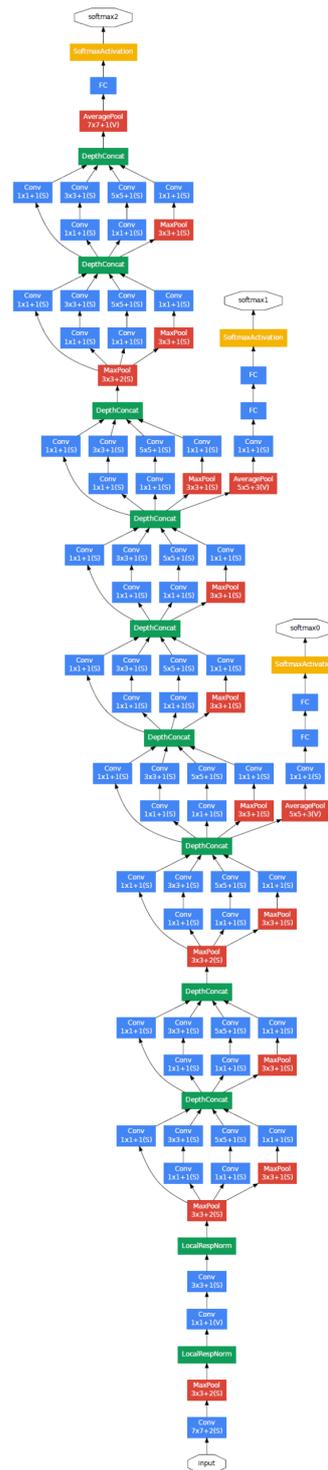


Fonte: (SZEGEDY et al., 2015)

A rede também apresenta a utilização de dois classificadores auxiliares durante o treinamento; é aplicado *SoftMax* (GOODFELLOW et al., 2016b) nos dados de saída de dois dos módulos de *inception* para calcular a *loss* auxiliar de cada classe. O valor total da função de *loss* é uma função peso da soma dos valores auxiliares com o valor da real da rede – o peso proposto pelo artigo foi de 0,3.

A arquitetura da rede é composta por módulos com redução de dimensão – Figura 22(b) – de *inception*, convoluções, operações de concatenação e classificadores auxiliares. São utilizados 9 módulos linearmente empilhados formando 22 camadas de profundidade, 27 incluindo as camadas de *pooling*, seguido de um *global average pooling* no final do último módulo de *inception* (SZEGEDY et al., 2015), Figura 22. Assim como o LeNet, o GoogLeNet possui variações conhecidas como *Inception v1*, *Inception v2*, *Inception v3*, *Inception v4* e *Inception-ResNet*.

Figura 22 – Arquitetura do GoogLeNet.



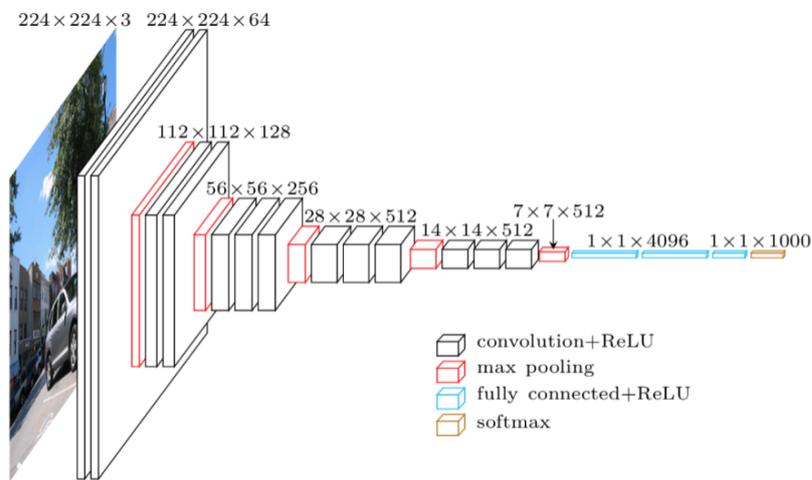
Fonte: (SZEGEDY et al., 2015)

2.3.3.3 VGGNet

Proposto por Simonyan e Zisserman (2014) também durante o ILSVRC 2014 (RUSSAKOVSKY et al., 2015) vencendo a categoria de localização e segundo lugar na

categoria de classificação. É uma arquitetura simples mas profunda composta por 16 a 19 camadas convolutivas com filtros de tamanho pequeno, 3×3 , empilhadas sequencialmente com a aplicação de *MaxPooling* para a redução do volume e duas camadas totalmente conectadas seguidas por *SoftMax* como função de ativação. A rede apresenta uma camada de pré-processamento que utiliza os valores dos *pixels* da imagem em RGB e subtrai a média dos valores. A Figura 23 mostra a arquitetura da VGG-16. Existem variações da técnica conhecidas como VGG-11, VGG-11 (LRN), VGG-13, VGG-16 (Conv1), VGG-16 e VGG-19.

Figura 23 – Arquitetura do VGGNet com 16 camadas.



Fonte: (FROSSARD, 2016).

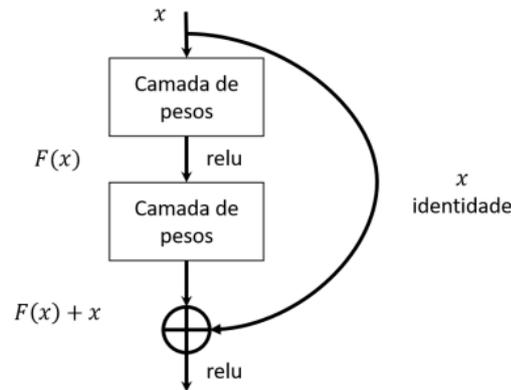
2.3.3.4 ResNet

Depois AlexNet ter vencido o ILSVRC 2012, o *Deep Residual Network* (HE et al., 2016), ou simplesmente Resnet como ficou conhecido, foi um dos trabalhos de visão computacional que mais chamou atenção nos últimos anos. A técnica torna possível o treinamento de centenas ou até milhares de camadas e ainda assim alcançar um desempenho competitivo. A rede venceu o ILSVRC 2015 com *top-5 erro* de 3,6%, superior a taxa de erro de humana que fica por volta de 5 a 10%.

A rede é composta por 152 camadas compostas por blocos residuais onde cada entrada x é submetida a operações de convolução-relu-convolução e seu resultado é adicionado à entrada original x . Como o próprio nome da rede indica, a rede introduz a técnica de aprendizado residual; na Figura 24 é mostrado a representação do bloco

residual.

Figura 24 – Bloco residual utilizados na composição do ResNet.



Fonte: Traduzido de [He et al. \(2016\)](#).

Ao longo dos anos foi surgindo a necessidade de estudo de redes cada vez mais e mais profundas para solução de problemas mais complexos e para a melhoria da acurácia de classificação. No entanto, quanto mais profunda a rede, mais difícil se torna o treinamento e a acurácia começa a saturar, fazendo com que sejam apresentados resultados mais baixos. Dessa forma, o aprendizado residual busca abordar ambos os problemas.

No aprendizado residual, ao invés de aprender algumas características como as redes no geral, os resíduos é que são aprendidos; o resíduo pode ser entendido como a subtração de características aprendidas dos dados de entrada da camada, isso é feito através da conexão direta da camada com o $f(x)$, função ao qual aplica as operações de convolução-relu-convolução, como mostrado na Figura 24. Essa forma de treinamento da rede é mais fácil de que o treinamento comum e também é capaz de mitigar o problema da degradação da acurácia.

2.3.3.5 Transferência de Conhecimento

Em *deep learning* existia, logo no início dos estudos na área, a crença de que os dados utilizados para treinamento da rede devem ser do mesmo tipo da classificação esperada e devem ser distribuídos igualmente, por exemplo, quando se quer classificar gatos e cachorros, se utiliza uma base de treinamento composta por variações de imagens de gatos e cachorros apenas. No entanto, quando os algoritmos são submetidos a problemas reais, essa separação de classes pode não ocorrer ou ainda algumas áreas de interesse

podem ter quantidade de dados insuficientes para o treinamento ou terem sua obtenção muito custosas, seguindo a lógica do exemplo acima, na prática é possível que a rede receba imagens de outros tipos de animais como onças e se perceba que isso ajuda a reconhecer melhor os gatos e cachorros, ou que não se tenha uma base de gatos e cachorros que permita o treinamento eficaz da rede. Nesses casos, pode-se recorrer a transferência de conhecimentos entre redes. A técnica também tem raízes na forma de aprendizado humana, sendo motivada pelo fato de que as pessoas podem utilizando a inteligência aplicar conhecimentos aprendidos previamente para resolver novos problemas de forma mais rápida ou com melhores soluções. No *deep learning*, ela possibilita a melhoria da performance, redução de custo do treinamento da rede e da necessidade de grandes massas de dados (WEISS et al., 2016).

Por definição, dado um domínio fonte \mathcal{D}_F e uma tarefa de aprendizado \mathcal{T}_F , um domínio alvo \mathcal{D}_A e uma tarefa de aprendizado \mathcal{T}_A , a transferência de conhecimento busca melhorar a performance de conhecimento do alvo utilizando conhecimento adquirido por \mathcal{D}_F e \mathcal{T}_F onde $\mathcal{D}_F \neq \mathcal{D}_A$ ou $\mathcal{T}_F \neq \mathcal{T}_A$ (PAN; YANG, 2010).

A literatura apresenta a aplicação da técnica em diversas áreas (ZHOU et al., 2014; HAREL; MANNOR, 2010; DUAN et al., 2012); nos estudos de redes neurais convolutivas voltados para imagens médicas, temos aplicações que se utilizam de CNNs pré-treinadas com imagens de diversas classificações, inclusive não médicas, aplicando a tarefas do domínio da saúde (SHIN et al., 2016; HUYNH et al., 2016; ESTEVA et al., 2017).

2.4 Algoritmos Evolucionários

2.4.1 Princípios da Evolução Natural

Na segunda metade do século XIX, o cientista Darwin (1859) propôs o que é aceito até hoje como a teoria biológica mais importante para explicação da origem e evolução humana. Segundo a teoria, que ficou conhecida como evolucionista ou teoria da evolução, a história da maioria dos seres vivos, sejam eles humanos, vegetais e/ou animais, é decorrente de um lento processo de alterações que agem sobre populações e espécies (DARWIN, 1859). Dessa forma, todos são originários de seres mais simples que foram se modificando e, conseqüentemente, evoluindo.

Segundo Darwin (1859), os processos pelos quais os seres vivos são naturalmente

submetidos são a reprodução, mutação, competição e seleção. A reprodução é o processo conhecido para a perpetuação da espécie; a mutação se refere as alterações de forma espontânea na estrutura dos cromossomos introduzindo e mantendo a diversidade genérica dos seres; a competição e a seleção surgem como consequência do aumento da população em um espaço finito onde os seres competem pelos recursos presentes, podendo a seleção ser vista como um guia para a manutenção ou incremento da habilidade de sobreviver e reproduzir no ambiente. O resultado da interação entre esses processos na população é o que garante a evolução da espécie geração após geração.

Em resumo, a natureza, com seus recursos, solucionou o problema de sobrevivência das espécies; em outras palavras, a natureza otimizou seus mecanismos para resolver o problema. Nos anos de 1950, começaram a surgir algoritmos inspirados nas ideias advindas desse princípio. Nessa época, autores como [Fraser \(1957\)](#) e [Friedberg \(1958\)](#) apresentaram estudos sobre o uso de processos evolucionários para solução de problemas computacionais; foi então que mais tarde entre 1950 e 1960 os algoritmos com perfis mais evolutivos começaram a surgir ([JONG et al., 1997](#)).

2.4.2 Computação Evolutiva

Uma das formas algorítmicas de aplicação da biologia seguindo o conceito de evolução natural vem tendo um crescimento considerável ao longo dos anos, é a chamada computação evolucionária (CE). Abordagens desse tipo são apresentadas na literatura desde o final dos anos 50 ([BOX, 1957](#); [BOX](#); [HUNTER, 1959](#)). Ideologicamente, são criados algoritmos com perfil evolucionário que produzem uma população de possíveis soluções para um dado problema e essas soluções vão sendo modificadas ao longo da execução do algoritmo até que se alcance um resultado esperado, próximo do esperado ou as possibilidades de tentativas sejam exauridas. Seguindo o conceito de evolução natural, quando se é gerado uma possível solução que é considerada ruim, ela é então descartada e a busca continua visando a obtenção de melhores soluções. Por sua característica evolucionária, esses tipos de algoritmos são vistos como otimizadores de problemas, buscando encontrar a melhor solução ou a mais otimizada.

A CE se utiliza de conceitos biológicos se apropriando da metáfora da evolução natural ([PENA-REYES; SIPPER, 2000](#)), sendo eles:

- Indivíduo: o próprio indivíduo em suas características externas ou físicas, na biologia conhecido como fenótipo;
- Gene: ou códon; é cada um dos elementos mais simples que compõem as características dos indivíduos;
- Cromossomos: elemento detector das informações genéticas do indivíduo, o genótipo. É formado por um conjunto de genes;
- População: conjunto de indivíduos ou, em outras palavras, um conjunto de soluções candidatas;
- Reprodução ou recombinação: forma de geração de novos indivíduos;
- Mutação: alteração nas características genéticas do indivíduo;
- Seleção: processo baseado no conceito de seleção natural introduzido por Darwin ([DARWIN, 1999](#)) no qual os indivíduos mais adaptados ao ambiente sobrevivem e se reproduzem, enquanto que os mais fracos tendem a desaparecer.

Temos ainda o conceito de meio ambiente que, ainda seguindo a metáfora, seria o espaço de buscas. Nesse ambiente, cada indivíduo corresponde a uma possível solução para o problema e o seu grau de adaptação é medido por uma métrica chamada *fitness*. O cálculo do *fitness* varia de acordo com o que se deseja ser resolvido ([BANZHAF et al., 1998](#)). O conjunto de indivíduos produzido é chamado de população. Cada ser possui seu genoma, ou genótipo, onde estão codificadas suas características. Aplicando operadores genéticos na população, como a reprodução, mutação e seleção, novos materiais genéticos são introduzidos formando novos seres pelo qual é esperado que sejam cada vez melhores, seguindo a própria ideia da evolução natural. Dessa forma, evolução é por si só um processo de otimização onde todo o processo se dá na busca por melhores soluções ([MAYR, 1988](#)).

O Algoritmo 1 mostra o pseudocódigo de um algoritmo evolucionário. O início do processo se dá com a inicialização da população, linha 2. Nesse passo é necessário uma função que produza um ser adequado ao problema. A população inicial tem seu *fitness* avaliado através de uma função adequada ao problema que seja capaz de determinar um valor para a adaptação do indivíduo ao problema analisado. Em seguida o melhor indivíduo até então é salvo, linha 4. Executada essa parte inicial, começa então o processo de escolha dos pais, essa função pode variar de acordo com a abordagem de seleção escolhida. Cada par de pais produz um par de filhos herdeiros de seus códigos genéticos; os filhos sofrem cada um o processo de mutação, também de acordo com o tipo

escolhido para o algoritmo, e as etapas iniciais de avaliação à adaptação e armazenamento do indivíduo mais adaptado se repetem até que os filhos produzidos são considerados a nova população a ser analisada. O processo se repete até que uma determinada condição de parada seja alcançada, normalmente um número máximo de iterações ou o alcance da melhor solução.

Algoritmo 1 Pseudocódigo de um algoritmo evolucionário.

```

1: função EC(PopulationSize)
2:   Population ← initializePopulation(PopulationSize)
3:   evaluatePopulation(Population)
4:   bestSolution ← getBestSolution(population)
5:   enquanto !stopCondition() faça
6:     Parents ← SelectParents(Population, PopulationSize)
7:     Children ← {}
8:     para cada Parent1, Parent2 ∈ Parents faça
9:       Children1, Children2 ← Crossover(Parent1, Parent2)
10:      Children ← Mutate(Children1)
11:      Children ← Mutate(Children2)
12:     fim para cada
13:     evaluatePopulation(Children)
14:     bestChildren ← getBestSolution(Children)
15:     bestSolution ← getBestSolution(bestChildren, bestSolution)
16:     Population ← Replace(Population, Children)
17:   fim enquanto
18:   devolve bestSolution
19: fim função

```

A função *fitness* é responsável por calcular o grau de adaptação do indivíduo ao ambiente em que pertence e varia de acordo com o problema a ser solucionado. Tomando o problema de geração de programas corretos para um dado fim, um exemplo de função de *fitness* seria a quantização do número de erros. (KOZA, 1994).

O *fitness*, na maioria das vezes, é utilizado também durante a etapa de seleção, favorecendo com que os mais adaptados sejam selecionados para a reprodução. No entanto, isso não é regra geral visto que algumas técnicas de seleção também consideram indivíduos com diversos valores de *fitness* objetivando manter a diversidade na população. Um desses métodos é a roleta, onde todos os indivíduos são considerados para a seleção. Na roleta, cada *fitness* é representado de acordo com sua proporção, como mostrado na Figura 25. Isso faz com que os indivíduos com maiores valores de *fitness*, em outras palavras, maior adaptação ao ambiente, apresentam maior probabilidade de serem escolhidos; no caso

de exemplo da imagem, o S_2 tem probabilidade de 47% para ser escolhido. Essa técnica mantém a ideia uma roleta, então, quando formada a roleta com todas as partes dos indivíduos, ela é girada por um determinado número de vezes de acordo com o tamanho da população e assim são escolhidos os indivíduos para as próximas etapas, podendo haver repetições.

Figura 25 – Indivíduos de uma população com seus devidos valores de *fitness* e a representação da população na roleta.



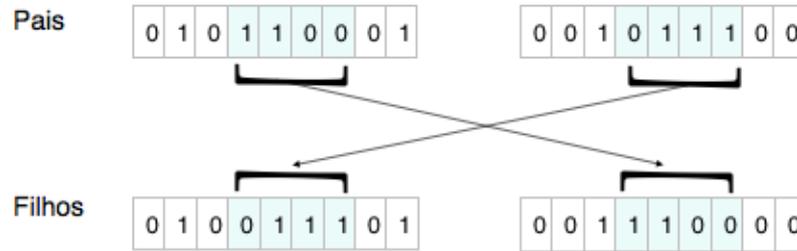
Fonte: Adaptado de [Ashlock \(2006\)](#).

Vale salientar que a seleção por si só não é capaz de introduzir novos indivíduos à população, isso fica a cargo dos operadores de reprodução e da mutação.

A reprodução faz com que o processo de busca siga por regiões mais promissoras do espaço, já que, normalmente, são utilizados dois indivíduos bons para gerar outro que se espera ser melhor. Nele são gerados filhos que carregam características de seus pais. Nesse processo, são escolhidos dois pais, que podem ou não ter tamanho e forma diferentes, e a partir deles são gerados os filhos. Existem vários tipos de métodos de reprodução, que variam de acordo com a forma de representação adotada para o problema. A Figura 26 mostra como funciona o processo de reprodução conhecido como dois pontos (do inglês, *two-point crossover*) aplicado a indivíduos representados por *string* binária. Nele, são escolhidos dois pontos aleatórios onde os valores entre esses pontos serão replicados nos filhos. Inicialmente é gerado um indivíduo que é a cópia de cada um dos pais e neles são inserido uma parte do gene do outro pai. Por consequência, cada um dos filhos carrega características de ambos os pais ([ASHLOCK, 2006](#)).

Já a mutação garante que sejam feitas alterações nos genes dos indivíduos gerados visando evitar uma convergência prematura para um ponto que pode ser o melhor local, mas não chega a ser o melhor global. O processo de mutação em si é a aplicação da

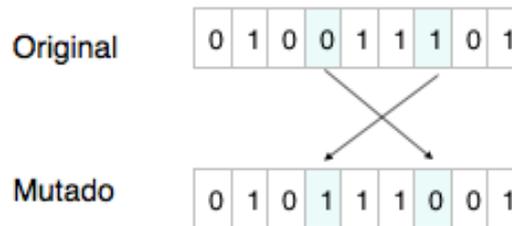
Figura 26 – Operação de reprodução de dois pontos, ou *two-point crossover*.



Fonte: Adaptado de [Ashlock \(2006\)](#).

alteração em um ou mais nucleotídeos nos genes no cromossomo do indivíduo. Existem vários tipos de mutação. A Figura 27 mostra um exemplo de mutação em um *string* binária, onde são escolhidos dois pontos aleatórios no cromossomo e os valores encontrados são substituídos entre os mesmos pontos, gerando assim um indivíduo que sofreu mutação.

Figura 27 – Exemplo de uma operação de mutação.



Fonte: Adaptado de [Ashlock \(2006\)](#).

Após as etapas de reprodução e mutação, os novos indivíduos são introduzidos a população. Cada uma dessas etapas possuem variadas possibilidades de implementação, o que determina cada uma delas é o tipo de abordagem de CE escolhido.

Um dos problemas enfrentados pela CE é como fazer a representação dos indivíduos para que descrevam todas as possíveis soluções candidatas do problema analisado ([MICHALEWICZ, 1996](#)). As representações devem ser completas e ao mesmo tempo simples para que possam ser buscadas em um tempo razoável. Algumas abordagens utilizam *string*, vetor de valores binários, árvores, dentre outros tipos de representação.

Utilizando esse princípio, foram surgindo vários tipos de algoritmos evolucionários, dentre eles a programação genética, os algoritmos genéticos, as estratégias evolucionárias e a programação evolucionária. Cada qual com características específicas mas compartilhando entre si o conceito evolucionário ([BACK et al., 1997](#)). Cada um desses algoritmos possuem importantes parâmetros que são basicamente o que caracterizam cada um deles, esses

parâmetros são a forma de representação, o mecanismo de seleção, forma de cruzamento, mutação, tamanho e variação da população e do genoma. Conforme o interesse desse estudo, apresentaremos com mais detalhes a programação genética.

2.4.2.1 Programação Genética

Técnica de CE, proposta por Koza (1990), que busca codificar as representações dos problemas como programas computacionais. Dessa forma, cada um dos indivíduos da população são programas que, quando executados, são por si próprios as soluções candidatas. Essa técnica é conhecida como programação genética (PG) (KOZA, 1994). Na prática, esses programas são evoluídos ao longo de várias gerações até que se alcance um determinado ponto de parada.

Os indivíduos na PG são representados por programas que são composições hierárquicas de funções que representam o domínio de um dado problema (KOZA, 1994). Para viabilizar que a aplicação das alterações propostas pela recombinação e mutação produzam indivíduos válidos, os programas são representados por árvore sintáticas¹⁴, do inglês *parse trees* ou *parsing tree*. O conjunto de nós internos não terminais da árvore são compostos por funções (F) definidas pelo usuário, enquanto que os nós terminais (T) são normalmente variáveis ou constantes. Por exemplo, consideremos o conjunto $F = +, -$ e $T = x, 3, z, 2$, possíveis árvores que respeitem os conjuntos F e T são mostradas na Figura 28. Representando as expressões em LISP¹⁵, temos a primeira como $(x + 3)$ e a seguinte como $(2 + (3 - z))$.

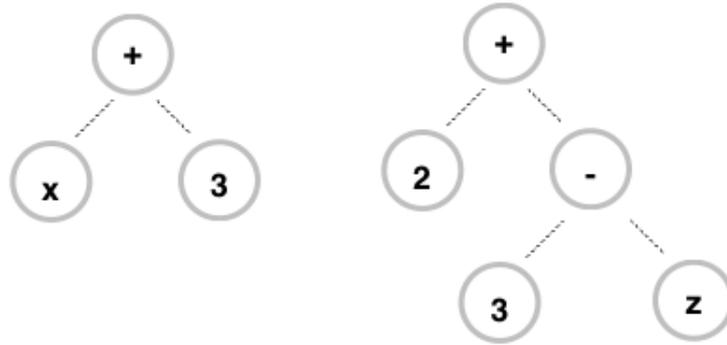
Para a PG, a busca de soluções de um dado problema é vista como a busca no espaço de todas as possíveis combinações de funções construídas com as funções F e os terminais T que formam a delimitação do problema.

A Figura 29 proposta por Koza (1994) mostra o fluxograma de um algoritmo de programação genética, onde M é o tamanho da população, i o número de indivíduos, PC a probabilidade de cruzamento e PR a probabilidade de reprodução.

¹⁴ Também conhecida como árvore de análise sintática, árvore de derivação (do inglês, *derivation tree*) ou árvore de sintaxe concreta (do inglês, *concrete syntax tree*); trata-se de uma estrutura de dados em árvore sendo uma representação estrutural sintática de alguma gramática formal.

¹⁵ Linguagem computacional representada por expressões simbólicas normalmente referenciadas por *S-expressions* com o tipo de estrutura em árvore sintática (MCCARTHY; LEVIN, 1965). LISP foi a linguagem escolhida por Koza (1990) para a implementação da programação genética devido ao fato de que as expressões simbólicas representam diretamente a árvore sintática de um programa.

Figura 28 – Forma de representação utilizada na programação genética.



Fonte: Adaptado de [Koza \(1994\)](#).

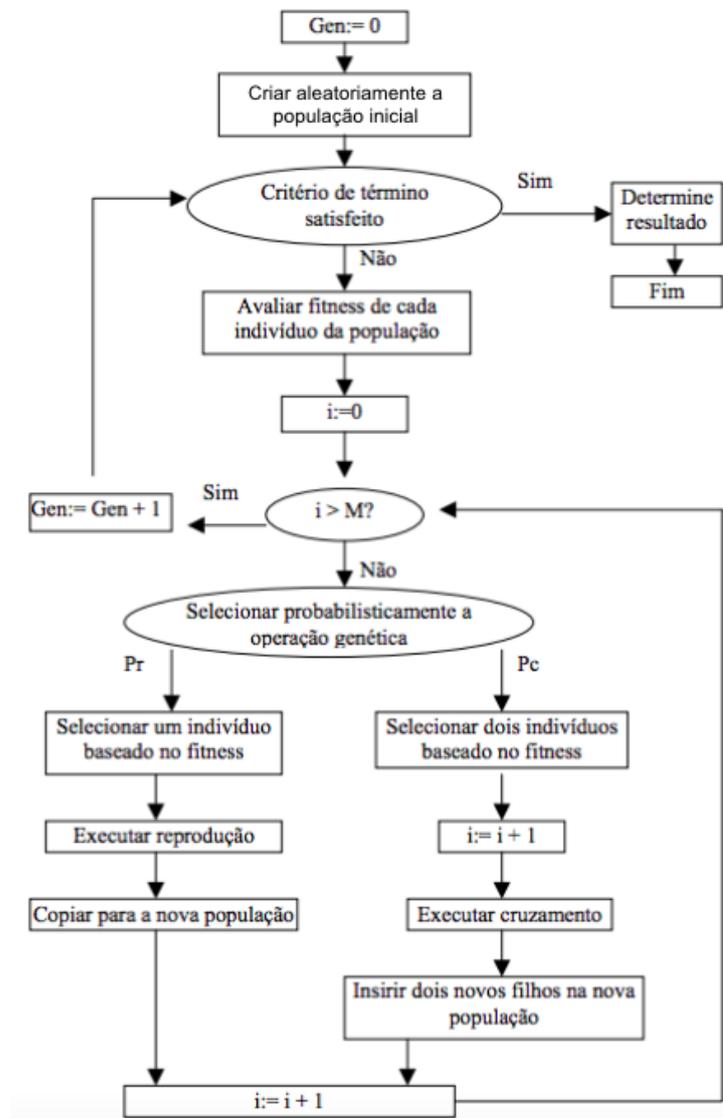
O processo de cruzamento na PG se dá através da troca de galhos da árvore dos pais. As árvores/programas pais podem ter tamanhos e formas diferentes e os filhos serão sub-árvores dos pais, também podendo ter tamanhos e formas diferentes dos seus genitores. A Figura 30 mostra um exemplo da aplicação do cruzamento. As duas primeiras árvores são os pais; deles é escolhido um ponto aleatório, sendo na primeira o ponto 2 enquanto que para a segunda o ponto 5; a árvore é então cortada nesse ponto e os filhos são gerados possuindo a cópia de um dos pais com o galho do outro pai.

Como já foi explicado, a mutação pode ser muito útil na manutenção da variabilidade da diversidade populacional da evolução. Entretanto, ela não é utilizada em algumas das implementações de PG por muitas vezes aplicar alterações que acabam invalidando o indivíduo. Dessa forma, no geral, ela pode ser aplicada desde que respeite o caráter hierárquico dos programas computacionais. Na ausência de mutação, o algoritmo deve possuir uma população inicial suficientemente numerosa e diversificada para que os blocos constituintes de cada programa, os galhos das árvores, possam ser adequadamente reagrupados apenas pelo processo de cruzamento. A Figura 31 mostra um exemplo de mutação válida aplicada na PG onde um dos nós internos da árvore tem seu valor alterado de forma que não invalida o resultado final. Outra alternativa seria a geração de um galho aleatório e a substituição em um determinado nó da árvore pelo galho gerado.

Após o processo de cruzamento e mutação quando aplicado, uma nova população é gerada e a anterior é inteiramente substituída pelos novos indivíduos.

Para garantir que as árvores produzidas nas execuções fossem válidas, [Koza \(1992\)](#) definiu para a PG a propriedade do fechamento, do inglês *closure*, onde, para satisfazê-la, “cada função do conjunto F deve aceitar, como seus argumentos, qualquer valor que possa

Figura 29 – Fluxo completo da programação genética.



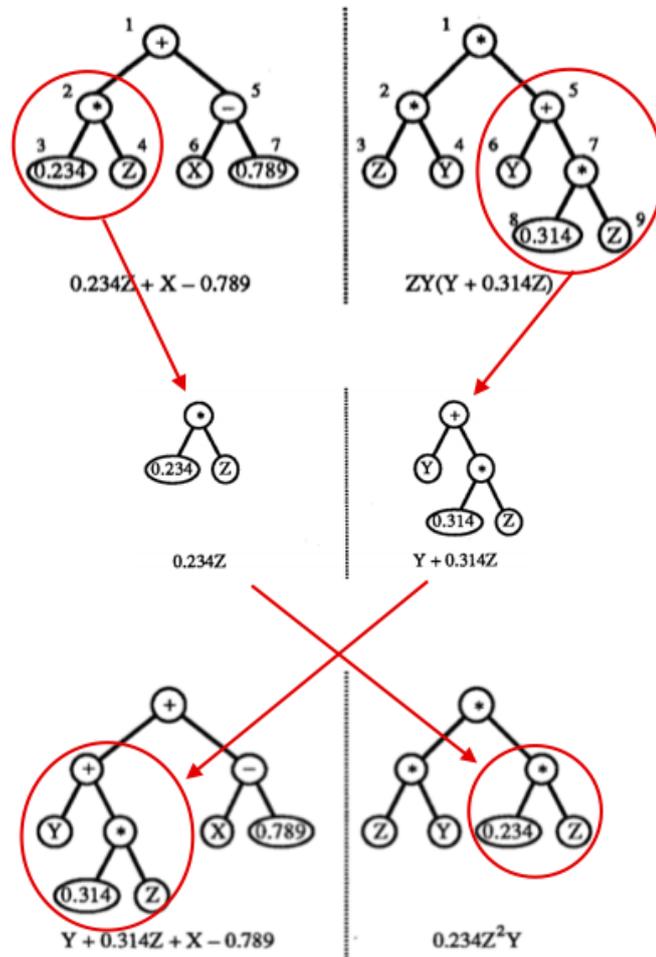
Fonte: Koza (1994).

ser retornado por qualquer função ou terminal do conjunto de funções F e qualquer valor ou dado que possa ser assumido por qualquer função terminal no conjunto de terminais”¹⁶ (KOZA, 1992). Essa propriedade garante que qualquer árvore gerada seja válida, em outras palavras, a árvore pode ser corretamente avaliada.

O fechamento faz com que soluções incorretas sejam descartadas do espaço de busca, contudo, nem sempre a definição das funções e dos terminais adequados ao problema são simples ou fáceis de definir de forma a obedecer essa restrição. Um típico caso de não atendimento a essa regra é a operação divisão, onde matematicamente não é possível

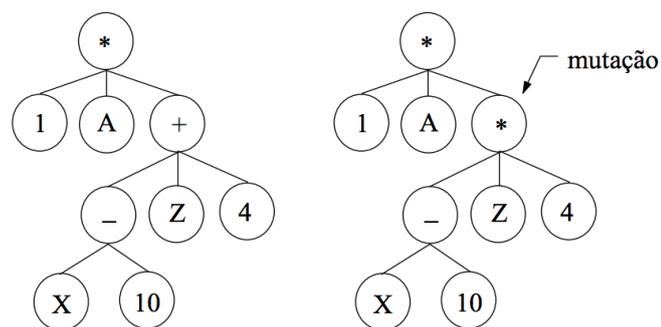
¹⁶ versão original em inglês: “each of the functions in the function set [must] be able to accept, as its arguments, any value and data type that may possibly be returned by any function in the function set and any value and data type that may possibly be assumed by any terminal in the terminal set (KOZA, 1992).”

Figura 30 – Cruzamento na programação genética.



Fonte: Koza (1994).

Figura 31 – Mutação na programação genética.



Fonte: Adaptado de

dividir um valor por zero, mas o zero por si só ainda assim é uma possibilidade válida para o conjunto de terminais. Dessa forma, a obrigatoriedade de aplicação do fechamento acaba limitando o domínio que engloba o problema de forma a impossibilitar a ampla aplicação da PG. Essa restrição estimulou o estudo de técnicas que poderiam abordar essa questão e, assim, Gruau (1996) propôs o uso de gramática para permitir que a PG seja aplicada

a qualquer problema que imponha restrições sintáticas. Com a gramática, são inseridas regras de formação dos programas aos conjuntos F e T norteando, assim, a aplicação de operadores genéticos para que se produza programas sintaticamente corretos ao domínio do problema.

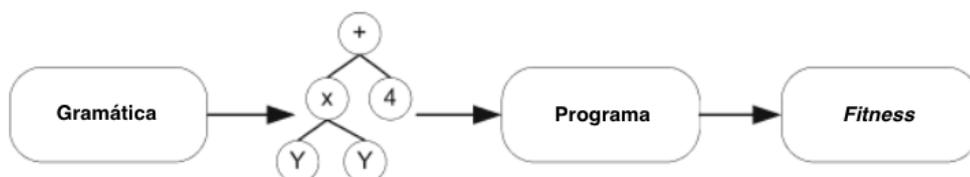
Gramáticas são amplamente utilizadas para aplicar restrições ou explicar interações em diversos domínios da ciência da computação. Por conta disso, elas tiveram um papel importante no desenvolvimento da PG (MCKAY et al., 2010) e, com o crescimento de novos estudos, evoluções da técnica foram surgindo. A combinação de PG com gramática viabiliza principalmente a existência de meios flexíveis de restrição do domínio do problema no espaço de busca e evita buscas improdutivas em regiões inatingíveis, atendendo ao fato de que, como levantado por Koza (1990), é muito importante que os indivíduos gerados por operações genéticas sejam capazes de serem avaliados, ou seja, equivalham a indivíduos válidos. A gramática fica responsável pela definição de como a árvore de decisão ou sintática do indivíduo deve ser construídas e como as operações de produção devem ser aplicadas.

Também conhecida por *Grammar-based Genetic Programming*, nessa técnica, o conjunto de terminais e funções, utilizado na PG, são substituídos por uma gramática que possui regras a serem aplicadas nos terminais e nas funções para garantir que toda a produção origine indivíduos sintaticamente válidos (PAPPA; FREITAS, 2009).

Algoritmos que aplicam a técnica devem ser compostos por uma gramática, um mecanismo de buscas e um processo de mapeamento do genótipo/fenótipo.

A Figura 32 mostra como funciona o fluxo da PG baseada em gramática. Temos uma gramática que vai definir como será a geração de cada indivíduo da população; esse indivíduo é representado em forma de árvore; o programa que ele representa é executado e, por fim, seu *fitness* é calculado.

Figura 32 – Fluxo da programação genética baseada em gramática.



Fonte: Pappa e Freitas (2009).

A forma de gramática aplicada a PG mais conhecida é a gramática livre de contexto

(GLC), do inglês, *context-free grammar*; ela é representada pela 4-tupla $\{N, T, P, S\}$, onde N é o conjunto de não terminais, T o conjunto de terminais, P o conjunto de regras de produção e S o simbolo inicial. As regras de produção tem a forma $x ::= y$, onde $x \in N$ e $y \in \{T \cup N\}^*$ e existindo um conjunto de possibilidades de produção para x , cada uma das possíveis produções é delimitada pelo simbolo disjuntivo ‘|’ (BARRETT et al., 1986; PAPPA; FREITAS, 2009). Um exemplo de uma GLC para geração de expressões aritméticas simples é mostrado a seguir, onde $G = \{N, T, P, S\}$:

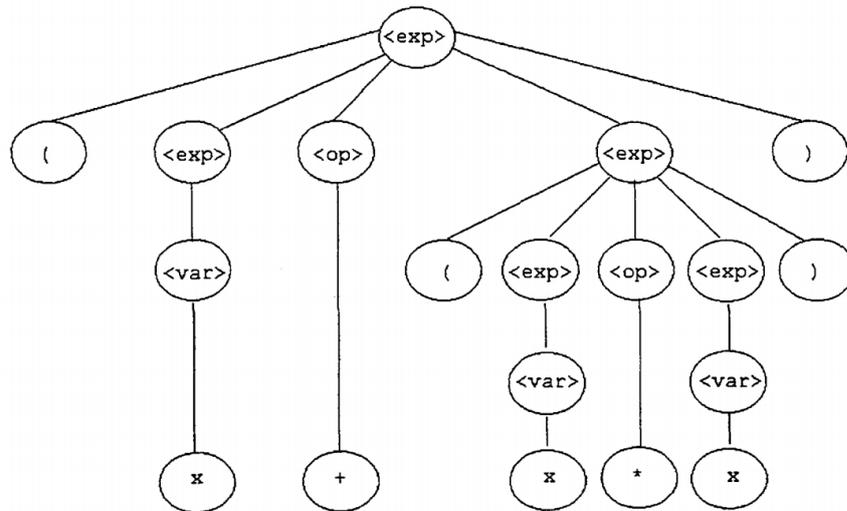
$$\begin{aligned}
 N &= \{ \langle \text{exp} \rangle, \langle \text{var} \rangle, \langle \text{op} \rangle \} \\
 T &= \{ x, +, -, /, * \} \\
 S &= \langle \text{exp} \rangle \\
 P &= \{ \\
 &\quad \langle \text{exp} \rangle ::= \langle \text{var} \rangle \\
 &\quad \quad | (\langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle) \\
 &\quad \langle \text{op} \rangle ::= + \\
 &\quad \quad | - \\
 &\quad \quad | / \\
 &\quad \quad | * \\
 &\quad \langle \text{var} \rangle ::= x \\
 &\quad \}
 \end{aligned}$$

Nesse exemplo, temos $\langle \text{exp} \rangle$, $\langle \text{var} \rangle$ e $\langle \text{op} \rangle$ como não terminais e como terminais temos o “x” e os operadores matemáticos $+$, $-$, $/$ e $*$. No contexto de G , um exemplo de indivíduo que pode ser gerado é $x + (x * x)$. Obedecendo as regras definidas em P , a gramática pode gerar toda uma população e todos serão indivíduos válidos para o problema. Cada uma dessas soluções propostas podem ser representados por sua árvore sintática (WONG; LEUNG, 2006).

A Figura 33 mostra como seria a árvore sintática para a representação do exemplo; nela é possível observar que cada nó da folha possui apenas um terminal e eles são exatamente uma das opções do que foi definido em T ; a expressão iniciando em S e cada produção pode ser vista como um nó da árvore.

A geração de programas baseados na gramática tem início selecionando a produção inicial S , o nodo filho será uma das opções de produção do pai escolhida aleatoriamente, no exemplo uma possibilidade seria $\langle \text{var} \rangle$; se esse nodo não for terminal, como é o de

Figura 33 – Exemplo de uma árvore sintática para representação do indivíduo $x+(x*x)$.



Fonte: Adaptado de [Rodrigues \(2002\)](#).

$\langle var \rangle$, então uma das suas opções de filhos deve ser escolhida aleatoriamente; para o caso de $\langle var \rangle$ só há uma opção de produção, o x , e o nodo é terminal, logo, se encerra aí a produção do indivíduo. No caso de exemplos maiores, a produção segue até que todos os nodos filhos sejam terminais ([SETHI, 1996](#)).

O processo de criação dos indivíduos baseados na gramática para inicialização da primeira geração se dá de forma relativamente simples. Existem várias técnicas propostas na literatura mas existem duas principais ([NICOLAU, 2017](#)), uma mais tradicional onde existe a geração de um genoma, que será posteriormente mapeado em uma árvore ([RYAN et al., 1998](#)), e uma mais recente que propõe o processo inverso, a geração de uma árvore de derivação que posteriormente produzirá um genoma que a represente ([FAGAN et al., 2016](#); [FORSTENLECHNER et al., 2016](#)).

Na forma tradicional de inicialização temos que cada programa é representado por uma árvore e essa árvore representa escolhas que foram determinadas pelo mapeamento do genótipo do indivíduo, sendo o genótipo um vetor de tamanho fixo de genes ou códons, representados por valores numéricos de 8-bits; esses valores são usados para decidir a produção que será selecionada sempre que um não terminal precise ser interpretado. Na geração do programa a partir do genótipo, em outras palavras, do vetor numérico, sempre que é gerado um nó não terminal e ele precisa ser interpretado, se é consultado o vetor e sequencialmente um de seus valores inteiro é selecionado; esse valor “y” é aplicado na

Equação 2.1:

$$f(y) = y \pmod{N}, \quad (2.1)$$

onde, N é a quantidade de opções para aquela produção. O resultado de $f(y)$ determinará qual produção será escolhida para o não terminal (RYAN et al., 1998). Por exemplo, supondo que temos um indivíduo com o seguinte genoma:

113	34	85	153	178	93	4	249	7
-----	----	----	-----	-----	----	---	-----	---

fazendo o mapeamento dele para a gramática G expressa acima e considerando que temos que tomar decisão de escolha de produção para o não terminal $\langle op \rangle$, iniciamos ordenando as opções de escolha de $\langle op \rangle$ de 0 a 3, conforme sua quantidade de variações possíveis:

$$\begin{aligned} \langle op \rangle ::= & + \quad (0) \\ & | - \quad (1) \\ & | / \quad (2) \\ & | * \quad (3) \end{aligned}$$

supondo que o próximo códon a ser selecionado é o de valor 85, aplicamos o valor na Equação 2.1, onde teremos:

$$f(85) = 85 \pmod{4} \rightarrow f(85) = 1, \quad (2.2)$$

dessa forma, a opção selecionada para a produção do $\langle op \rangle$ será a de número 1, o terminal $-$. Após a decodificação dos cromossomos em indivíduos, o processo evolutivo segue para as etapas seguintes.

A inicialização tradicional da população por genoma é realizada criando um vetor aleatório de valores que posteriormente é traduzido em uma árvore, dessa forma, não possibilita garantia de controle sobre nenhum aspecto da população inicial. Normalmente se é especificado uma profundidade máxima para evitar árvores muito grandes, no entanto, durante a geração da árvore não é possível prever sua profundidade, somente após a completa tradução que isso pode ser verificado, o que traz como consequência a invalidação dos indivíduos fora do limite de profundidade. Técnicas mais recentes propõem a geração inicial da árvore de derivação permitindo o controle de vários aspectos da população, como

por exemplo as variações no limite de profundidade das árvores. Conforme o interesse desse estudo, focaremos na técnica de inicialização proposta por [Fagan et al. \(2016\)](#) conhecida como *Position Independent Grow* (PIGrow), ou “crescimento independente da posição” em tradução livre.

PIGrow é uma variação da técnica proposta por [Koza \(1992\)](#) denominada *Grow*. O método de [Koza \(1992\)](#) sugere a criação de árvores sintáticas iniciando pelo nó raiz do conjunto de funções e seguindo escolhendo aleatoriamente funções ou terminais até que todas as folhas sejam compostas por terminais; o autor sugere ainda que se algum galho da árvore alcançar o tamanho máximo de profundidade especificado, então a próxima opção escolhida será obrigatoriamente um terminal, dessa forma, a profundidade máxima não é extrapolada.

Assim como o *Grow*, no *PIGrow* as árvores são criadas a partir da raiz e crescem com produções aleatórias até uma determinada profundidade máxima previamente especificada. No entanto, no procedimento apresentado por [Fagan et al. \(2016\)](#), a escolha do próximo galho a ser expandido é feita de forma aleatória, em outras palavras, independente da posição em que estão dado que técnicas tradicionais optam por aplicar a expansão nos galhos mais a esquerda, e também se é garantido que pelo menos um dos galhos tenha a profundidade máxima especificada; isso é feito escolhendo produções recursivas se houver apenas escolhas não terminais e a profundidade máxima ainda não houver sido alcançada por nenhum outro galho.

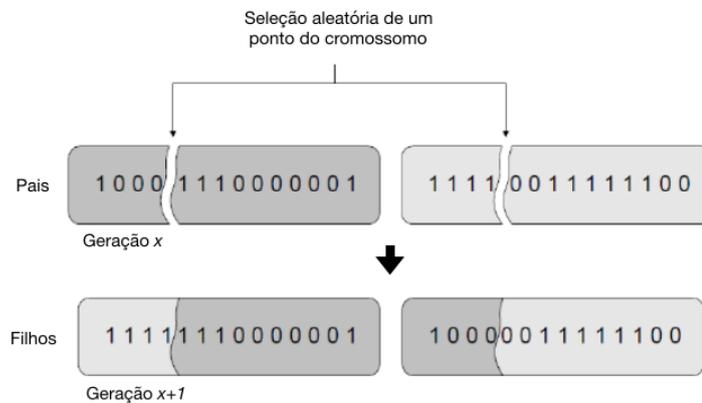
Exceto o fato de que na GGGP é necessário haver a codificação e decodificação do cromossomo, os operadores genéticos aplicados na GGGP, cruzamento e mutação, seguem a mesma sequência geral da CE e devem respeitar a gramática do problema para que não hajam alterações que permitam a criação de indivíduos inválidos.

Para os fins de interesse desse estudo, abordaremos a técnica de seleção conhecida como torneio, do inglês *tournament selection*. Nessa abordagem, “*n*” indivíduos são aleatoriamente escolhidos na população e eles competem entre si, sendo selecionado o ser com maior adaptabilidade ao problema, em outras palavras, o melhor do grupo ([GOLDBERG; DEB, 1991](#)).

O cruzamento pode ocorrer de diversas formas, de acordo com o problema a ser analisado e desde que continue obedecendo a regra de geração de seres válidos. Uma das formas que observaremos aqui é a reprodução de um ponto, pelo qual são selecionados

dois pais, conforme o método de seleção aplicado, e o genoma de cada um é “cortado” em um determinado ponto aleatório produzindo dois filhos de forma que um dos filhos é formado pelos cromossomos do início de um dos pais até o ponto de corte unido com os cromossomos do segundo pai a partir do corte até o final. O outro filho é gerado pelos cromossomos complementares não utilizados no primeiro filho (POLI; LANGDON, 1998). Dessa forma, ambos os filhos tem características dos dois pais e a geração da árvore dos indivíduos continua respeitando a gramática quando ele é interpretado, como mostra a Figura 34.

Figura 34 – Exemplo de uma operação de cruzamento em um ponto no cromossomo.



Fonte: Adaptado de Chiroma et al. (2017).

Assim como o cruzamento, também há formas de mutação que podem ser aplicada diretamente no genoma do indivíduo, como é o caso do *Flip Per Codon*. Na técnica todos os códons do indivíduo são mutados considerando uma dada probabilidade de mutação “ x ” especificada de acordo com o problema analisado.

Por fim, um dos últimos processos ao qual a geração é submetida é a decisão sobre quais indivíduos farão parte do ciclo seguinte. A técnica de definição os indivíduos que formarão a população da geração seguinte de interesse desse estudo é a geracional, onde o qual todos os pais são substituídos pelos filhos que foram gerados (FOGEL, 1995).

3 Trabalhos Relacionados

3.1 Estado da Arte

Nasr-Esfahani et al. (2016) apresenta um sistema de detecção automático de melanoma. É proposto a aplicação de um pré-processamento nas imagens das lesões pois, segundo o autor, possuem a presença de efeitos de iluminação resultantes da inexistência de um padrão de iluminação para captura das imagens e pelo reflexo natural da superfície da pele podem provocar interferências nos resultados. Se visa reduzir efeitos que podem fazer com que a rede se atrapalhe e aprenda padrões dos ruídos que não são interessante para a classificação. A etapa realiza uma correção na iluminação aplicando algo muito similar ao proposta por Giotis et al. (2015) que detecta os efeitos de iluminação pelas mudanças bruscas nos canais de saturação e os valores de cores do HSV¹; os efeitos que se desejam remover são extraídos através da exclusão de um conjunto específico de gradientes. Observando que as imagens de lesões contém também partes saudáveis de pele e entendendo que isso pode levar a rede a aprender características que não são interessantes, é aplicada a máscara de segmentação *k-means*² utilizando $k = 2$ para extração da região da imagem composta pela lesão; posteriormente se é utilizado o filtro gaussiano³ com $\sigma = 2$ na parte da máscara composta pela pele saudável. Após isso, as imagens são submetidas a execução de uma rede neural convolutiva simples de seis camadas no total, sendo duas convolutivas de filtro de tamanho 20 e 50 respectivamente e formato 5×5 ; cada uma delas seguidas por um *pooling*; e mais duas camadas totalmente conectadas com 100 e 2 neurônios responsáveis pela classificação final. A proposta tem sua qualidade avaliada utilizando uma base de dados com 170 imagens, sendo 70 melanoma, alcançando acurácia de 0,81%.

Visando sanar as variações que as imagens fotográficas que as lesões podem possuir, Esteva et al. (2017) utiliza uma rede neural pré-treinada em 1.28 milhões de imagens de objetos do ImageNet 2014 (RUSSAKOVSKY et al., 2015). A rede pode ser aplicada a

¹ HSV trata-se de um canal de cores composto por matiz (*Hue*), saturação (*Saturation*) e valor (*Value*), também conhecido como HSB, matiz (*Hue*), saturação (*Saturation*) e brilho (*Brightness*) (CARDANI, 2001).

² A técnica consiste em dividir determinado conjunto em k subgrupos disjuntos que compartilham características semelhantes entre si.

³ Filtro gaussiano reduz possíveis ruídos da imagem através da aplicação da função gaussiana, produzindo um efeito de suavização ou desfoque (AURICH; WEULE, 1995).

ambos tipos de imagens de lesões, fotográficas ou dermatoscópicas, sem a necessidade de pré-processamento nos dados. É proposto a aplicação do GoogLeNet - *Inception v3* treinado e todo o aprendizado adquirido pela rede é utilizado, através da técnica de transferência de aprendizado, para a classificação das lesões de pele. A técnica é testada de duas formas, ambas utilizando uma base de 129.450 imagens sendo 3.374 dermatoscópicas; a primeira é a divisão das classes em três – lesões benignas, malignas e não neoplásicas – apresentando acurácia de 72,1%; e a segunda dividindo as imagens em nove classes de doenças que compartilham entre si tratamento similar, a acurácia alcançada foi de 55,4%.

Encontramos também na literatura aplicações de programação genética voltada à detecção do melanoma em imagens dermatoscópicas. [Ain et al. \(2017\)](#) propõem a combinação de conhecimentos da dermatologia com técnicas de visão computacional. Inicialmente é utilizada a extração de características das imagens com o Local Binary Patterns (LBP)⁴, processo que produz 59 características; esse resultado é então acrescentado de mais 12 características obtidas a partir da observação da lesão de pele por um dermatologista. Como a base de dados utilizadas não é balanceada, o cálculo do *fitness* foi feito utilizando a técnica proposta por [Patterson e Zhang \(2007\)](#) que considera ambas as classes de classificação com igual importância. Os testes foram feitos considerando três cenários: 1. apenas as características advindas dos dermatologistas; 2. as resultantes do LBP; 3. e ambas. O melhor resultado foi alcançado com o terceiro cenário de testes citados, utilizando características extraídas pelo LBP juntamente com as definidas pela equipe médica que deu assistência os estudos, apresentando a média da acurácia de 5 cinco execuções com 900 gerações com 30 indivíduos cada de 98,34%; para o segundo cenário, onde não há presença humana no processo de extração de características, a média nas mesmas condições foi de 85,42%; e no primeiro cenário, 97,08%.

[Winkler et al. \(2011\)](#) apresentam um estudo que também faz utilização de características, no entanto, a base de dados utilizada para validação da técnica não é composta por imagens e sim por um conjunto de características de pacientes coletados pelo laboratório central do *General Hospital Linz* entre os anos de 2005 e 2008. As características são diversas e variam entre idade, sexo até a existência ou não de substâncias no sangue ou tecidos do corpo conhecidas por indicarem a presença de tumores, nenhuma delas

⁴ LBP é um descritor de texturas proposto inicialmente por [Ojala et al. \(1996\)](#), mas que só ficou conhecido em [Ojala et al. \(2002\)](#), é amplamente utilizado em aplicações de visão computação para extração de características; nele se é computado a representação local da textura construída através da comparação de cada *pixel* com seus vizinhos.

diz respeito diretamente a valores como tamanho de lesão de pele, formato ou análises desse tipo. Se busca detectar três tipos de câncer: melanoma, câncer de mama e de pulmão. É utilizado programação genética para otimização das características de cada indivíduo que se mostram importantes para a classificação. O cálculo do *fitness* é feito utilizando algoritmo de aprendizado de máquina, máquinas de vetores de suporte⁵, para estimar a quantidade de valores previstos corretamente. Foi alcançado 74% de acurácia na identificação do melanoma, vale salientar também que a base de dados não continha dados de pacientes sem um dos três tipos de tumores analisados.

Haenssle et al. (2018) trazem uma comparação entre a performance de uma CNN para classificação de melanoma em imagens dermatoscópicas com performance de um grupo de 58 dermatologistas, sendo 30 deles especialistas na área. A técnica utiliza a rede neural *Inception v4* treinando-a com imagens dermatoscópicas. O estudo revela que quando os médicos receberam apenas as imagens das lesões para classificá-las, a CNN apresentou melhores resultados. No entanto, sabendo que no diagnóstico clínico o dermatologista conta com vários outros dados do paciente além da visualização da lesão, então dados clínicos foram adicionados às imagens e se observou melhora na performance, todavia, menor de que a da CNN, apresentando especificidade de 75,7% contra 82,5%.

Na técnica apresentada por Aima e Sharma (2019) são utilizadas 514 imagens da base de dados pública do ISIC (International Skin Imaging Collaboration, 2019) aplicando um pré-processamento antes de utilizadas como entrada para uma rede neural convolutiva. Durante o pré-processamento, as imagens foram inicialmente convertidas de RGB para o formato HSV com um valor específico para o conjunto de variação de cores que resultou em imagens contendo apenas a lesão com as demais partes correspondente a pele se tornaram pretas; as cores foram então convertidas para o sistema de cores preto e branco e as imagens foram redimensionadas para o tamanho 100×100 ; é aplicado um processo de

⁵ Popularmente conhecidas como SVMs por sua denominação em inglês, *Support Vector Machines*, as máquinas de vetores de suporte são classificadores descritivos lineares baseados no conceito de plano de decisão que define os limites de decisões aprendidas entre classes diferentes, lineares ou não-lineares. Elas são uma técnica de aprendizado supervisionado que estima um hiperplano ideal separa em classes um determinado dado de entrada (RUSSELL; NORVIG, 2009).

segmentação por limiarização⁶ seguido pelas operações morfológicas de erosão⁷ e dilatação⁸ visando reduzir ruídos da imagem; a área de interesse, nesse caso a lesão de pele, é extraída da imagem e esse resultado final são os dados de entrada da CNN; detalhes da arquitetura da CNN não são descritos, sabendo-se apenas que o otimizador utilizado é o SGD⁹. O estudo apresenta acurácia de 74,76%.

O estudo de [Vasconcelos e Vasconcelos \(2017\)](#) utiliza uma base de dados com 900 imagens do ISIC 2016 e, sabendo que a quantidade de imagens influencia diretamente no resultado das redes neurais, a técnica apresenta uma proposta de *data augmentation* para que o número de amostras seja ampliado; dentre os tipos de variações para o aumento dos dados, foram aplicados cortes formando novas imagens com outros formatos incluindo variações de ângulos de forma a garantir que em todas elas a lesão estava presente; variação de cores dentro de uma extensão considerada próxima das cores observadas nas imagens de lesões de pele; o ponto forte do estudo é a proposta da aplicação de um novo tipo de técnica para aumento da quantidade de dados chamada de *data augmentation* baseada no conhecimento de especialistas, nela é proposto a criação de novas imagens a partir da distorção do eixo principal da imagem original mantendo os padrões e simetrias da lesão, dessa forma são criadas novas figuras artificiais que podem alimentar a CNN. É utilizada a rede neural GoogLeNet e o resultado máximo de acurácia apresentado é 84,4%.

No que diz respeito a programação genética orientada à gramática, recentemente [Assunção et al. \(2018\)](#) propôs DENSER, uma abordagem para evolução de redes de *deep learning*, aplicadas a redes neurais convolutivas. São combinados princípios de algoritmos genéticos com evolução gramatical para evoluir diretamente uma lista sequencial de camadas, encapsulando os valores dos parâmetros numa posição do genótipo para facilitar o uso de operadores genéticos. Dessa forma, sendo capaz de reaplicar esse método para diferentes estruturas de rede e domínios apenas alterando sua gramática. O método

⁶ Trata-se de um princípio pelo qual determinada imagem é separada em duas regiões ou duas classes, sendo uma a área de interesse e o plano de fundo. O processo também é conhecido como binarização e, ao final, tem-se uma imagem binária em preto e branco com duas classes distintas, sendo, em geral, a região branca a região de interesse. A técnica busca encontrar no histograma da imagem o vale entre dois picos que será considerado o limiar para separar as duas classes. Posteriormente são agrupados todos os *pixels* maiores que o valor do limiar em uma classe e os menores em outra ([PHAM et al., 2000](#)).

⁷ Essa técnica reduz os tamanhos das áreas mais escuras da imagem e o resultado final aparenta ser mais claro que o original.

⁸ Técnica que atenua os detalhes escuros das imagens e acentua os detalhes claros.

⁹ *Stochastic Gradient Descent* (SGD) é uma técnica que atualiza os pesos da rede após o treinamento de cada amostra de entrada

superou as abordagens evolutivas anteriores das gerações de CNNs, criando redes com desempenho sendo os novos estados da arte na área tendo menos conhecimento prévio e evoluiu CNNs com topologias novas do que as projetadas manualmente. O estudo foi aplicado na base de dados CIFAR-10.

3.2 Resumo do Estado da Arte e Contribuições do Trabalho Proposto

É possível encontrar na literatura diversos estudos recentes voltados à detecção do melanoma utilizando algum tipo de técnica derivada do aprendizado de máquina. Nas abordagens de *deep learning* vemos [Nasr-Esfahani et al. \(2016\)](#) apresentar um algoritmo que aplica pré-processamento nas imagens seguido por uma CNN simples alcançando 81% de acurácia. Seguindo a mesma ideia, vemos [Aima e Sharma \(2019\)](#) com um pré-processamento que elimina partes saudáveis da pele da imagem de forma a alimentar a CNN apenas com as imagens das lesões de pele, não foi descrito o tipo de arquitetura da CNN, mas o resultado da acurácia obtida nos testes foi de 74,76%. [Vasconcelos e Vasconcelos \(2017\)](#) apresentam ainda uma espécie de pré-processamento sem alterações nas imagens em si, é proposto a aplicação da técnica de *data augmentation*, inclusive com uma nova abordagem proposta para a geração de novas lesões de pele a partir das imagens existentes, visando aumentar a quantidade de dados de entrada para melhorar o resultado da CNN escolhida, GoogLeNet. [Haenssle et al. \(2018\)](#) compara a performance do GoogLeNet - *Inception v4* com a classificação de lesões realizadas por um grupo de dermatologistas trazendo comprovação de que o resultado da máquina foi melhor de que o humano. Ainda utilizando o GoogLeNet, temos [Esteva et al. \(2017\)](#) propondo transferência de conhecimento de uma rede GoogLeNet - *Inception v3* treinada com dados do ImageNet. Os resultados, mesmo no melhor caso, não alcançaram uma acurácia tão alta mas, ainda assim, foi possível comprovar que a máquina teve um melhor desempenho de que a análise humana. Vale salientar que as acurácias descritas se referem aos valores apresentados nos estudos, não refletindo, no entanto, a análise com a aplicação da mesma base de dados para todos.

Olhando um pouco para outros tipos de abordagens, temos em 2011 [Winkler et al. \(2011\)](#) utilizando a programação genética. A abordagem apresentada, no entanto, classifica dados do fisiológicos e biológicos do paciente, não havendo foco apenas na lesão em si,

outra característica é que o algoritmo não analisa imagens das lesões e não existe na base de dados amostras de pacientes saudáveis, todos tem algum tipo de câncer. Já a técnica proposta por [Ain et al. \(2017\)](#), que também usa programação genética, apresenta um bom resultado se comparado as demais, contudo, como são utilizadas características extraídas das imagens e outras especificações definidas manualmente por dermatologistas, a análise dos dados fica limitada a necessidade de levantamento humano de um especialista para catalogação de algumas das características das imagens.

Quanto a estudos utilizando programação genética orientada à gramática, temos [Assunção et al. \(2018\)](#) apresentando uma técnica de evolução gramatical que conseguiu evoluir redes neurais convolutivas capazes que alcancem o estado da arte na área. Os experimentos realizados pela pesquisa foram feitos com a base de dados CIFAR-10, não havendo análises com dados médicos.

A Tabela 1 mostra um resumo das técnicas utilizadas por cada um dos autores descritos no estado da arte.

Tabela 1 – Estudos analisados no estado da arte.

Referência	Técnica
1 - Nasr-Esfahani et al. (2016)	Pré-processamento + CNN
2 - Esteva et al. (2017)	Transferência de conhecimento (<i>Inception v3</i>)
3 - Ain et al. (2017)	PG + (LBP + extração de características por dermatologistas)
4 - Winkler et al. (2011)	Programação genética sem uso de imagens
5 - Haenssle et al. (2018)	<i>Inception v4</i>
6 - Aima e Sharma (2019)	Pré-processamento + CNN
7 - Vasconcelos e Vasconcelos (2017)	<i>Data augmentation</i> + GoogLeNet
8 - Assunção et al. (2018)	Algoritmos genéticos com evolução gramatical

Fonte: A autora.

Não foram encontrados na literatura trabalhos que utilizam programação genética

orientada a gramática para otimizar redes neurais de classificação de melanoma a partir de imagens dermatoscópicas de lesões de pele.

4 Método Proposto

Este capítulo apresenta o método proposto por esse estudo e os detalhes das decisões tomadas em cada etapa. A gramática é descrita com suas particularidades e exemplos, de forma a facilitar seu completo entendimento.

4.1 Análises Prévias

Observando que o estado da arte de visão computacional atualmente traz as CNNs como a técnica que apresenta os melhores resultados para classificação de imagens (RUSSAKOVSKY et al., 2015), inclusive com resultados promissores para imagens médicas (LI et al., 2014; KAYALIBAY et al., 2017; KHVOSTIKOV et al., 2018), a técnica foi escolhida para um estudo da aplicação em imagens dermatoscópicas.

CNNs pequenas e grandes foram projetadas e tiveram suas performances analisadas para a base de dados do estudo, no entanto, devido a toda a diversidade de escolhas de parâmetros envolvidas no processo, a definição de uma arquitetura adequada ao problema se mostrou ser uma tarefa complexa. Foi então que, percebendo uma maior tendência de melhores classificações obtidas a partir de redes convolutivas com arquitetura menores, foi decidido aplicar a programação genética com a utilização de gramática para realizar a busca por boas redes, de forma que as decisões arquiteturais da CNN fossem exploradas dispensando a necessidade de um especialista para defini-las.

A complexidade do trabalho se mostrou na definição do espaço de busca de forma a não deixá-lo muito grande, o que poderia fazer com que a busca demorasse muito a convergir para uma boa região, e não limitá-la demais para que se permitisse alcançar resultados competitivos.

4.2 Fluxograma do Método Proposto

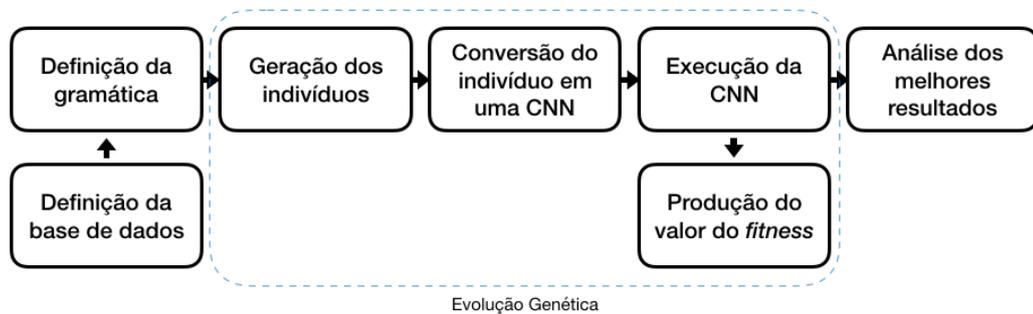
A abordagem proposta se baseia principalmente na escolha da gramática a ser utilizada, ela é responsável por viabilizar buscas no espaço correto de amostragem com possibilidade de resultados promissores.

A definição da base de dados também é importante para que todos os indivíduos

sejam avaliados utilizando os mesmos parâmetros, em outras palavras, é importante que a acurácia das redes corresponda a valores resultantes do mesmo conjunto de treinamento e testes, variando apenas a arquitetura das CNNs que foram definidas pela interpretação do indivíduo produzido durante a evolução genética.

A Figura 35 mostra o fluxograma do método proposto; na imagem é possível observar as ações tomadas antes do início do processo evolutivo e as que englobam a evolução. Vale salientar que o procedimento não conta com interferências de ações humanas na classificação das imagens.

Figura 35 – Fluxograma do método proposto.



Fonte: A autora.

Ainda sobre a Figura 35, apenas partes de interesse desse estudo foram projetadas no fluxograma, é importante ressaltar que o processo de evolução genética envolve também outros passos, como por exemplo mutação e cruzamento; todas essas etapas seguem normalmente e detalhes dos valores e tipos de cada um aplicados serão descritos nas próximas sessões.

4.3 Técnica Proposta

O estudo apresenta a aplicação de programação genética trazendo como diferencial a gramática é apresentada na sessão 4.3.2, descrevendo também a evolução da gramática apresentado na sessão 4.3.1.

4.3.1 Evolução da Gramática Proposta

Durante todo o período de estudos dessa pesquisa, vários tipos de gramáticas foram testados até se chegar na proposta. Foi uma grande evolução de aprendizado e experiência sobre o que pode se adaptar melhor à base de dados e o que não traz resultados tão satisfatórios. A acurácia evoluiu de uma média de 50 a 60%, para mais de 80%.

Os estudos iniciaram com testes utilizando a base de melanoma e o CIFAR-10 (KRIZHEVSKY; HINTON, 2009); o objetivo de aplicar o CIFAR-10 foi o fato da existência de estudos anteriores que utilizam abordagem semelhante para a base. Dessa forma seria possível ter noção se a gramática se mostrava competitiva e, sendo assim, poderia ser aplicada a outras bases. Resultados do estudo com o CIFAR-10 foram publicados em (DINIZ et al., 2018).

A Figura 36 mostra a definição da gramática inicialmente estudada; nela as camadas de convolução e de *pooling* são agrupadas em blocos, sendo a quantidade de convoluções definidas pela produção de $\langle N \rangle$ e a de blocos por $\langle M \rangle$; de acordo com a produção de $\langle POOL \rangle$, a arquitetura pode ou não conter *poolings*; as camadas FCs variam conforme o valor de $\langle K \rangle$.

Figura 36 – Primeira gramática utilizada pelo estudo.

$$\begin{aligned}
 \langle \text{FINAL_EXP} \rangle & \models \langle \text{EXP_2} \rangle \langle \text{FC} \rangle \\
 \langle \text{EXP_2} \rangle & \models (\langle \text{EXP_1} \rangle \times \langle M \rangle) \\
 \langle \text{EXP_1} \rangle & \models (\langle \text{CONV} \rangle \langle \text{POOL} \rangle) \\
 \langle \text{FC} \rangle & \models \text{fc} \times \langle K \rangle \\
 \langle \text{POOL} \rangle & \models \text{pool} \mid \varepsilon \\
 \langle \text{CONV} \rangle & \models (\text{conv} \times \langle N \rangle) \\
 \langle N \rangle & \models 1 \mid 2 \mid 3 \\
 \langle K \rangle & \models 0 \mid 1 \mid 2 \\
 \langle M \rangle & \models 1 \mid 2 \mid 3
 \end{aligned}$$

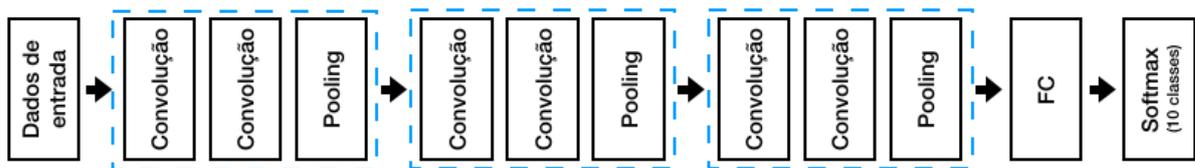
Fonte: A autora.

A gramática inicial permitia a produção de redes neurais com até nove camadas convolutivas, todavia, sabendo que o ImageNet mostrava que redes maiores também poderiam apresentar boa performance de classificação, foram aplicadas alterações para que a gramática permitisse a criação de arquiteturas com até 30 convoluções. Se pensarmos que

só de convoluções haviam 30 opções de produções diferentes, existindo ainda as variações de *pooling* e FC, pode-se imaginar que a gramática ficou com uma enorme quantidade de variações possíveis. Fica claro também que as execuções precisavam ser grandes o suficiente para serem capazes cobrir um espaço de busca exponencialmente maior. Além disso, a base de dados do CIFAR-10 possui 60 mil imagens, cada treinamento e teste das redes maiores era bem mais custosos considerando tempo e recursos de processamento. Os testes nesse cenário não apresentaram resultados muito competitivos, com os melhores sendo redes com arquiteturas menores, logo, a ampliação da quantidade de convoluções não evoluiu.

Foi notado então que as CNNs que produziam os melhores resultados para o CIFAR-10, não necessariamente apresentavam boa performance para a base de imagens dermatoscópicas. A rede que apresentou a melhor acurácia para a base de dados do CIFAR-10 tem uma arquitetura composta por três blocos sequenciais compostos por duas camadas convolutivas e um *pooling* cada um, como mostrado na Figura 37. Para a base de dados, ela obteve uma acurácia de 79,35%. No entanto, para a base de melanoma desse estudo, revelou ter uma performance de 67,71% – média resultante de cinco execuções da arquitetura sob as mesmas condições. Foi então que os estudos começaram a ser focados apenas nas imagens de lesões de pele e, observando que os resultados máximos não alcançavam os 70% de acurácia, mudanças precisavam ser feitas.

Figura 37 – Representação da melhor arquitetura da gramática apresentada em [Diniz et al. \(2018\)](#) para a base de dados CIFAR-10.



Fonte: A autora.

Existia a necessidade de exploração da criação de redes sem a limitação dos blocos para que a quantidade de convoluções e a existência de *poolings* não ficasse restrita a hierarquia dos blocos, permitindo exploração em um espaço de buscas maior sem a necessidade do aumento da quantidade de convoluções. Para tanto, a gramática precisava ser reestruturada de forma a viabilizar as novas alterações. Nesse momento, também se objetivou que a reestruturação continuasse permitindo a criação de indivíduos no modelo

da arquitetura da gramática inicial, ou seja, redes com a arquiteturas como as dos blocos; dessa forma ampliamos as opções de buscas enquanto mantivemos o espaço já conhecido. A gramática evoluiu para uma versão que já adquiria características da proposta, como é mostrado na Figura 38.

Figura 38 – Evolução da gramática inicialmente estudada.

```

⟨network-architecture⟩ ⊢ ⟨layers⟩ ⟨classification⟩
  ⟨layers⟩ ⊢ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    ⟨layer⟩
    | ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩ ⟨layer⟩
    ⟨layer⟩ ⟨layer⟩
  ⟨layer⟩ ⊢ ⟨convolution⟩
    | ⟨pooling⟩
  ⟨classification⟩ ⊢ (⟨layer⟩ × ⟨num-fc⟩)
  ⟨convolution⟩ ⊢ layer:conv
  ⟨pooling⟩ ⊢ layer:pool-max
    | layer:pool-avg
  ⟨num-fc⟩ ⊢ 0 | 1 | 2

```

Fonte: A autora.

As redes podiam agora ter no máximo dez camadas e cada uma delas individualmente poderiam ser convolução ou *pooling*; isso significa que a rede podia começar já por um *pooling* e, ainda que isso seja um comportamento incomum dentre as CNNs conhecidas, não chega a ser tecnicamente incorreta, então foi mantido. Como mostrado na produção <pooling>, foi inserido a opção do *pooling* ser de média ou máxima.

Apesar da gramática ter sofrido alterações, apenas os parâmetros responsáveis por definir a quantidade de camadas e o tipo estava variando. Sabemos que a criação de uma CNN envolve a escolha de vários outros parâmetros e hiper-parâmetros, eles não estavam sendo descritos na gramática e ficavam definidos de forma estática no código

que interpretava cada indivíduo gerado. À vista disso, a gramática evoluiu novamente ficando muito mais próxima a final aqui proposta, se diferenciando no fato de que ainda era possível a rede iniciar com *pooling*; essa possibilidade foi removida após a observação de que os resultados das redes que iniciavam dessa forma eram mais baixos de que as demais; sendo também a motivação para a remoção da opção do *pooling* de mediana.

Adicionar mais camadas a rede faz com que mais características dos dados de entrada sejam extraídos. No entanto, isso só representa um atributo para até um certo número máximo de camadas, em outras palavras, adicionar infinitas camadas não vai necessariamente se mostrar eficaz, existe um valor limite para cada problema, tal qual, após isso, a rede tende a se moldar as características do treinamento, fazendo com que só se mostre eficaz na classificação dos dados utilizados no treinamento, o chamado *overfitting*. Esse estado faz com que normalmente sejam alcançados melhores acurácias no treinamento, mas uma performance muito ruim durante o teste. Dessa forma, sem que haja uma base de dados para treinamento grande o suficiente, as redes com maior profundidade tendem a convergir para o *overfitting* e apresentarem acurácia mais baixas, isso e o fato de que testes revelavam a baixa performance, encorajou a redução na quantidade de camadas.

4.3.2 Gramática Proposta

A gramática utilizada pela programação genética é a grande responsável pela delimitação do ambiente de buscas e pela aplicação de regras que se fazem necessárias ao problema. Durante os testes realizados no decurso da análise prévia, foi observado que alguns parâmetros envolvidos no processo de classificação através de CNN não refletiam em grandes alterações nos resultados finais enquanto outros se faziam importantes. Levando isso em conta, foi criada uma gramática capaz de englobar as alterações que mais influenciaram os resultados de acordo com o estudo e também com base no estado da arte de redes neurais convolutivas, fixando os valores das demais.

A gramática aqui proposta engloba não apenas a definição da arquitetura a ser utilizada mas também hiper-parâmetros. Para fins de facilidade de leitura, algumas produções possuem ainda dados fixos, referentes aos parâmetros adotados na camada, descritos na gramática. Conforme mostrado na Figura 39, a produção inicial é <network-architecture>.

Figura 39 – Gramática proposta no estudo.

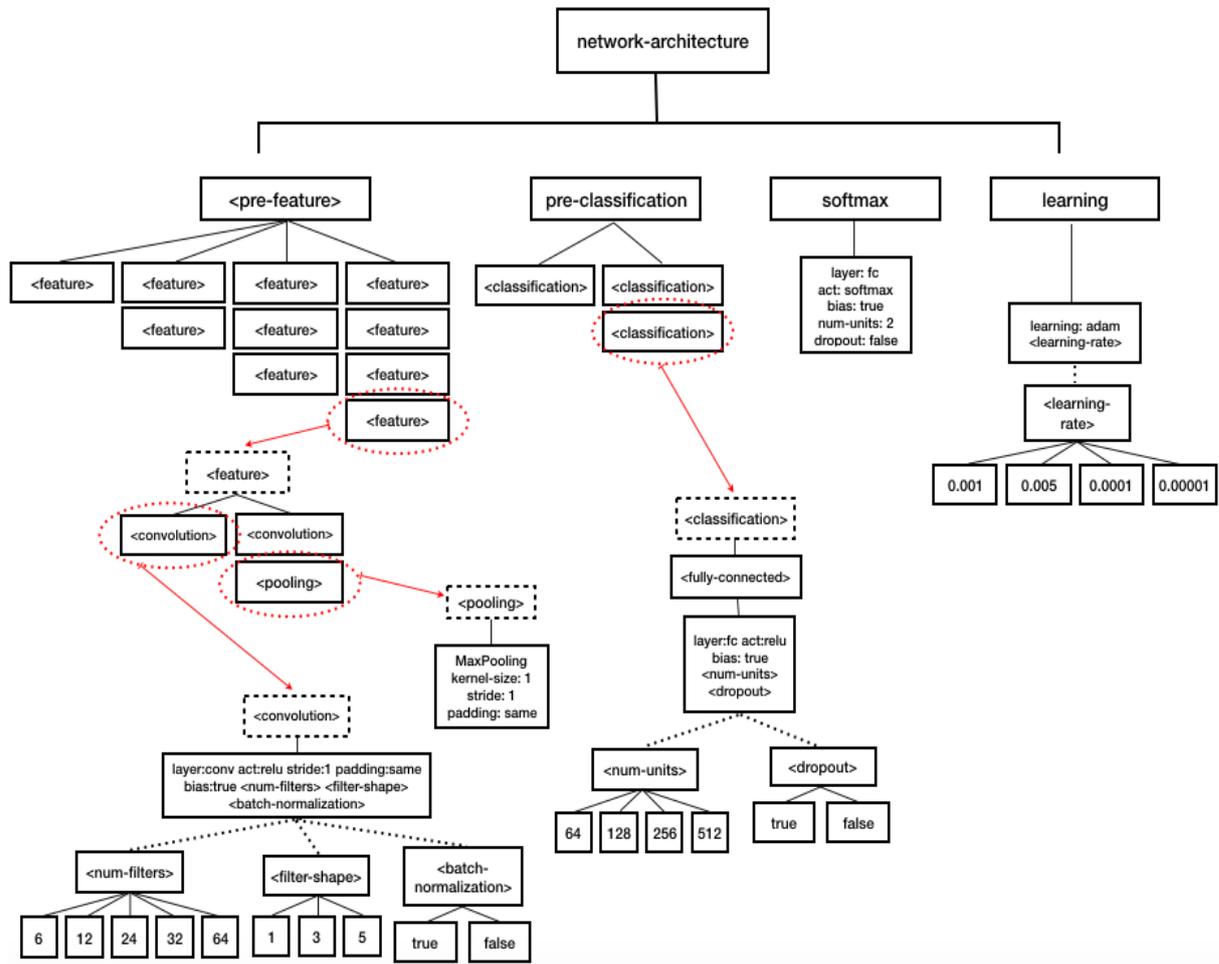
$\langle \text{network-architecture} \rangle$	\models	$\langle \text{pre-features} \rangle \langle \text{pre-classification} \rangle (\langle \text{softmax} \rangle) (\langle \text{learning} \rangle)$
$\langle \text{pre-features} \rangle$	\models	$\langle \text{features} \rangle$ $\langle \text{features} \rangle \langle \text{features} \rangle$ $\langle \text{features} \rangle \langle \text{features} \rangle \langle \text{features} \rangle$ $\langle \text{features} \rangle \langle \text{features} \rangle \langle \text{features} \rangle \langle \text{features} \rangle$
$\langle \text{features} \rangle$	\models	$(\langle \text{convolution} \rangle) (\langle \text{pooling} \rangle)$ $(\langle \text{convolution} \rangle)$
$\langle \text{convolution} \rangle$	\models	layer:conv num-filters: $\langle \text{num-filters} \rangle$ filter-shape: $\langle \text{filter-shape} \rangle$ stride:1 padding:same act:relu bias:True $\langle \text{batch-normalization} \rangle$
$\langle \text{batch-normalization} \rangle$	\models	batch-normalisation: $\langle \text{boolean} \rangle$
$\langle \text{pooling} \rangle$	\models	layer:pool-max kernel-size:1 stride: $\langle \text{stride} \rangle$ padding:same
$\langle \text{pre-classification} \rangle$	\models	$\langle \text{classification} \rangle$ $\langle \text{classification} \rangle \langle \text{classification} \rangle$
$\langle \text{classification} \rangle$	\models	$\langle \text{fully-connected} \rangle$
$\langle \text{fully-connected} \rangle$	\models	layer:fc act:relu num-units: $\langle \text{num-units} \rangle$ bias:True dropout: $\langle \text{boolean} \rangle$
$\langle \text{softmax} \rangle$	\models	layer:fc act:softmax num-units:2 bias:True dropout:False
$\langle \text{learning} \rangle$	\models	learning:adam learning-rate: $\langle \text{learning-rate} \rangle$
$\langle \text{num-filters} \rangle$	\models	6 12 24 32 64
$\langle \text{filter-shape} \rangle$	\models	1 3 5
$\langle \text{num-units} \rangle$	\models	64 128 256 512
$\langle \text{boolean} \rangle$	\models	True False
$\langle \text{learning-rate} \rangle$	\models	0.001 0.005 0.0001 0.00001

Fonte: A autora.

A representação da gramática é também mostrada na Figura 40 em forma de árvore com todas as possíveis escolhas e caminhos determinantes para a formação do indivíduo é exposto. Conforme explicado no sub-capítulo anterior, optou-se por aplicar uma gramática que permitisse a criações de arquiteturas de rede pequenas. Experimentos com redes muito profundas mostraram que os resultados não evoluíram muito apresentando acurácia na faixa dos 50 a 60% caracterizada pelo *overfitting* do dados. Dessa forma, o tamanho máximo de camadas convolutivas foi limitado para quatro, cada uma delas individualmente podendo ter *pooling* ou não; observe que não é mais permitido que as redes iniciem por uma camada de *pooling*, como citado anteriormente, o comportado foi experimentado durante análises mas não chegou a evoluir.

As camadas convolutivas possuem a possibilidade de serem seguidas de *batch*

Figura 40 – Árvore de derivação da gramática proposta.



Fonte: A autora.

normalization, variar o número de filtros e o tamanho do formato do filtro.

Todos os valores de terminais aplicados foram os mais comuns presentes na literatura. As ativações utilizadas são ReLU, o *pooling* é *MaxPooling* e o otimizador Adam.

Por envolver operações matemáticas mais simples, a função de ativação ReLU tem menos custo computacional de que Tanh ou Sigmoides (Glorot et al., 2011) (Arora et al., 2016). Outra característica importante é que a função não sofre com a dissipação de gradientes, que faz com que durante o treinamento os neurônios das camadas anteriores aprendam muito mais lentamente de que os das camadas posteriores.

Sabendo que a operação de *pooling* tem como objetivo reduzir o espaço de características analisado de forma que aquelas que são importantes para o problema sejam aprendidas, para esse estudo se optou por aplicar o *MaxPooling* pois ele é capaz de extrair as características mais importantes da imagem, enquanto o *AvgPooling* faz um processo de

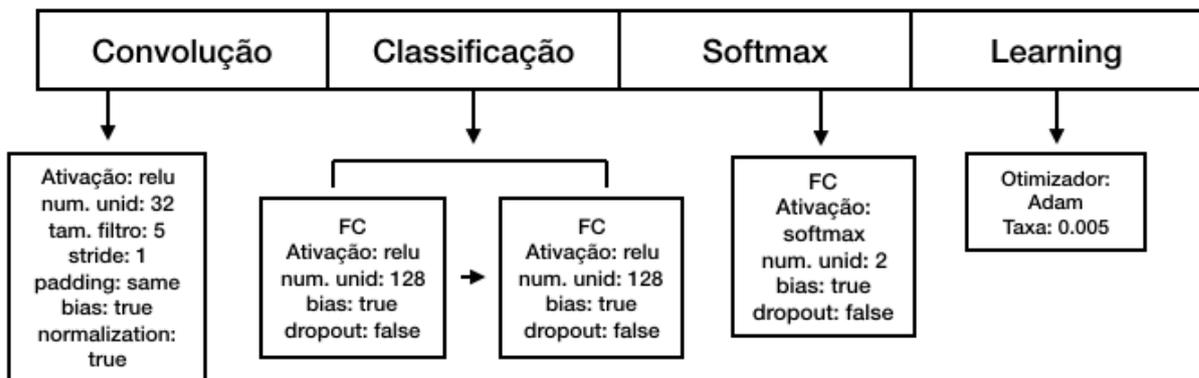
suavização, por vezes não fazendo boas capturas por considerar todos os valores e calcular sua média.

O otimizador Adam (KINGMA; BA, 2014) é popular na área de *deep learning* por sua capacidade de alcançar mais rapidamente bons resultados sendo também recomendado como padrão dentre os demais otimizadores presentes na literatura por frequentemente performar ligeiramente melhor (RUDER, 2016).

Na forma que é aqui apresentada, a gramática possui um total de 15.360 diferentes possibilidades de criação de indivíduos, sendo possível também criar arquitetura de redes conhecidas, como a LeNet. Dentre as mais de 15 mil opções, nem todas precisam necessariamente ser inspecionadas, sabe-se que o algoritmo evolutivo tem a capacidade de exploração das áreas mais promissoras para o problema.

Seguindo as regras de produção propostas, um exemplo de indivíduo que pode ser gerado é mostrado na Figura 41:

Figura 41 – Representação de um indivíduo produzido pela gramática.



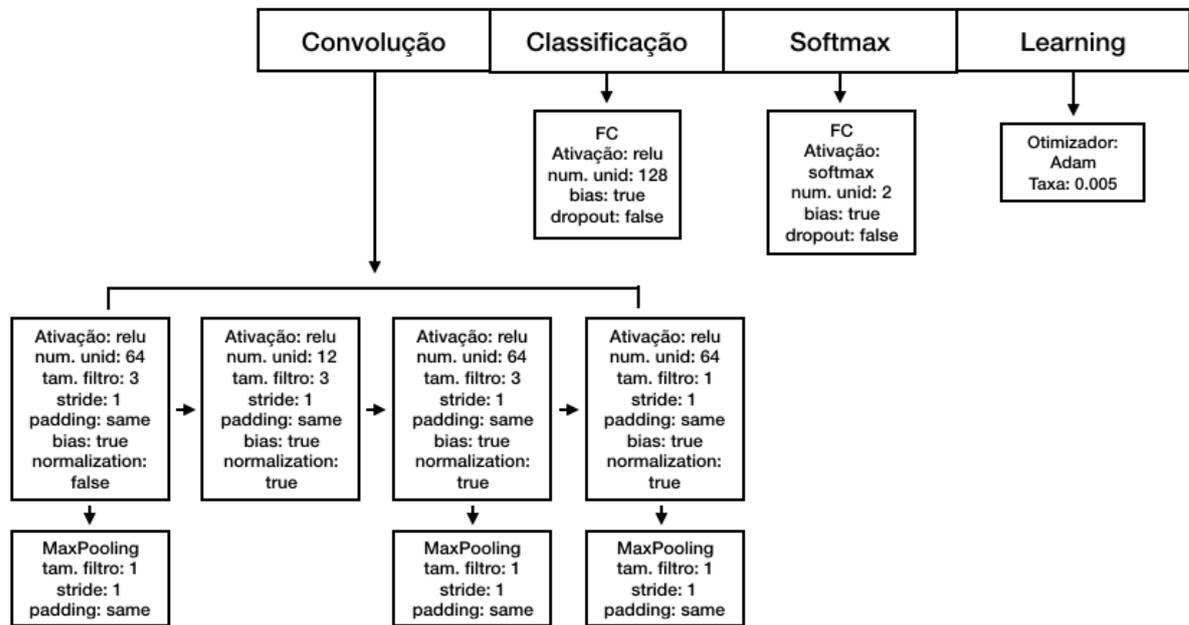
Fonte: A autora.

Esse é um indivíduo de tamanho pequeno, caso não tivesse duas camadas totalmente conectadas, ele formaria uma das variações da arquitetura com menor tamanho; a Figura 42 apresenta um outro exemplo de arquitetura maior e mais profunda.

O indivíduo da Figura 41 possui apenas uma camada convolutiva e ela é seguida pelo *batch normalization*; após a convolução é adicionado duas camadas totalmente conectadas todos com *dropout* e 128 neurônios; e por fim, a camada totalmente conectada de saída com apenas 2 neurônios referentes a classificação final do dado analisado. A Figura 43 mostra a arquitetura da CNN desse indivíduo citado no exemplo.

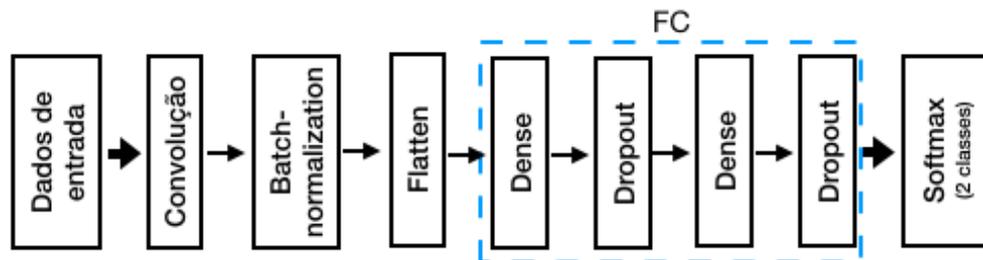
Como o problema abordado busca a otimização de redes neurais convolutivas, o

Figura 42 – Representação de um indivíduo produzido pela gramática com arquitetura maior e mais profunda.



Fonte: A autora.

Figura 43 – Exemplo de uma rede neural convolutiva gerada para um indivíduo produzido pela gramática.



Fonte: A autora.

cálculo do *fitness* utilizado no algoritmo é o valor da acurácia do conjunto de testes obtida pela execução da rede ao qual a gramática produziu. No entanto, sabe-se que CNNs são técnicas com padrão estocástico e, para mitigar a possibilidade do resultado final ser determinado por uma rede que aleatoriamente produziu uma boa acurácia com pequenas chances de ocorrência do sucesso novamente, o *fitness* é representado pela média aritmética de três execuções da rede.

Para interpretação de cada um dos indivíduos, que são gerados em forma de *string*, foi implementado um código capaz de traduzir cada um dos indivíduos em CNN executável, treiná-los e testá-los.

4.4 Considerações do Capítulo

Neste capítulo foram apresentados detalhes da técnica proposta, especialmente a gramática, grande diferencial trazido pelo estudo. Também foi toda a evolução dos estudos até a definição da gramática final bem como as discussões acerca das decisões quanto a definição do que vem determinar o espaço de buscas do problema; essas decisões influenciam diretamente o sucesso da técnica. Por fim, é mostrado exemplos de como são os indivíduos produzidos e como eles são interpretados transformando-os em uma rede neural executável. Detalhes de como o método proposto foi aplicado à computação evolutiva durante os testes podem ser encontrados nos próximos capítulos.

5 Metodologia

Este capítulo descreve brevemente as ferramentas e técnicas utilizadas no desenvolvimento do trabalho.

5.1 Base de Dados

O presente trabalho utiliza, para testes dos algoritmos replicados e da técnica proposta, imagens disponibilizadas gratuitamente pela base de dados do *International Skin Imaging Collaboration* (ISIC) através do *Melanoma Project* ([International Skin Imaging Collaboration, 2019](#)).

O ISIC dispõe de uma arquivo público de imagens de lesões de pele de diversas classificações de doenças buscando auxiliar a indústria e a academia na busca de soluções para a redução de mortalidade provocadas pelo melanoma e a realização de biópsias desnecessárias através da melhoria da acurácia e eficiência na detecção da doença em seus estágios iniciais ([International Skin Imaging Collaboration, 2019](#)).

A base de dados disponibilizadas pelo ISIC contém um total de 23.906 imagens de lesões de pele, dentre elas 2.286 estão catalogadas como melanoma e as demais são divididas entre outras classes de doenças, como por exemplo dermatofibroma, actinic keratosis, keratosis seborrheic, dentre outras.

Para os fins de interesse desse estudo, considerando também a limitação de capacidade computacional para execução dos experimentos e após observação da existência de repetição de imagens da mesma lesão, optou-se por pela utilização de 1.058 imagens divididas igualmente entre lesões benignas e melanoma. A seleção foi feita de forma aleatória, garantindo, no entanto, que não houvessem repetições de mesmas lesões.

Durante a execução dos testes das redes neurais, as imagens da base de dados não haviam sido redimensionadas e o tamanho delas, que além de grande eram desiguais, fez com que todo o processamento fosse custoso e demorado, por vezes extrapolando a quantidade de memória disponível nas máquinas utilizadas nos experimentos. Por conta disso, a base de dados do estudo passou um processo de redução de tamanho para uma altura máxima de 200 *pixels* mantendo suas devidas resoluções e proporções.

Após o redimensionamento da base, cada uma das classes que compõem a base

foi aleatoriamente dividida em 80% para utilização nos treinamentos das redes e 20% na validação.

5.2 Ambiente Experimental

5.2.1 Linguagem de Desenvolvimento

A linguagem de desenvolvimento adotada foi PythonTM. Trata-se de uma linguagem de programação funcional de código aberto de alto nível utilizada pelo *software* de mesmo nome. A linguagem permite desenvolver problemas matemáticos de forma rápida e eficaz, sendo amplamente utilizada por sua facilidade de abstrações.

5.2.2 Ambiente de Desenvolvimento

Para os experimentos aqui descritos, foi utilizado um notebook com processador 3 GHz Intel Core i7 e 8GB 1600 MHz DDR3 de memória e um computador com processador GPU e 16GB de memória RAM.

5.3 Acurácia da Classificação

A métrica determina a porcentagem de classificações previstas corretamente (ŠIMUNDIĆ, 2009). Ela é obtida pelo pegando o número de acertos (AC) e dividindo pelo número de todas as classificações realizadas (n), como mostra a Equação 5.1:

$$Acc = \frac{AC}{n} \quad (5.1)$$

Em outras palavras, a acurácia é obtida dividindo a soma de amostras que apresentam classificação verdadeira positiva (VP) somada as amostras verdadeiras negativas (VN), esse valor é o número de acertos, que é então dividido pelo valor total de amostras, como mostrado na Equação 5.2:

$$Acc = \frac{VP + VN}{n} \quad (5.2)$$

Essa é a métrica de avaliação mais comum para análise de problemas que envolvem classificação. Vale salientar que ela representa melhor a qualidade do problema quando utilizada com quantidades iguais de dados para cada uma das classes, dessa forma, todas as previsões e erros têm valores equivalentes.

Para o melhor entendimento dessa métrica, também são exibidos os valores correspondentes a frequências de classificação para cada classe através da matriz de confusão, como mostrado na Tabela 2.

Tabela 2 – Exemplo da matriz de confusão.

		Classificação Prevista	
		Benigna	Maligna
Classificação Original	Benigna	Verdadeiro Positivo	Falso Negativo
	Maligna	Falso Positivo	Verdadeiro Negativo

Fonte: A autora.

Na tabela, o valor verdadeiro positivo é a quantidade de imagens benignas que foi corretamente prevista. O falso negativo representa a quantidade de imagens previstas como malignas que possuem a classificação benigna. Falso verdadeiro é a previsão de lesão benigna para imagens de classificação maligna. O verdadeiro negativo é a quantidade de imagens corretamente previstas como malignas.

5.4 Técnicas Comparadas

Foram escolhidos alguns algoritmos de destaques propostos pela literatura para análise e comparação dos resultados quando aplicados as imagens dermatoscópicas utilizadas no trabalho proposto. Como o estudo abrange a otimização de redes neurais para classificação de lesões de pele utilizando programação genética orientada a gramática, a comparação da performance foi aplicada à técnicas de *deep learning* e PGOG. Todos eles foram executados com a mesma base de dados. As técnicas aplicadas são as seguintes:

1. LeNet-5 ([LECUN et al., 1998](#))
2. Mini GoogLeNet - conforme proposto por [Haenssle et al. \(2018\)](#)

3. Mini VGGNet-16 ([SIMONYAN; ZISSERMAN, 2014](#))
4. ResNet ([HE et al., 2016](#))
5. Transferência de Conhecimento do LeNet-5
6. Transferência de Conhecimento do ResNet
7. Transferência de Conhecimento do GoogLeNet (*Inception v3*) - conforme proposto por [Esteva et al. \(2017\)](#)
8. Replicação do algoritmo de evolução genética com uso de gramática proposto por [Assunção et al. \(2018\)](#)

Nas técnicas de transferência de conhecimento, foi extraído o conhecimento da execução das redes utilizando a base de dados ImageNet ([RUSSAKOVSKY et al., 2015](#)), realizando o treinamento com as imagens dermatoscópicas apenas a última camada da rede. Quanto ao algoritmo proposto por [Assunção et al. \(2018\)](#), a replicação foi feita conforme a própria descrição presente do estudo. Tanto a gramática quanto demais detalhes presentes sobre a execução foram aplicados de forma a se obter uma replicação próxima ao trabalho proposto pelos autores.

A escolha dos algoritmos para aplicação da técnica de transferência de conhecimento foi feita dado os altos resultados alcançados pelo LeNet com as imagens dermatoscópicas dentre as opções de arquiteturas conhecidas de CNN. O mesmo foi feito para o ResNet (ResNet-trans) pelo fato da arquitetura ser uma das mais comuns na técnica de transferência de aprendizado. Por fim, foi aplicada ao GoogLeNet - *Inception v3*, conforme proposto por [Esteva et al. \(2017\)](#).

5.5 Detalhes das Execuções

Para execução da evolução gramatical, foi empregado o *framework* PonyGE2 ([FENTON et al., 2017](#)) com a utilização da gramática aqui especificada.

As redes neurais produzidas em cada geração são executadas três vezes e o valor que vai representar o *fitness* da CNNs é a média aritmética da acurácia das três execuções do conjunto de teste. Em cada uma das execuções é realizado o treinamento e testes, havendo, portanto, o teste com pesos que podem variar de acordo com o treinamento. As configurações gerais e a arquitetura se mantêm as mesmas para todas as três execuções. Dada a característica estocástica, podemos, dessa forma, reduzir a probabilidade de uma

rede ruim ser considerada boa por apenas uma execução boa.

As CNNs foram executadas por 50 épocas e *batches* de tamanho 32, aplicando ainda redimensionamento das imagens para o tamanho 75×75 . Para implementação das redes foi utilizada a biblioteca do Keras (CHOLLET et al., 2015). Os valores escolhidos são frutos de análises que também consideraram, por exemplo, a capacidade computacional para execução do experimento.

A quantidade de dados utilizada para o treinamento das redes está diretamente ligado a qualidade das classificações; este estudo provem de uma base de dados com 848 imagens para o treinamento, no entanto, se observarmos que o ImageNet possui uma base de dados atualmente com mais de 14 milhões de imagens catalogadas, entendemos que a utilizada pelo estudo pode ser ampliada de forma a auxiliar a performance das redes. Em vista disso, foi aplicado em todas as CNNs a técnica de *data augmentation* operando variações aleatórias de *zoom* nas imagens.

Conforme especificado na gramática, as camadas totalmente conectadas podem ser seguidas por *dropout*, no entanto a taxa a ser aplicada não é apresentada na gramática. Toda a execução foi testada com dois valores bastante presentes na literatura: 0,5 e 0,25.

Os testes foram executados utilizando uma população composta por 50 indivíduos com inicialização do tipo *PIGrow* e executada por 50 gerações primeiramente aplicando o *dropout* de 0,5 e executada novamente com *dropout* de 0,25. Para esse estudo, o *fitness* apresenta valores entre 0 e 1, com o 1 sendo o melhor resultado, dessa forma, melhores acurácias levam a *fitness* maiores.

Os parâmetros utilizados na execução da evolução genética estão resumidos na Tabela 3.

Tabela 3 – Parâmetros das execuções da programação genética.

Parâmetro	Valor
Gerações	50
População	50
População inicial	<i>PIGrow</i>
Taxa de cruzamento	75%
Proporção da seleção	50%
Profundidade mínima da árvore	Nenhuma
Profundidade máxima da árvore	10
Mutação	<i>Int Flip Per Codon</i>
Taxa de mutação	1/Tamanho do genoma
Tipo de seleção	Torneio
Tamanho do torneio	2
elite size	1

Fonte: A autora.

6 Resultados

Este capítulo apresenta os resultados referentes a aplicação da gramática proposta apresentada no Capítulo 4 durante a execução dos experimentos produzidos conforme descritos no Capítulo 5. É feita também a comparação com resultados de algoritmos do estado da arte em redes neurais convolutivas.

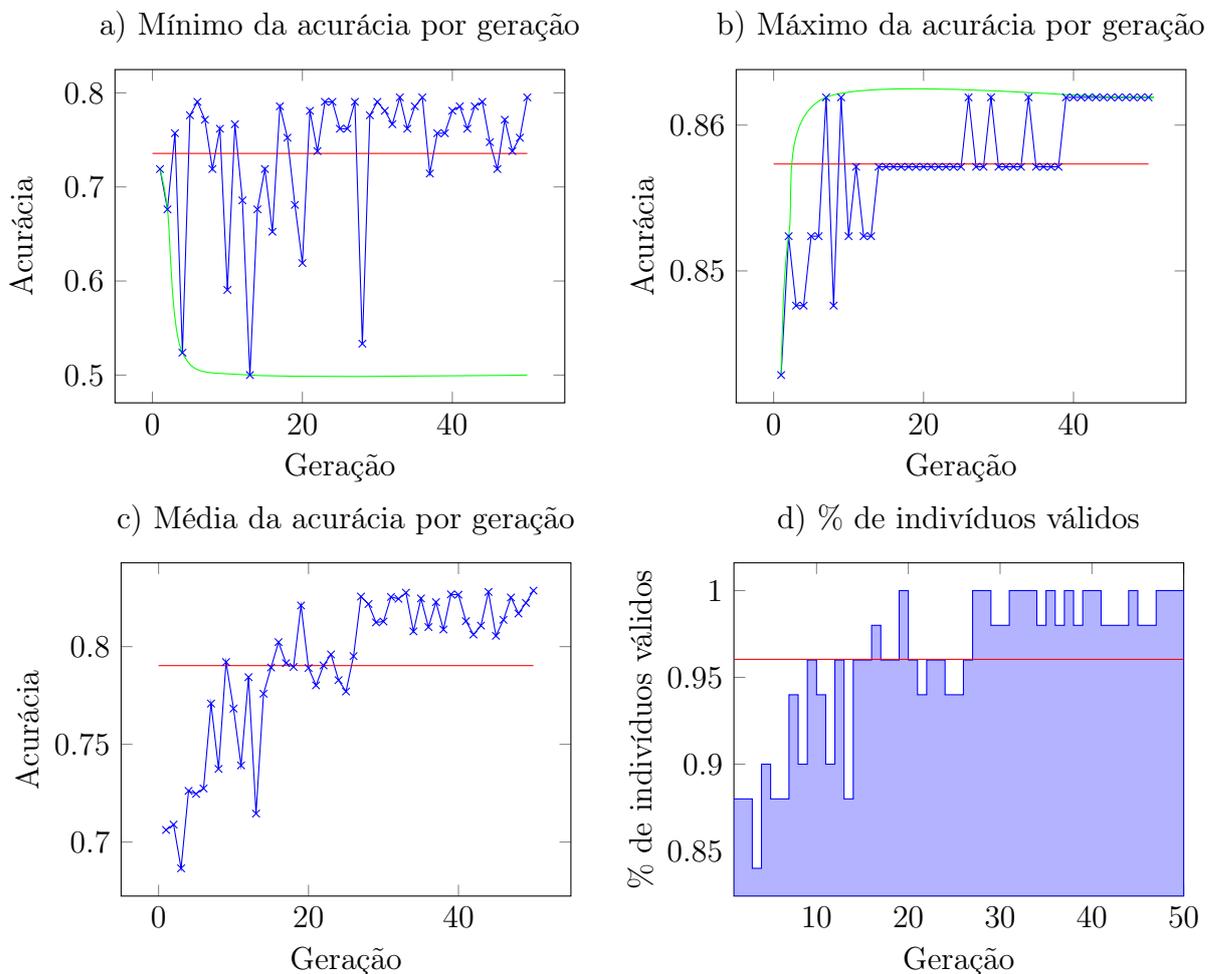
6.1 Resultados Experimentais

Conforme descrito nos capítulos anteriores, a técnica proposta foi executada por duas vezes cada uma com valores diferentes de *dropout*, 0,25 e 0,5, ambas alcançaram valores máximos muito próximos, no entanto os resultados aqui apresentados se referem a execução com o *dropout* no valor de 0,5; esse foi o valor escolhido por ter sido observado que a performance das redes com relação ao *loss* e acurácia terem sido mais próximos uns dos outros durante o treinamento/teste de que com o 0,25 que mostrava grandes disparidades de valores entre uma época e outra.

A Figura 44 mostra quatro gráficos com os resultados da execução da técnica proposta; em todos, a reta horizontal marcada em vermelho corresponde a média global dos resultados referentes a cada uma das análises. O gráfico (a) mostra a variação da acurácia mínima obtida pelas redes ao logo das gerações e a linha em verde destaca os menores resultados gerais. É possível notar que nas primeiras execuções existe uma variação grande quanto aos resultados, tendo uma oscilação de 50% a quase 80%; isso acontece devido ao fato da exploração por soluções estar percorrendo áreas não muito férteis para o problema. Na segunda metade dos experimentos notamos que os resultados mínimos já ficam acima dos 70%, significando que a evolução genética foi convergindo para melhores regiões. Um comportamento parecido ao observado em (a), acontece no gráfico (b). Nele também é interessante destacar que o algoritmo genético conseguiu alcançar bons resultados já antes da metade das gerações, denotado pela linha verde com a curvatura dos maiores resultados gerais; isso pode normalmente não ser um padrão da gramática e é um fator que vai depender da exploração da região de busca, nesse caso foi convergido em meados da metade das gerações. No gráfico (c) temos a média da acurácia de cada geração e fica mais claro o crescimento do resultados, apesar do final apresentar uma consistência

de altos resultados maiores confirmando que o algoritmo já convergiu para uma região boa. Por fim, temos o gráfico (d), deixando claro a porcentagem de indivíduos válidos por geração, no total foram observados 97 redes que por algum motivo não conseguiram ter sua acurácia mensurada; sabendo que a própria existência da gramática adiciona regras ao processo de criação de indivíduos, o que causa a anulação da rede é por exemplo, a máquina não conseguir executá-la por questões de memória ou processamento necessário. Também vale lembrar que variáveis importantes para a determinação da quantidade de parâmetros envolvidos no processo é determinado pela gramática, permitindo que exista a produção de redes que requerem o treinamento de milhares de parâmetros.

Figura 44 – Gráfico de representação dos resultados por gerações.

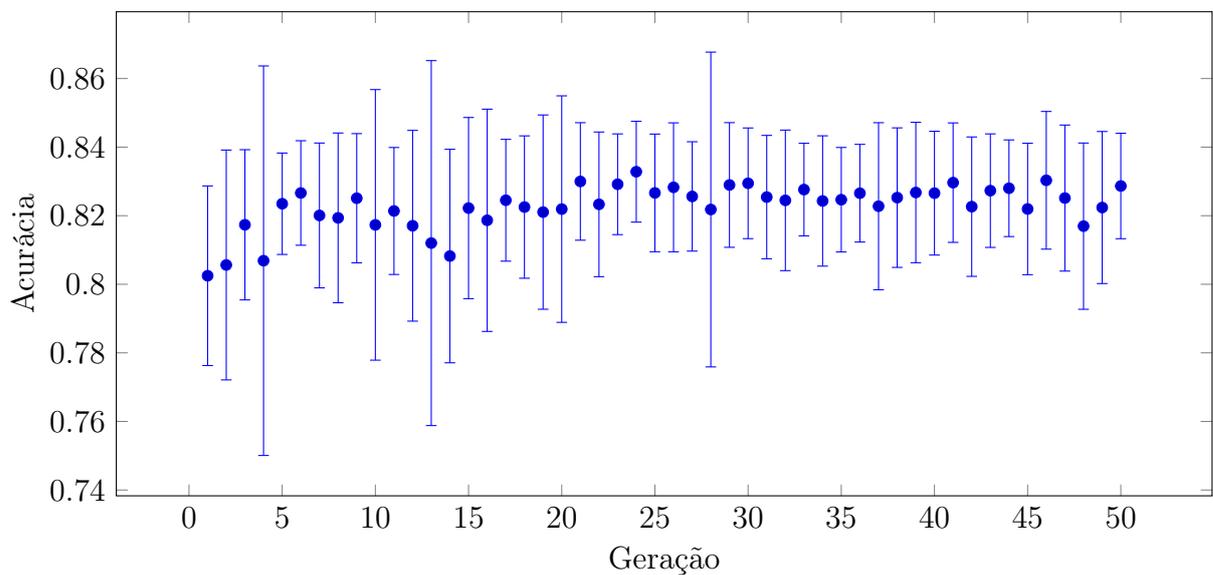


Fonte: A Autora.

A média da acurácia por geração é exposta na Figura 45 com o acréscimo do desvio padrão de cada geração, dessa forma é possível visualizar o nível de dispersão

das acurácias em cada ponto obtendo, assim, uma variação onde os valores podem ser considerados normais. Como já destacado durante a observação dos gráficos da Figura 44, aproximadamente a partir da metade da evolução do algoritmo os valores com desvio padrão alcançam uma regularidade maior resultante do fato de que a busca convergiu para uma boa região, não variando muito mais além desse ponto. Os valores de desvio não chegaram a comprometer 10% do valor da média da acurácia nem no pior caso que alcançou 5%, sendo a média da rede 73%; no melhor caso o desvio foi de 1%.

Figura 45 – Gráfico do desvio padrão da média da acurácia por geração.

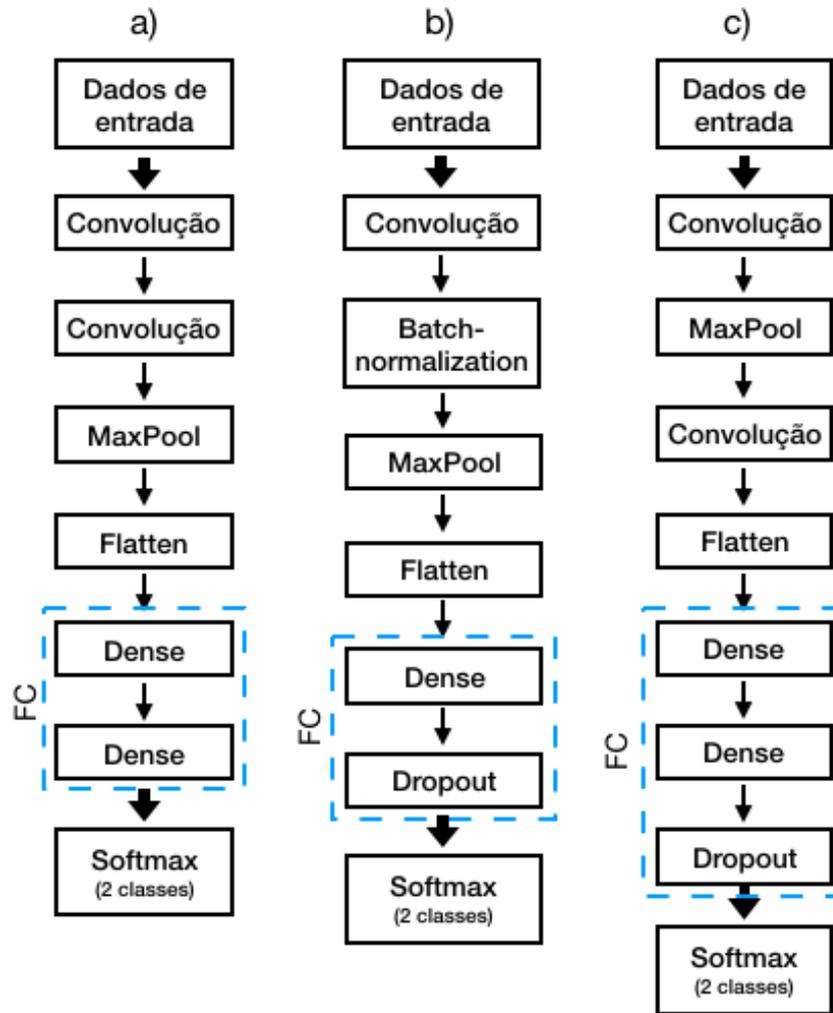


Fonte: A Autora.

Fazendo uma análise mais profunda do perfil das redes geradas ao início e ao final do processo evolutivo, foi notado que o tamanho das arquiteturas foi um fator de destaque de divergência. A Figura 46 apresenta a arquitetura de três CNNs produzidas a partir da decodificação de indivíduos concebidos na primeira geração; elas obtiveram acurácia próxima a média da geração.

A arquitetura da CNN também deve ser implementada pensando no tamanho da base de dados ao qual ela será aplicada. Esse estudo conta com uma base de dados com 1.058 imagens, fator que, no mundo do *deep learning*, não representa uma base muito grande. Sabe-se também que redes grandes podem apresentar melhores resultados com arquiteturas maiores, uma vez que a existência de mais camadas possibilita a extração de mais características dos dados, no entanto, existe uma quantidade de camadas adequada ao

Figura 46 – Exemplos de CNNs da primeira geração.

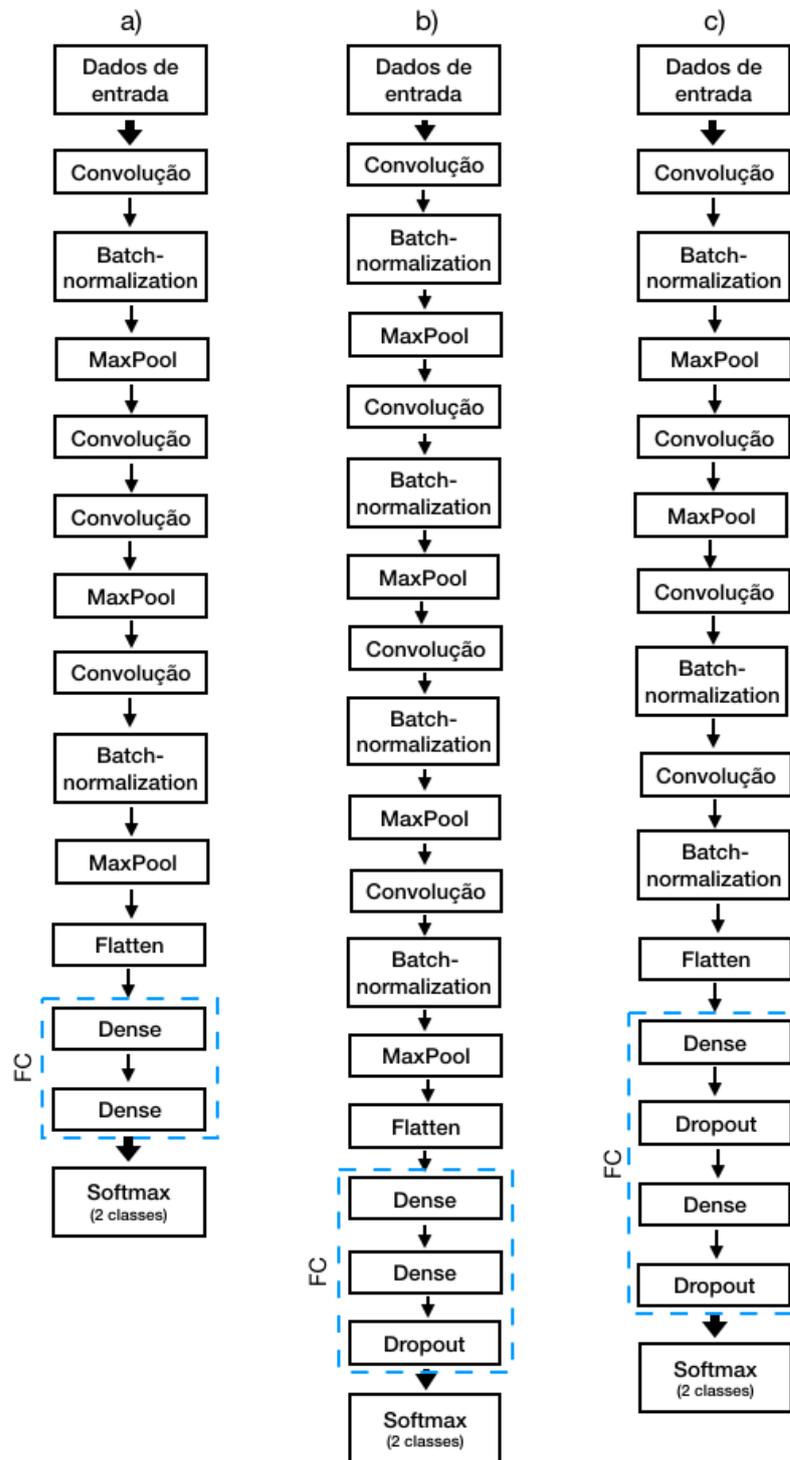


Fonte: A autora.

problema tal qual redes com profundidade maior de que o limiar suportado pelo problema tendem a apresentar uma redução na qualidade. Em contraste com muitas das redes da primeira geração, a Figura 47 mostra como o comportamento das redes divergiu da Figura 46, tendo na última geração redes mais profundas. A imagem destaca a arquitetura de três exemplos de redes que representam a última geração e obtiveram acurácias entre os valores mais altos. Vale também observar que todas elas possuem duas camadas totalmente conectadas, representada na imagem pelo retângulo “Dense”, além da *softmax*, responsável pela classificação final. A presença de mais de uma camada totalmente conectada permite a criação de uma hierarquia de características e a redução do *under-fitting* da rede ao mesmo tempo que aumenta o aprendizado das características dos dados analisados. Nas camadas totalmente conectadas todos os nós das camadas são conectados com todos os outros nós das camadas subsequentes para a tomada de decisões (classificação ou segmentação); a

adição de mais de uma camada totalmente conectada proporciona uma melhor interação e relação com todas as possíveis dependências a nível de característica.

Figura 47 – Exemplos de CNNs da última geração.

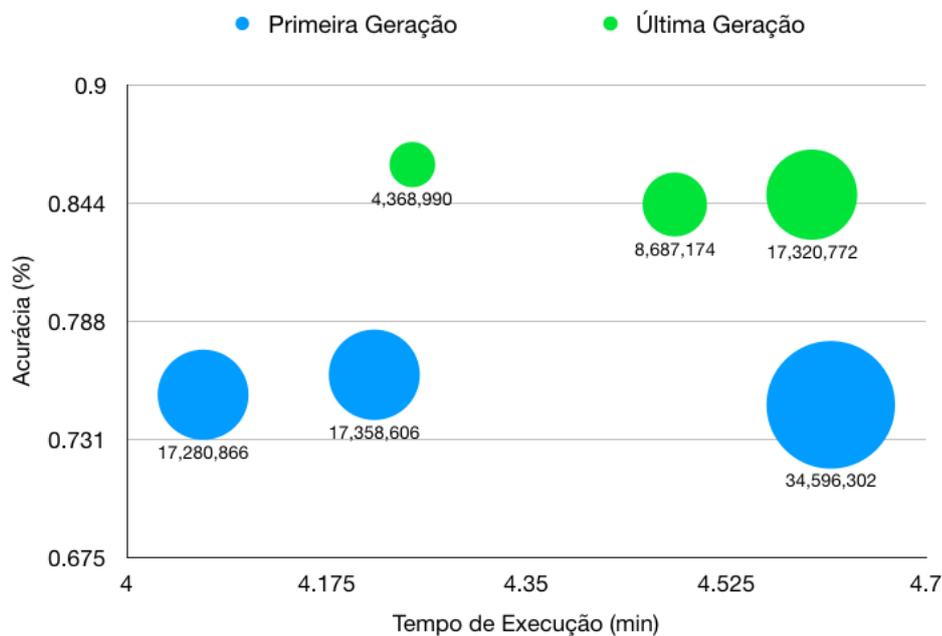


Fonte: A autora.

O comportamento dos indivíduos quanto ao tempo de execução pela acurácia, com visualização da quantidade de parâmetros da primeira e última geração destacados também

é mostrado na Figura 48; o tempo de execução se refere a todo o tempo de treinamento e testes e o tamanho de cada bolha é representado pela quantidade de parâmetros da rede, que também está representado pelos valores abaixo de cada bolha. Observando a imagem é possível inferir que a quantidade de parâmetros da rede não é necessariamente um fator determinante do tempo de execução da CNN, no entanto, é importante observar que a variação de tempo da mais rápida para a mais demorada foi de apenas pouco mais de 30 milissegundos, não sendo isso um valor tão grande.

Figura 48 – Resultados da acurácia em função do tempo de execução de CNNs da primeira (azul) e da última geração (verde) com visualização da quantidade de parâmetros de cada CNN.

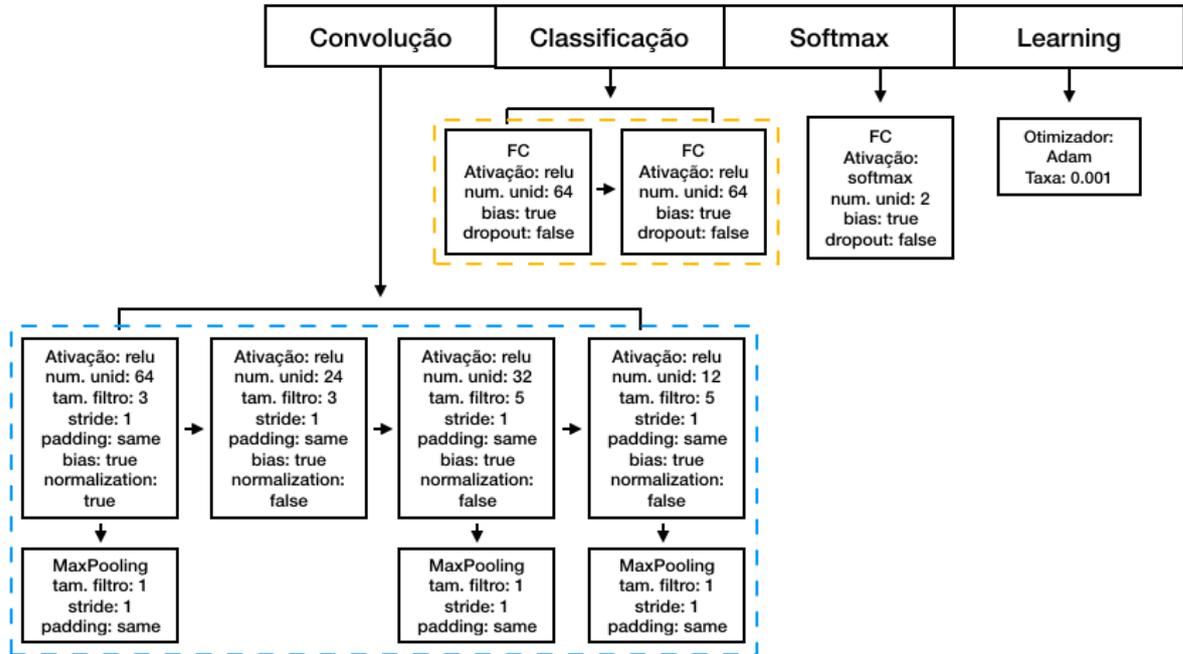


Fonte: A autora.

Quanto aos demais parâmetros, foi observado que os casos em que a rede demonstrou as piores performance, com acurácia na casa dos 50%, todos compartilhavam a taxa de aprendizado de 0,005 e seus valores de *loss* e *val_loss* iniciaram baixos mas cresceram significativamente durante o treinamento alcançando valores que não voltaram a cair.

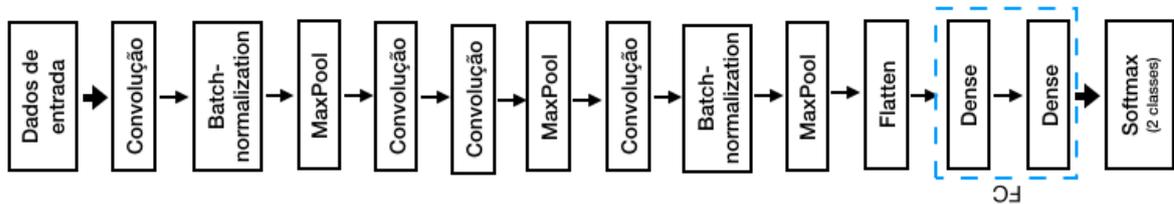
De toda a execução, o indivíduo que alcançou a maior média de acurácia das três execuções da CNN obteve 86,19%; a Figura 49 mostra sua representação em forma de árvore. Sua arquitetura é destacada na Figura 50 e os valores de acurácia e *loss* tanto do treinamento quanto da validação desse indivíduo são mostrados na Figura 51. É possível notar que houve uma variação bem maior dos valores de validação na primeira metade das épocas, até que foi alcançada uma certa estabilidade.

Figura 49 – Representação do melhor indivíduo produzido pela gramática.



Fonte: A autora.

Figura 50 – Arquitetura do indivíduo que apresentou maior performance.



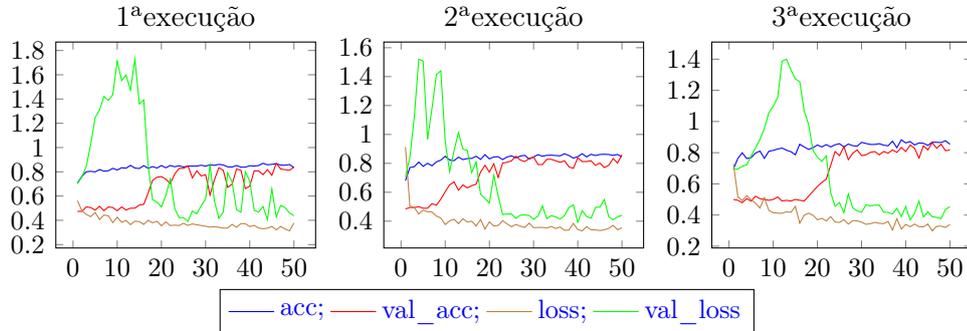
Fonte: A autora.

A matriz de confusão da melhor rede é demonstrada na Tabela 4, nela é possível notar que a maior quantidade de classificações previstas corretamente foram referentes a classe de imagens malignas.

6.2 Comparação dos Resultados com o Estado da Arte

Nesta sessão são apresentados os resultados obtidos na classificação de imagens dermatoscópicas por alguns algoritmos conhecidos na literatura. Todos eles foram implementados utilizando a biblioteca pública do Keras (CHOLLET et al., 2015) e foram treinadas e testadas com a mesma base de dados, respeitando também a divisão de imagens

Figura 51 – Gráfico com os valores de acurácia do treinamento (*acc*), acurácia da validação (*val_acc*), *loss* do treinamento (*loss*) e *loss* da validação (*val_loss*) ao longo das época referente as três execuções do melhor indivíduo.



Fonte: A autora.

Tabela 4 – Matriz de confusão.

		Classificação Prevista	
		Benigna	Maligna
Classificação Original	Benigna	83	22
	Maligna	7	98

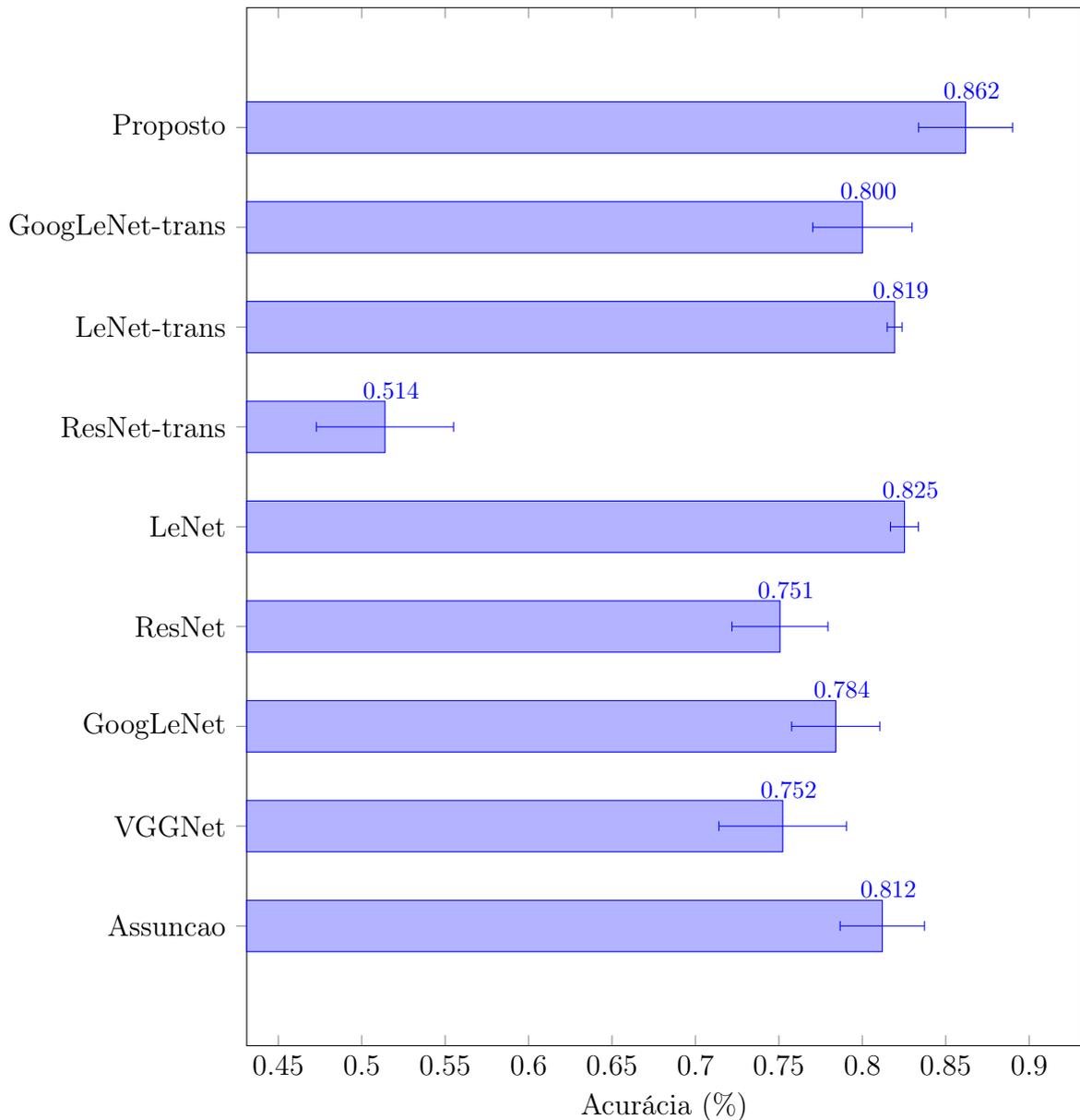
Fonte: A autora.

para treinamento e teste. Para viabilizar a comparação de forma igualitária, as redes foram executadas utilizando a mesma quantidade de épocas e os resultados correspondem a média aritmética da acurácia de três execuções consecutivas.

Objetivando a busca pela melhor técnica de detecção de melanoma de forma automática a partir de imagens de lesões de pele e sem a necessidade de interferência humana no processo, foi feita a análise da performance de alguns algoritmos do estado da arte na base de dados escolhida. A Figura 52 mostra os resultados obtidos por cada um dos algoritmos.

Como foi citado anteriormente, a quantidade de parâmetros envolvidos no processo de treinamento e teste da rede neural é influenciada pelo processo evolutivo, dessa forma, na Figura 53 mostra a quantidade de parâmetros do indivíduo que apresentou o melhor resultado de acurácia em comparação com as demais redes comparadas. A parte com mais ocorrências do gráfico está ampliada na representação do gráfico menor. o gráfico revela também a média do tempo de execução das redes, cada uma delas foi executada por

Figura 52 – Gráfico de representação dos resultados dos algoritmos do estado da arte. Todos os resultados apresentam o desvio padrão e o sufixo ‘-trans’ se refere a aplicação da rede utilizando a técnica de transferência de aprendizado.

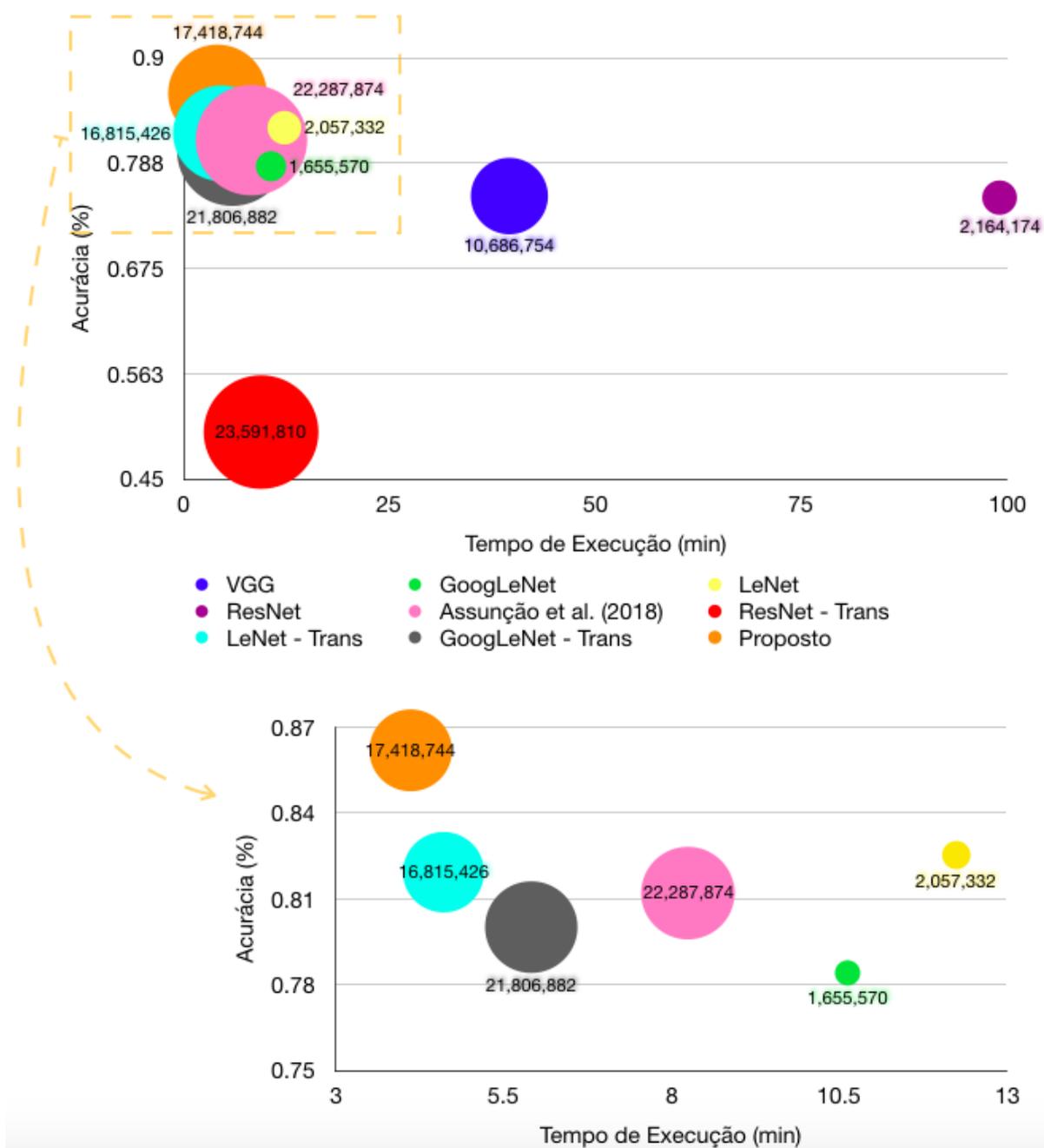


Fonte: A Autora.

três vezes seguidas e tanto o tempo quanto a acurácia representam a média aritmética de cada uma. Um fator interessante de observar nesse gráfico é que apesar de algumas redes, como a proposta, ter 17 milhões de parâmetros, o tempo de execução foi um dos menores quando comparado com os demais. A rede que utiliza a transferência de aprendizado do LeNet, representada pelo círculo em ciano, também tem um tempo pequeno e a uma grande quantidade de parâmetros, no entanto, vale lembrar que o próprio processo de transferência de aprendizado tem como característica não treinar as camadas que tiveram

o aprendizado de outro treinamento, isso faz com que o tempo total de execução seja menor. Para esse gráfico, todas as redes foram executadas na mesma máquina, uma de cada vez, de forma que estavam nas mesmas condições de processamento e memória sem a existência de competição por recursos.

Figura 53 – Resultados em função do tempo de execução pela acurácia com visualização da quantidade de parâmetros de cada CNN. O sufixo ‘-trans’ se refere a aplicação da rede utilizando a técnica de transferência de aprendizado. O segundo gráfico se refere a ampliação do gráfico principal na parte mais aglomerada das bolhas.



Fonte: A autora.

Analisando os indivíduos produzidos pelas execuções do algoritmo genético, foi possível notar que a gramática inclusive foi capaz de produzir redes conhecidas pelo estado da arte, como o LeNet. Como foi visto nos resultados das redes comparadas, mostrado na Figura 52, o LeNet foi o que apresentou melhor performance dentre as arquiteturas de CNN conhecidas; isso mostra que o algoritmo também foi capaz de explorar esse espaço e encontrou um resultado ainda melhor de que o LeNet.

6.3 Discussão dos Resultados

Foram escolhidas diversas técnicas do estado da arte de redes neurais convolutivas e evolução gramatical e o algoritmo proposto apresentou a melhor performance dentre elas além de não ter demandado um tempo de execução muito grande para a melhor rede.

Conforme mostrado na Figura 52, dentre as técnicas apresentadas, o LeNet obteve o resultado mais competitivo com a técnica proposta. O algoritmo proposto por Assunção et al. (2018) que também faz uso de gramática, de mesmo modo, apresentou boa acurácia, o que mostra que o uso da evolução gramatical pode ser adequada ao problema.

A gramática proposta foi capaz de encontrar regiões mais promissoras do espaço de buscas do problema alcançando acurácia maior de que 85% sem a necessidade de existência de um especialista em *deep learning* para a construção das redes.

A técnica apresentada por Assunção et al. (2018) foi replicada utilizando os mesmos parâmetros de quantidade de indivíduos e gerações do método proposto, no entanto, o tempo total de processamento necessário foi maior de que o algoritmo proposto; o algoritmo permite a criação de redes com arquiteturas maiores de que a proposta pelo estudo, no entanto, foi utilizada para o treinamento e teste das redes a base de dados do CIFAR-10, que é significativamente maior de que a de melanoma. Sabemos que qualquer camada adicionada à arquitetura da rede neural deve ter o objetivo de auxiliar no aprendizado de alguma aspecto parcial do problema analisado, mais camadas atuarão na extração de mais características. Normalmente, redes maiores necessitam de uma maior quantidade de dados para que apresentem boa acurácia, caso contrário, é comum que aconteça *over-fitting* dos dados pois a rede aprende muitas características específicas dos dados de treinamento, não classificando bem os demais dados. O aumento do tempo de execução total do algoritmo também é resultado do aumento do custo computacional necessário para o processamento

dos dados.

7 Conclusão

7.1 Considerações Finais e Contribuições do Trabalho

Sabendo que o Melanoma é um dos tipo de câncer de pele mais letal ([ATTIA et al., 2017](#)) e a quantidade de casos cresce todos os anos enquanto que o diagnóstico das lesões de pele realizado por médicos não evolui muito, este estudo propõe a utilização de algoritmos evolutivos com programação genética orientada a gramática para criar e otimizar arquiteturas de redes neurais convolutivas para classificação de lesões de pele melanomicas ou não visando auxiliar médicos no diagnóstico da doença.

Cada uma das CNNs produzidas pelo algoritmo foi treinada e testada buscando solucionar o problema de classificação das imagens utilizando imagens de lesões de pele da base de dados do ISIC ([International Skin Imaging Collaboration, 2019](#)).

O método proposto envolve a utilização da evolução genética baseada em gramática para a otimização de redes neurais convolutivas. Nesse contexto, é apresentada uma gramática capaz de explorar o espaço de busca do problema de forma a viabilizar a criação de indivíduos válidos capazes de serem qualitativamente analisados.

Além do processo de criação de indivíduos válidos, a avaliação do *fitness* é uma das etapas mais importantes em técnicas evolutivas, ela é responsável por determinar a qualidade da solução proposta e, conseqüentemente, guiar a exploração do ambiente por áreas consideradas promissoras. A medição do grau de adaptação das soluções aqui aplicada constitui na média aritmética da acurácia obtida pela classificação da rede neural resultante da execução da rede três vezes consecutivas com a mesma base de dados.

Os resultados mostraram que foi possível produzir arquiteturas de CNNs que gradualmente evoluíram alcançando acurácias maiores para a base de dados trabalhada. As redes passaram pelo processo de evolução durante a execução do algoritmo genético e isso indicou que o uso de gramática para otimização das redes pode produzir resultados promissores. Em comparação com os estudos para classificação de lesões de pele recentes, vários deles citados no estado da arte desse estudo, considerando a análise das imagens de lesões de pele sem que haja interferência humana de especialistas no processo, em outras palavras, totalmente automatizado, a gramática foi capaz de produzir redes que alcançaram resultados maiores de que os apresentados na literatura.

Os recursos computacionais e o tempo necessário para execução das soluções também são fatores importantes de se destacar. Embora os experimentos tenham sido realizados em um conjunto reduzido de recursos computacionais foi possível produzir boas soluções. Espera-se que melhores resultados possam ser obtidos com a utilização de sistemas computacionais de maior desempenho, capazes de processar inclusive base de dados potencialmente maiores, fator que pode impactar em melhora nos resultados de classificações.

Por fim, é possível concluir que a gramática em conjunto com a técnica de evolução genética tem um bom potencial para descobertas de soluções no domínio do problema apresentado. Dado que as soluções propostas não estão diretamente vinculadas a base de dados utilizada no estudo, a técnica pode ser também aplicada em outras áreas de conhecimentos para abordar diferentes problemas. A melhor solução pode não necessariamente ser a mesma para o problema do melanoma, mas se sabe que ela pode ser explorada no espaço de buscas.

Os resultados iniciais do trabalho foram publicados e está disponível nos Anais do XV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2018) com o título “A Grammar-based Genetic Programming Approach to Optimize Convolutional Neural Network Architectures” ([DINIZ et al., 2018](#)).

7.2 Trabalhos Futuros

Ao decorrer de todo o desenvolvimento dessa pesquisa, foram observadas possibilidades de novos trabalhos que podem contribuir para o estado da arte, sendo um deles a criação de uma gramática capaz de utilizar redes conhecidas, como as utilizadas nas análises comparativas, e incrementá-las ou mesmo permitir a junção de mais de uma dessas arquiteturas na mesma rede, criando assim variações de CNNs baseadas nas já conhecidas por produzir bons resultados.

Considerando a capacidade reduzida de recursos computacionais utilizados, uma outra possibilidade de trabalhos futuros é o estudo da performance com uma base de dados maior o suficiente para suportar a análise com redes de arquiteturas maiores. O algoritmo evolutivo também pode ser executado em uma quantidade maior de repetições.

Referências

- ABBASI, N. R.; SHAW, H. M.; RIGEL, D. S.; FRIEDMAN, R. J.; MCCARTHY, W. H.; OSMAN, I.; KOPF, A. W.; POLSKY, D. Early diagnosis of cutaneous melanoma: revisiting the abcd criteria. **Jama**, American Medical Association, v. 292, n. 22, p. 2771–2776, 2004.
- AIMA, A.; SHARMA, A. K. Predictive approach for melanoma skin cancer detection using cnn. **Available at SSRN 3352407**, 2019.
- AIN, Q. U.; XUE, B.; AL-SAHAF, H.; ZHANG, M. Genetic programming for skin cancer detection in dermoscopic images. In: IEEE. **2017 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2017. p. 2420–2427.
- ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2014.
- American Academy of Dermatology. **What to look for: The ABCDEs of melanoma**. 2013. Disponível em: <["https://www.aad.org/spot-skin-cancer/understanding-skin-cancer/how-do-i-check-my-skin/what-to-look-for"](https://www.aad.org/spot-skin-cancer/understanding-skin-cancer/how-do-i-check-my-skin/what-to-look-for)>.
- American Cancer Society. **What Is Melanoma Skin Cancer?** 2016. Último acesso 04 de Março de 2019. Disponível em: <["https://www.cancer.org/cancer/melanoma-skin-cancer/about/what-is-melanoma.html"](https://www.cancer.org/cancer/melanoma-skin-cancer/about/what-is-melanoma.html)>.
- _____. **Key Statistics for Melanoma Skin Cancer**. 2018. Último acesso 04 de Março de 2019. Disponível em: <["https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html"](https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html)>.
- ARGENZIANO, G.; CATRICALÀ, C.; ARDIGO, M.; BUCCINI, P.; SIMONE, P. D.; EIBENSCHUTZ, L.; FERRARI, A.; MARIANI, G.; SILIPO, V.; SPERDUTI, I. et al. Seven-point checklist of dermoscopy revisited. **British Journal of Dermatology**, Wiley Online Library, v. 164, n. 4, p. 785–790, 2011.
- ARGENZIANO, G.; KITTLER, H.; FERRARA, G.; RUBEGNI, P.; MALVEHY, J.; PUIG, S.; COWELL, L.; STANGANELLI, I.; GIORGI, V. D.; THOMAS, L. et al. Slow-growing melanoma: a dermoscopy follow-up study. **British Journal of Dermatology**, Wiley Online Library, v. 162, n. 2, p. 267–273, 2010.
- ARGENZIANO, G.; SOYER, H. P. Dermoscopy of pigmented skin lesions—a valuable tool for early. **The lancet oncology**, Elsevier, v. 2, n. 7, p. 443–449, 2001.
- ARGENZIANO, G.; SOYER, H. P.; CHIMENTI, S.; TALAMINI, R.; CORONA, R.; SERA, F.; BINDER, M.; CERRONI, L.; ROSA, G. D.; FERRARA, G. et al. Dermoscopy of pigmented skin lesions: results of a consensus meeting via the internet. **Journal of the American Academy of Dermatology**, Elsevier, v. 48, n. 5, p. 679–693, 2003.
- ARORA, R.; BASU, A.; MIANJY, P.; MUKHERJEE, A. Understanding deep neural networks with rectified linear units. **arXiv preprint arXiv:1611.01491**, 2016.
- ASHLOCK, D. **Evolutionary computation for modeling and optimization**. [S.l.]: Springer Science & Business Media, 2006.

ASSUNÇÃO, F.; LOURENÇO, N.; MACHADO, P.; RIBEIRO, B. Evolving the topology of large scale deep neural networks. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2018. p. 19–34.

ATTIA, M.; HOSSNY, M.; NAHAVANDI, S.; YAZDABADI, A. Skin melanoma segmentation using recurrent and convolutional neural networks. In: IEEE. **2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)**. [S.l.], 2017. p. 292–296.

AURICH, V.; WEULE, J. Non-linear gaussian filters performing edge preserving diffusion. In: **Mustererkennung 1995**. [S.l.]: Springer, 1995. p. 538–545.

AUTIER, P.; EPIMEL, D. for; GROUP, J.-F. E. M. C. Influence of sun exposures during childhood and during adulthood on melanoma risk. **International Journal of Cancer**, Wiley Online Library, v. 77, n. 4, p. 533–537, 1998.

BACK, T.; HAMMEL, U.; SCHWEFEL, H.-P. Evolutionary computation: Comments on the history and current state. **IEEE transactions on Evolutionary Computation**, IEEE, v. 1, n. 1, p. 3–17, 1997.

BANZHAF, W.; NORDIN, P.; KELLER, R.; FRANCONI, F. **GP—An Introduction; On the Automatic Evolution of Computer Programs and its Applications**. [S.l.]: Morgan Kaufmann San Francisco, 1998.

BARKER, J. **From the Frontline: How Deep Learning Plays Critical Role in Military Problem-Solving**. 2016. Último acesso 17 de Março de 2019. Disponível em: <<https://blogs.nvidia.com/blog/2016/06/29/deep-learning-6/>>.

BARRETT, W. A.; BATES, R. M.; GUSTAFSON, D. A.; COUCH, J. D. Compiler construction: theory and practice. SRA School Group, 1986.

BASHEER, I. A.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. **Journal of microbiological methods**, Elsevier, v. 43, n. 1, p. 3–31, 2000.

BASTOS, A. F. **Estudo de Métodos Classificatórios da Atenção Seletiva Auditiva a Partir de Potenciais Cerebrais Evocados**. 23 p. Tese (Doutorado) — Universidade Federal de Viçosa, 06 2014.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1798–1828, 2013.

BENGIO, Y.; GOODFELLOW, I. J.; COURVILLE, A. Deep learning. **Nature**, v. 521, p. 436–444, 2015.

BENGIO, Y. et al. Learning deep architectures for ai. **Foundations and trends® in Machine Learning**, Now Publishers, Inc., v. 2, n. 1, p. 1–127, 2009.

BOX, G. E. Evolutionary operation: A method for increasing industrial productivity. **Applied statistics**, JSTOR, p. 81–101, 1957.

BOX, G. E.; HUNTER, J. S. Condensed calculations for evolutionary operation programs. **Technometrics**, Taylor & Francis Group, v. 1, n. 1, p. 77–95, 1959.

BRAUN, R.; FRENCH, L.; SAURAT, J. Dermoscopy of pigmented lesions: a valuable tool in the diagnosis of melanoma. **Swiss medical weekly**, Citeseer, v. 134, n. 7/8, p. 83–90, 2004.

BRAUN, R. P.; RABINOVITZ, H. S.; OLIVIERO, M.; KOPF, A. W.; SAURAT, J. H. Pattern analysis: a two-step procedure for the dermoscopic diagnosis of melanoma. **Clinics in dermatology**, Elsevier, v. 20, n. 3, p. 236–239, 2002.

BRAY, F.; FERLAY, J.; SOERJOMATARAM, I.; SIEGEL, R. L.; TORRE, L. A.; JEMAL, A. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. **CA: a cancer journal for clinicians**, Wiley Online Library, v. 68, n. 6, p. 394–424, 2018.

BRIDLE, J. S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: **Neurocomputing**. [S.l.]: Springer, 1990. p. 227–236.

_____. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1990. p. 211–217.

CARDANI, D. Adventures in hsv space. **Laboratorio de Robótica, Instituto Tecnológico Autónomo de México**, Citeseer, 2001.

CARLI, P.; MANNONE, F.; GIORGI, V. de; NARDINI, P.; CHIARUGI, A.; GIANNOTTI, B. The problem of false-positive diagnosis in melanoma screening: the impact of dermoscopy. **Melanoma research**, LWW, v. 13, n. 2, p. 179–182, 2003.

CASCINELLI, N.; FERRARIO, M.; TONELLI, T.; LEO, E. A possible new tool for clinical diagnosis of melanoma: the computer. **Journal of the American Academy of Dermatology**, Elsevier, v. 16, n. 2, p. 361–367, 1987.

CELEBI, M. E.; WEN, Q.; IYATOMI, H.; SHIMIZU, K.; ZHOU, H.; SCHAEFER, G. **A state-of-the-art survey on lesion border detection in dermoscopy images**. [S.l.]: Boca Raton, CRC Press, 2015. 97–129 p.

Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia - CEP SRM - UFRGS. **Estrutura das Imagens de SR**. 2016. Último acesso 02 de Março de 2019. Disponível em: <"<http://www.ufrgs.br/engcart/PDASR/estrim.html>">.

CHIROMA, H.; ABDULKAREEM, S.; ABUBAKAR, A.; HERAWAN, T. Neural networks optimization through genetic algorithm searches: a review. **Appl. Math**, v. 11, n. 6, p. 1543–1564, 2017.

CHOLLET, F. et al. **Keras**. 2015. <<https://keras.io>>.

CODELLA, N. C.; GUTMAN, D.; CELEBI, M. E.; HELBA, B.; MARCHETTI, M. A.; DUSZA, S. W.; KALLOO, A.; LIOPYRIS, K.; MISHRA, N.; KITTLER, H. et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: **IEEE. 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)**. [S.l.], 2018. p. 168–172.

DAHL, G. E.; SAINATH, T. N.; HINTON, G. E. Improving deep neural networks for lvsr using rectified linear units and dropout. In: IEEE. **Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on**. [S.l.], 2013. p. 8609–8613.

DARWIN, C. On the origin of species by means of natural selection. 1859. **Murray, London**, v. 502, 1859.

_____. **The origin of species: by means of natural selection or the preservation of favoured races in the struggle for life**. [S.l.]: Bantam Classics, 1999. 126-127 p.

Deep Learning Tutorials. **Tutorials**. 2015. Último acesso 04 de Março de 2019. Disponível em: <http://swinghu.github.io/deep_learning/2015/10/09/dl-tutorials.html>.

DeepAI. **What is Max Pooling?** 2019. Último acesso 10 de Janeiro de 2019. Disponível em: <<https://deepai.org/machine-learning-glossary-and-terms/max-pooling>>.

Dermoscopy. **Dermoscopy Tutorial**. 2016. Último acesso 02 de Março de 2019. Disponível em: <<http://www.dermoscopy.org/atlas/base.htm>>.

DHUNGEL, N.; CARNEIRO, G.; BRADLEY, A. P. Automated mass detection in mammograms using cascaded deep learning and random forests. In: IEEE. **2015 international conference on digital image computing: techniques and applications (DICTA)**. [S.l.], 2015. p. 1–8.

DINIZ, J. B.; CORDEIRO, F. R.; MIRANDA, P. B. C.; SILVA, L. A. T. da. A grammar-based genetic programming approach to optimize convolutional neural network architectures. In: **Anais do XV Encontro Nacional de Inteligência Artificial e Computacional**. Porto Alegre, RS, Brasil: SBC, 2018. p. 82–93. Disponível em: <<http://portaldeconteudo.sbc.org.br/index.php/eniac/article/view/4406>>.

DL4J. **Introduction to Deep Neural Networks**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <<https://deeplearning4j.org/neuralnet-overview#concept>>.

DUAN, L.; XU, D.; TSANG, I. Learning with augmented features for heterogeneous domain adaptation. **arXiv preprint arXiv:1206.4660**, 2012.

DUDA, R. O.; HART, P. E. et al. **Pattern classification and scene analysis**. [S.l.]: Wiley New York, 1973. v. 3.

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**, Nature Publishing Group, v. 542, n. 7639, p. 115, 2017.

FAGAN, D.; FENTON, M.; O'NEILL, M. Exploring position independent initialisation in grammatical evolution. In: IEEE. **2016 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2016. p. 5060–5067.

FENTON, M.; MCDERMOTT, J.; FAGAN, D.; FORSTENLECHNER, S.; HEMBERG, E.; O'NEILL, M. Ponyge2: Grammatical evolution in python. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2017. p. 1194–1201.

FOGEL, D. B. **Evolutionary computation: toward a new philosophy of machine intelligence**. [S.l.]: John Wiley & Sons, 1995. v. 1.

FORSTENLECHNER, S.; NICOLAU, M.; FAGAN, D.; O'NEILL, M. Grammar design for derivation tree based genetic programming systems. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 2016. p. 199–214.

FRANÇA, R. N. **File:Representação de uma imagem digital - Matriz de pixel**. 2016. Último acesso 02 de Março de 2019. Disponível em: <["https://commons.wikimedia.org/wiki/File:Representa%C3%A7%C3%A3o_de_uma_imagem_digital_-_Matriz_de_pixel.jpg"](https://commons.wikimedia.org/wiki/File:Representa%C3%A7%C3%A3o_de_uma_imagem_digital_-_Matriz_de_pixel.jpg)>.

FRASER, A. S. Simulation of genetic systems by automatic digital computers. **Australian Journal of Biological Sciences**, CSIRO, v. 10, n. 4, p. 484–491, 1957.

FRIEDBERG, R. M. A learning machine: Part i. **IBM Journal of Research and Development**, IBM, v. 2, n. 1, p. 2–13, 1958.

FRIEDMAN, R. J.; RIGEL, D. S.; KOPF, A. W. Early detection of malignant melanoma: the role of physician examination and self-examination of the skin. **CA: a cancer journal for clinicians**, Wiley Online Library, v. 35, n. 3, p. 130–151, 1985.

FROSSARD, D. **VGG in TensorFlow**. 2016. Último acesso 14 de Fevereiro de 2018. Disponível em: <["https://www.cs.toronto.edu/~frossard/post/vgg16/"](https://www.cs.toronto.edu/~frossard/post/vgg16/)>.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological cybernetics**, Springer, v. 36, n. 4, p. 193–202, 1980.

GERSHENSON, C. Artificial neural networks for beginners. **arXiv preprint cs/0308031**, 2003.

GIOTIS, I.; MOLDERS, N.; LAND, S.; BIEHL, M.; JONKMAN, M. F.; PETKOV, N. Med-node: a computer-assisted melanoma diagnosis system using non-dermoscopic images. **Expert systems with applications**, Elsevier, v. 42, n. 19, p. 6578–6585, 2015.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: **Proceedings of the fourteenth international conference on artificial intelligence and statistics**. [S.l.: s.n.], 2011. p. 315–323.

GOLDBERG, D. E.; DEB, K. A comparative analysis of selection schemes used in genetic algorithms. In: **Foundations of genetic algorithms**. [S.l.]: Elsevier, 1991. v. 1, p. 69–93.

GONZALEZ, R. C.; WOODS, R. E. et al. **Digital image processing**. [S.l.]: Prentice hall Upper Saddle River, NJ., 2002.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. [S.l.]: MIT press, 2016. v. 1. <<http://www.deeplearningbook.org>>.

_____. **Deep learning - 6.2.2.3 Softmax Units for Multinoulli Output Distributions**. [S.l.]: MIT press, 2016. v. 1. 180–184 p. <<http://www.deeplearningbook.org>>.

GRUAU, F. On using syntactic constraints with genetic programming. In: MIT PRESS. **Advances in genetic programming**. [S.l.], 1996. p. 377–394.

HAENSSLE, H.; FINK, C.; SCHNEIDERBAUER, R.; TOBERER, F.; BUHL, T.; BLUM, A.; KALLOO, A.; HASSEN, A. B. H.; THOMAS, L.; ENK, A. et al. Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. **Annals of Oncology**, Oxford University Press, v. 29, n. 8, p. 1836–1842, 2018.

HALICEK, M.; LU, G.; LITTLE, J. V.; WANG, X.; PATEL, M.; GRIFFITH, C. C.; EL-DEIRY, M. W.; CHEN, A. Y.; FEI, B. Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging. **Journal of biomedical optics**, International Society for Optics and Photonics, v. 22, n. 6, p. 060503, 2017.

HALPERN, A. C.; LIEB, J. A. Early melanoma diagnosis: a success story that leaves room for improvement. **Current opinion in oncology**, LWW, v. 19, n. 2, p. 109–115, 2007.

HAN, J.; MORAGA, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In: SPRINGER. **International Workshop on Artificial Neural Networks**. [S.l.], 1995. p. 195–201.

HAREL, M.; MANNOR, S. Learning from multiple outlooks. **arXiv preprint arXiv:1005.0027**, 2010.

HAYKIN, S. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice Hall PTR, 1994.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

HUBEL, D. H.; WIESEL, T. N. Receptive fields of single neurones in the cat's striate cortex. **The Journal of physiology**, Wiley Online Library, v. 148, n. 3, p. 574–591, 1959.

_____. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of physiology**, Wiley Online Library, v. 160, n. 1, p. 106–154, 1962.

_____. Receptive fields and functional architecture of monkey striate cortex. **The Journal of physiology**, Wiley Online Library, v. 195, n. 1, p. 215–243, 1968.

_____. **Brain and visual perception: the story of a 25-year collaboration**. [S.l.]: Oxford University Press, 2004.

HUYNH, B. Q.; LI, H.; GIGER, M. L. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. **Journal of Medical Imaging**, International Society for Optics and Photonics, v. 3, n. 3, p. 034501, 2016.

Instituto Nacional de Câncer - INCA. **Câncer de pele melanoma**. 2018. Último acesso 04 de Março de 2019. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>>.

_____. **Estatísticas de câncer**. 2018. Último acesso 04 de Março de 2019. Disponível em: <["https://www.inca.gov.br/numeros-de-cancer"](https://www.inca.gov.br/numeros-de-cancer)>.

International Agency for Research on Cancer. Solar and ultraviolet radiation. monographs on the evaluation of the carcinogenic risk of chemicals to humans. **IARC, Lyon**, v. 55, 1992.

International Skin Imaging Collaboration. **ISIC Archive**. 2019. Último acesso 06 de Março de 2019. Disponível em: <["https://www.isic-archive.com"](https://www.isic-archive.com)>.

JONG, K. D.; FOGEL, D.; SCHWEFEL, H.-P. A history of evolutionary computation. In: _____. [S.l.: s.n.], 1997. p. A2.3:1–12.

KALMAN, B. L.; KWASNY, S. C. Why tanh: choosing a sigmoidal function. In: IEEE. **[Proceedings 1992] IJCNN International Joint Conference on Neural Networks**. [S.l.], 1992. v. 4, p. 578–581.

KAYALIBAY, B.; JENSEN, G.; SMAGT, P. van der. Cnn-based segmentation of medical imaging data. **arXiv preprint arXiv:1701.03056**, 2017.

KHVOSTIKOV, A.; ADERGHAL, K.; BENOIS-PINEAU, J.; KRYLOV, A.; CATHELIN, G. 3d cnn-based classification using smri and md-dti images for alzheimer disease studies. **arXiv preprint arXiv:1801.05968**, 2018.

KIMBALL, A. B.; JR, J. S. R. The us dermatology workforce: a specialty remains in shortage. **Journal of the American Academy of Dermatology**, Elsevier, v. 59, n. 5, p. 741–745, 2008.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KIRSNER, R. S.; WILKINSON, J. D.; MA, F.; PACHECO, H.; FEDERMAN, D. G. The association of medicare health care delivery systems with stage at diagnosis and survival for patients with melanoma. **Archives of dermatology**, American Medical Association, v. 141, n. 6, p. 753–757, 2005.

KITTLER, H.; PEHAMBERGER, H.; WOLFF, K.; BINDER, M. Diagnostic accuracy of dermoscopy. **The lancet oncology**, Elsevier, v. 3, n. 3, p. 159–165, 2002.

KOROTKOV, K.; GARCIA, R. Computerized analysis of pigmented skin lesions: a review. **Artificial intelligence in medicine**, Elsevier, v. 56, n. 2, p. 69–90, 2012.

KOZA, J. R. **Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems**. [S.l.]: Stanford University, Department of Computer Science Stanford, CA, 1990. v. 34.

_____. **Genetic programming: on the programming of computers by means of natural selection**. [S.l.]: MIT press, 1992. v. 1.

_____. Genetic programming as a means for programming computers by natural selection. **Statistics and computing**, Springer, v. 4, n. 2, p. 87–112, 1994.

KRIZHEVSKY, A.; HINTON, G. Learning multiple layers of features from tiny images. Citeseer, 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural computation**, MIT Press, v. 1, n. 4, p. 541–551, 1989.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Y.; JACKEL, L.; BOTTOU, L.; BRUNOT, A.; CORTES, C.; DENKER, J.; DRUCKER, H.; GUYON, I.; MULLER, U.; SACKINGER, E. et al. Comparison of learning algorithms for handwritten digit recognition. In: PERTH, AUSTRALIA. **International conference on artificial neural networks**. [S.l.], 1995. v. 60, p. 53–60.

LI, Q.; CAI, W.; WANG, X.; ZHOU, Y.; FENG, D. D.; CHEN, M. Medical image classification with convolutional neural network. In: IEEE. **2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)**. [S.l.], 2014. p. 844–848.

LIU, S.; LIU, S.; CAI, W.; PUJOL, S.; KIKINIS, R.; FENG, D. Early diagnosis of alzheimer's disease with deep learning. In: IEEE. **2014 IEEE 11th international symposium on biomedical imaging (ISBI)**. [S.l.], 2014. p. 1015–1018.

Machine Learning Crash Course - Google Developers. **ML Practicum: Image Classification**. 2019. Último acesso 28 de Abril de 2019. Disponível em: <<https://developers.google.com/machine-learning/practica/image-classification/preventing-overfitting>>.

MASOOD, A.; AL-JUMAILY, A. A. Computer aided diagnostic support system for skin cancer: a review of techniques and algorithms. **International journal of biomedical imaging**, Hindawi Publishing Corporation, v. 2013, 2013.

MathWorks®. **Convolutional Neural Network**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <<https://www.mathworks.com/discovery/convolutional-neural-network.html>>.

MAYER, J. Systematic review of the diagnostic accuracy of dermatoscopy in detecting malignant melanoma. **The Medical Journal of Australia**, v. 167, n. 4, p. 206–210, 1997.

MAYR, E. **Toward a new philosophy of biology: Observations of an evolutionist**. [S.l.]: Harvard University Press, 1988.

MCCARTHY, J.; LEVIN, M. I. **LISP 1.5 programmer's manual**. [S.l.]: MIT press, 1965.

MCCONNELL, J. **Computer Graphics: Theory Into Practice**. 1ª edição. ed. [S.l.]: Jones & Bartlett Learning, 2005. ISBN 978-0763722500.

MCKAY, R. I.; HOAI, N. X.; WHIGHAM, P. A.; SHAN, Y.; O'NEILL, M. Grammar-based genetic programming: a survey. **Genetic Programming and Evolvable Machines**, Springer, v. 11, n. 3-4, p. 365–396, 2010.

MENZIES, S. The menzies method. **Atlas of dermoscopy**. Taylor & Francis, London, p. 99–109, 2005.

MENZIES, S. W. A method for the diagnosis of primary cutaneous melanoma using surface microscopy. **Dermatologic clinics**, Elsevier, v. 19, n. 2, p. 299–305, 2001.

MICHALEWICZ, Z. Evolution strategies and other methods. In: **Genetic algorithms+ data structures= evolution programs**. [S.l.]: Springer, 1996. p. 159–177.

MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. **Machine learning: An artificial intelligence approach**. [S.l.]: Springer Science & Business Media, 2013.

MISHRA, N. K.; CELEBI, M. E. An overview of melanoma detection in dermoscopy images using image processing and machine learning. **arXiv preprint arXiv:1601.07843**, 2016.

MITCHELL, T. M. Machine learning. 1997. **Burr Ridge, IL: McGraw Hill**, v. 45, n. 37, p. 870–877, 1997.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012.

MUSCI, M. **Representação Digital de Imagens**. 2016. Último acesso 22 de Setembro de 2016. Disponível em: <["http://www.musci.com.br/multimedia/ImagensDesenhos3D.pdf"](http://www.musci.com.br/multimedia/ImagensDesenhos3D.pdf)>.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.

NASR-ESFAHANI, E.; SAMAVI, S.; KARIMI, N.; SOROUSHMEHR, S. M. R.; JAFARI, M. H.; WARD, K.; NAJARIAN, K. Melanoma detection by analysis of clinical images using convolutional neural network. In: IEEE. **2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S.l.], 2016. p. 1373–1376.

National Cancer Institute - NIH - PubMed Health Glossary. **Axons**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <["https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0023687/"](https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0023687/)>.

_____. **Dendrites**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <["https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0023688/"](https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0023688/)>.

_____. **Neurons (Nerve Cells)**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <["https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024269/"](https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024269/)>.

_____. **Synapses**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <["https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024276/"](https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024276/)>.

- NICOLAU, M. Understanding grammatical evolution: initialisation. **Genetic Programming and Evolvable Machines**, Springer, v. 18, n. 4, p. 467–507, 2017.
- OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. **Pattern recognition**, Elsevier, v. 29, n. 1, p. 51–59, 1996.
- OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, n. 7, p. 971–987, 2002.
- O'NEIL, R. et al. Convolution operators and $l(p, q)$ spaces. **Duke Mathematical Journal**, Duke University Press, v. 30, n. 1, p. 129–142, 1963.
- PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 22, n. 10, p. 1345–1359, 2010.
- PAPPA, G. L.; FREITAS, A. A. Evolving rule induction algorithms with multi-objective grammar-based genetic programming. **Knowledge and information systems**, Springer, v. 19, n. 3, p. 283–309, 2009.
- PATTERSON, G.; ZHANG, M. Fitness functions in genetic programming for classification with unbalanced data. In: SPRINGER. **Australasian Joint Conference on Artificial Intelligence**. [S.l.], 2007. p. 769–775.
- PEHAMBERGER, H.; STEINER, A.; WOLFF, K. In vivo epiluminescence microscopy of pigmented skin lesions. i. pattern analysis of pigmented skin lesions. **Journal of the American Academy of Dermatology**, Elsevier, v. 17, n. 4, p. 571–583, 1987.
- Pelle Digitale. **Dermatoscopia - Por que é importante examinar meus sinais ('pintas')?** 2016. Último acesso 10 de Setembro de 2016. Disponível em: <["http://pelledigitale.com.br/dermatoscopia.html"](http://pelledigitale.com.br/dermatoscopia.html)>.
- PENA-REYES, C. A.; SIPPER, M. Evolutionary computation in medicine: an overview. **Artificial Intelligence in Medicine**, Elsevier, v. 19, n. 1, p. 1–23, 2000.
- PHAM, D. L.; XU, C.; PRINCE, J. L. Current methods in medical image segmentation 1. **Annual review of biomedical engineering**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 2, n. 1, p. 315–337, 2000.
- POLI, R.; LANGDON, W. B. Genetic programming with one-point crossover. In: **Soft Computing in Engineering Design and Manufacturing**. [S.l.]: Springer, 1998. p. 180–189.
- PubMed Health Glossary. **Cerebral Cortex**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <["https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024757/"](https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0024757/)>.
- QUEIROZ, J. E. R. de. Representação de imagens. Notas de aula da disciplina Processamento Digital de Imagens - UFCG. 2015.
- REDA, I.; AYINDE, B. O.; ELMOGY, M.; SHALABY, A.; EL-MELEGY, M.; EL-GHAR, M. A.; EL-FETOUH, A. A.; GHAZAL, M.; EL-BAZ, A. A new cnn-based system for early diagnosis of prostate cancer. In: IEEE. **2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)**. [S.l.], 2018. p. 207–210.

- REZZE, G. G.; SÁ, B. C. S. de; NEVES, R. I. Dermatoscopia: o método de análise de padrões – dermoscopy: the pattern analysis. **Anais Brasileiros de Dermatologia**, SciELO Brasil, v. 81, p. 261–8, 2006. ISSN 0365-0596.
- RIGEL, D. S.; FRIEDMAN, R. J.; KOPF, A. W.; POLSKY, D. Abcde—an evolving concept in the early detection of melanoma. **Archives of dermatology**, American Medical Association, v. 141, n. 8, p. 1032–1034, 2005.
- RIGEL, D. S.; RUSSAK, J.; FRIEDMAN, R. The evolution of melanoma diagnosis: 25 years beyond the abcds. **CA: a cancer journal for clinicians**, Wiley Online Library, v. 60, n. 5, p. 301–316, 2010.
- ROBINSON, J. K.; TURRISI, R. Skills training to learn discrimination of abcde criteria by those at risk of developing melanoma. **Archives of dermatology**, American Medical Association, v. 142, n. 4, p. 447–452, 2006.
- RODRIGUES, E. L. M. **Evolução de Funções em Programação Genética Orientada a Gramática**. 116 p. Dissertação (Mestrado em Informática) — Universidade Federal do Paraná, Curitiba, 2002.
- RUDER, S. An overview of gradient descent optimization algorithms. **arXiv preprint arXiv:1609.04747**, 2016.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986.
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.
- RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Third. [S.l.]: Prentice Hall, 2009. (Series in Artificial Intelligence).
- RYAN, C.; COLLINS, J. J.; NEILL, M. O. Grammatical evolution: Evolving programs for an arbitrary language. In: SPRINGER. **European Conference on Genetic Programming**. [S.l.], 1998. p. 83–96.
- SÁEZ, A.; ACHA, B.; SERRANO, C. Pattern analysis in dermoscopic images. In: **Computer Vision Techniques for the Diagnosis of Skin Cancer**. [S.l.]: Springer, 2014. p. 23–48.
- SATO, K.; YOUNG, C.; PATTERSON, D. An in-depth look at google’s first tensor processing unit (tpu). **Google Cloud Big Data and Machine Learning Blog**, v. 12, 2017.
- SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 815–823.
- SETHI, R. **Programming languages: concepts and constructs**. Second. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1996.

SHIN, H.-C.; ROTH, H. R.; GAO, M.; LU, L.; XU, Z.; NOGUES, I.; YAO, J.; MOLLURA, D.; SUMMERS, R. M. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. **IEEE transactions on medical imaging**, IEEE, v. 35, n. 5, p. 1285–1298, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

ŠIMUNDIĆ, A.-M. Measures of diagnostic accuracy: basic definitions. **Ejifcc**, International Federation of Clinical Chemistry and Laboratory Medicine, v. 19, n. 4, p. 203, 2009.

Skin Cancer Foundation. **Melanoma - What is Melanoma?** 2019. Último acesso 04 de Março de 2019. Disponível em: <["https://www.skincancer.org/skin-cancer-information/melanoma"](https://www.skincancer.org/skin-cancer-information/melanoma)>.

SOYER, H.; ARGENZIANO, G.; CHIMENTI, S.; MENZIES, S.; PEHAMBERGER, H.; RABINOVITZ, H.; STOLZ, W.; KOPF, A. Dermoscopy of pigmented skins lesions. an atlas based on the consensus net meeting on dermoscopy 2000. EDRA Medical Publishing & New Media, 2001.

SOYER, P. H.; ARGENZIANO, G.; HOFMANN-WELLENHOF, R.; ZALAUDEK, I. **Guia Ilustrado de Dermatoscopia**. 2ª edição. ed. [S.l.]: Elsevier Brasil, 2012. ISBN 9788535265491.

SPANHOL, F. A.; OLIVEIRA, L. S.; PETITJEAN, C.; HEUTTE, L. Breast cancer histopathological image classification using convolutional neural networks. In: IEEE. **2016 international joint conference on neural networks (IJCNN)**. [S.l.], 2016. p. 2560–2567.

Stanford University. **CS231n: Convolutional Neural Networks for Visual Recognition**. 2018. Último acesso 17 de Fevereiro de 2018. Disponível em: <["http://cs231n.github.io/"](http://cs231n.github.io/)>.

SUK, H.-I.; SHEN, D. Deep learning-based feature representation for ad/mci classification. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [S.l.], 2013. p. 583–590.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.

TSAO, H.; OLAZAGASTI, J. M.; CORDORO, K. M.; BREWER, J. D.; TAYLOR, S. C.; BORDEAUX, J. S.; CHREN, M.-M.; SOBER, A. J.; TEGELER, C.; BHUSHAN, R. et al. Early detection of melanoma: reviewing the abcdes. **Journal of the American Academy of Dermatology**, Elsevier, v. 72, n. 4, p. 717–723, 2015.

VASCONCELOS, C. N.; VASCONCELOS, B. N. Increasing deep learning melanoma classification by classical and expert knowledge based image transforms. **CoRR**, abs/1702.07025, v. 1, 2017.

VESTERGAARD, M.; MACASKILL, P.; HOLT, P.; MENZIES, S. Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting. **British Journal of Dermatology**, Wiley Online Library, v. 159, n. 3, p. 669–676, 2008.

WADHAWAN, T.; SITU, N.; RUI, H.; LANCASTER, K.; YUAN, X.; ZOURIDAKIS, G. Implementation of the 7-point checklist for melanoma detection on smart handheld devices. In: IEEE. **2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. [S.l.], 2011. p. 3180–3183.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big Data**, SpringerOpen, v. 3, n. 1, p. 9, 2016.

WERBOS, P.; JOHN, P. J. P. Beyond regression: new tools for prediction and analysis in the behavioral sciences. 01 1974.

Wikimedia Commons. **File:Resolution illustration**. 2006. Último acesso 19 de Novembro de 2016. Disponível em: <["https://commons.wikimedia.org/wiki/File:Resolution_illustration.png"](https://commons.wikimedia.org/wiki/File:Resolution_illustration.png)>.

WINKLER, S. M.; AFFENZELLER, M.; JACAK, W.; STEKEL, H. Identification of cancer diagnosis estimation models using evolutionary algorithms: a case study for breast cancer, melanoma, and cancer in the respiratory system. In: ACM. **Proceedings of the 13th annual conference companion on Genetic and evolutionary computation**. [S.l.], 2011. p. 503–510.

WONG, M. L.; LEUNG, K. S. **Data mining using grammar based genetic programming and applications**. [S.l.]: Springer Science & Business Media, 2006. v. 3.

World Cancer Research Fund International. **Worldwide cancer data**. 2018. Último acesso 04 de Março de 2019. Disponível em: <["https://www.wcrf.org/dietandcancer/cancer-trends/worldwide-cancer-data"](https://www.wcrf.org/dietandcancer/cancer-trends/worldwide-cancer-data)>.

ZAWADZKI, J. **The Deep Learning(.ai) Dictionary**. 2018. Último acesso 17 de Março de 2019. Disponível em: <["https://towardsdatascience.com/the-deep-learning-ai-dictionary-ade421df39e4"](https://towardsdatascience.com/the-deep-learning-ai-dictionary-ade421df39e4)>.

ZHOU, J. T.; PAN, S. J.; TSANG, I. W.; YAN, Y. Hybrid heterogeneous transfer learning through deep learning. In: **Twenty-eighth AAAI conference on artificial intelligence**. [S.l.: s.n.], 2014.