

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**Redes Neurais Residuais Profundas para Classificação de
Imagens de Placas de Petri**

Guilherme dos Santos Marcon



São Carlos – SP

Redes Neurais Residuais Profundas para Classificação de Imagens de Placas de Petri

Guilherme dos Santos Marcon

Orientador: Prof. Dr. Ricardo Marcacini

Monografia final de conclusão de curso apresentada
ao Instituto de Ciências Matemáticas e de
Computação – ICMC-USP, como requisito parcial
para obtenção do título de Bacharel em Computação.
Área de Concentração: Aprendizado de Máquina

USP – São Carlos
Junho de 2020

Marcon, Guilherme dos Santos

Redes Neurais Residuais Profundas para Classificação
de Imagens de Placas de Petri / Guilherme dos
Santos Marcon. - São Carlos - SP, 2020.

46 p.; 29,7 cm.

Orientador: Ricardo Marcacini.

Monografia (Graduação) - Instituto de Ciências
Matemáticas e de Computação (ICMC/USP), São Carlos -
SP, 2020.

1. Aprendizado de Máquina. 2. Redes Neurais
Profundas Residuais. I. Marcacini, Ricardo. II.
Instituto de Ciências Matemáticas e de Computação
(ICMC/USP). III. Título.

Aos meus pais, Eliane e Nivaldo.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Ricardo Marcacini pela oportunidade e orientação tanto no desenvolvimento do projeto quanto na escrita deste documento.

Agradeço também a minha família e aos amigos pelo suporte.

RESUMO

MARCON, G. S.. **Redes Neurais Residuais Profundas para Classificação de Imagens de Placas de Petri.** 2020. 46 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

A cultura de micro-organismos é um método utilizado para identificar e descrever características de certa colônia de bactérias, que são cultivadas em um recipiente cilíndrico e achatado, geralmente feito de vidro ou plástico, chamada placa de Petri ou caixa Petri. Esses recipientes são então preenchidos com ágar, um caldo líquido cheio de nutrientes necessários para a bactéria em questão se desenvolver. As morfologias de colônias possuem diferenças em seu formato, tamanho, cor e textura, todos dependentes do tipo de bactéria e quantidade/composição do ágar inserido na placa. Assim, identificar um organismo com base nessas características é importante, porém os métodos são caros e demorados, dependentes de um especialista humano para a análise, o que torna o processo trabalhoso. Portanto, o estudo e desenvolvimento de um processo para automatizar a identificação do micro-organismo a partir de métodos computacionais para identificar e caracterizar espécies, seria uma grande inovação na área. O objetivo deste trabalho é investigar a classificação de imagens de Placas de Petri por meio de uma rede neural residual profunda, que são consideradas estado-da-arte para identificação de objetos em imagens. Uma rede neural profunda possui várias camadas, que conseguem aprender atributos em diversos níveis de abstração, de forma hierárquica. Enquanto métodos tradicionais de aprendizado de máquina exigem uma etapa de pré-processamento para extração de características das imagens, os diferentes níveis de abstração de redes neurais profundas substituem essa etapa — permitindo identificar características relevantes da imagem dentro do próprio processo de aprendizado. Por outro lado, adicionar muitas camadas na rede neural apresenta o “problema de degradação”, quando o modelo passa a ter um desempenho pior pela dificuldade inerente ao treinamento da rede. As redes neurais residuais existem para mitigar o problema de degradação. Elas mapeiam a rede de forma indireta por meio de “atalhos” entre blocos de camadas, de modo que algumas camadas podem ser “puladas” durante o treinamento da rede. Dentre os resultados obtidos com o desenvolvimento deste trabalho, vale destacar que foi gerado um modelo de classificação com acurácia média de 0.935. O mesmo conjunto de dados, quando avaliado em outros trabalhos usando modelos tradicionais de aprendizado de máquina, obtém acurácia menor, por exemplo, 0.73 com o método *Random Forests*, 0.49 com o método SVM e 0.67 com o método KNN. Dessa forma, há evidências de que a proposta investigada neste trabalho é competitiva.

Palavras-chave: Aprendizado de Máquina, Redes Neurais Profundas Residuais.

ABSTRACT

MARCON, G. S.. **Redes Neurais Residuais Profundas para Classificação de Imagens de Placas de Petri.** 2020. 46 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

The culture of microorganisms is a method used to identify and describe characteristics of a certain colony of bacteria, which are grown in a cylindrical and flattened container, usually made of glass or plastic, called a Petri dish. These containers are then filled with agar, a liquid broth filled with nutrients necessary for the bacteria in question to develop. Colony morphologies have differences in shape, size, color and texture, all dependent on the type of bacteria and the amount / composition of the agar inserted in the plate. Thus, identifying an organism based on these characteristics is important, but the methods are expensive and time-consuming, dependent on a human specialist for analysis, which makes the process laborious. Therefore, the study and development of a process to automate the identification of the microorganism by using computational methods to identify and characterize species would be a great innovation in the area. The objective of this work is to investigate the classification of Petri dishes images by means of a deep residual neural network, which are considered state-of-the-art to identify objects in images. A deep neural network has several layers, which can learn attributes at different levels of abstraction, in a hierarchical way. While traditional machine learning methods require a pre-processing step to extract characteristics from the images, the different levels of abstraction of deep neural networks replace this step — allowing the identification of relevant image features within the learning process itself. On the other hand, adding many layers in the neural network presents the “degradation problem”, i.e., when the model starts to perform worse due to the difficulty inherent in the training of the network. Residual neural (ResNet) networks mitigate this degradation problem. ResNet map the network indirectly through “shortcuts” between layer blocks, so that some layers can be ‘‘skipped’’ during training step of the network. Among the results obtained with the development of this work, We trained a deep residual network that achieved an average accuracy of 0.935. When evaluated in other studies using traditional machine learning models in this same dataset, classification models obtain less accuracy, for example, 0.73 with the Random Forests method, 0.49 with the SVM method and 0.67 with the KNN method. Thus, there is evidence that the proposal investigated in this work is competitive.

Key-words: Machine Learning, Deep Residual Neural Networks.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de uma cultura em placa de Petri.	18
Figura 2 – Esquema geral de um Perceptron. Fonte: Autor.	21
Figura 3 – Dois tipos de arquiteturas de redes neurais. Fonte: Autor.	22
Figura 4 – Exemplos de funções de ativação. Fonte: Autor.	23
Figura 5 – Exemplo de convolução	25
Figura 6 – Exemplo de pooling	25
Figura 7 – Exemplo de uma Rede Neural Convolucional	26
Figura 8 – Um bloco de aprendizado residual. Fonte: (HE <i>et al.</i> , 2016)	27
Figura 9 – Comparação entre 3 modelos de redes neurais profundas: VGG-19, composta de 19 camadas; uma rede não-residual com 34 camadas; uma rede residual com 34 camadas. As camadas "fc" são as "fully connected", mesmo conceito das MLPs, e as setas pontilhadas na rede residual significa que há diferença nas dimensões dos inputs, portanto existe um pooling de um bloco para outro. Fonte: (HE <i>et al.</i> , 2016)	28
Figura 10 – Exemplos das morfologias das colônias. Fonte: (AHERN, 2015)	32
Figura 11 – Imagens de exemplo das diferentes bactérias do conjunto. Fonte: Autor.	33
Figura 12 – Bactéria bacillus-aryabhattai-01-columbia	34
Figura 13 – Bactéria bacillus-aryabhattai-02-columbia	34
Figura 14 – Bactéria bacillus-bingmayongensis-columbia	34
Figura 15 – Bactéria bacillus-megaterium-columbia	34
Figura 16 – Bactéria bacillus-mycooides-columbia	35
Figura 17 – Bactéria bacillus-paranthracis-01-columbia	35
Figura 18 – Bactéria bacillus-paranthracis-02-columbia	35
Figura 19 – Bactéria bacillus-wiedmannii-01-columbia	35
Figura 20 – Bactéria bacillus-wiedmannii-02-columbia	36
Figura 21 – Bactéria lysinibacillus-boronitolerans-columbia	36
Figura 22 – Resumo dos métodos utilizados. Fonte: Autor.	36
Figura 23 – Erro (Loss) do treino (eixo y) (em função da época (eixo x)).	39
Figura 24 – Funcionamento do <i>Leave-One-Out</i> . A parte cinza indica um exemplo de teste e a parte branca os exemplos de treino. Fonte: Autor.	39
Figura 25 – Exemplo da utilização do LIME na classe bacillus-wiedmannii-02-columbia, a imagem esquerda possui um exemplo onde a classificação foi correta, já a direita possui um classificado errado.	42

LISTA DE TABELAS

Tabela 1 – Parâmetros utilizados no <i>Data Augmentation</i> .	37
Tabela 2 – Resultado do treino, por classe. Métricas provenientes da Matriz de Confusão: accuracy, f1 score, precision e recall.	40

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Motivação e Contextualização	17
1.2	Objetivos	19
1.3	Principais Contribuições	19
1.4	Organização do Trabalho	20
2	MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS	21
2.1	Redes Neurais Profundas	21
2.2	Redes Neurais Convolucionais	24
2.3	Redes Neurais Residuais	26
2.4	Considerações Finais	27
3	DESENVOLVIMENTO	31
3.1	Classificação de Bactérias a partir de Imagens de Placa de Petri	31
3.2	Conjunto de Dados	31
3.3	Método	32
3.3.1	<i>Leitura de imagens e preparação dos dados</i>	37
3.3.2	<i>Modelo e treinamento da Rede Residual ResNet50</i>	37
3.3.3	<i>Avaliação com o Leave-One-Out</i>	38
3.4	Resultados	39
3.4.1	<i>Desempenho da Classificação</i>	39
3.4.2	<i>Interpretando a Rede Neural</i>	41
4	CONCLUSÃO	43
4.1	Considerações finais	43
4.2	Considerações sobre o curso de graduação	43
4.3	Sugestões para o curso de graduação	44
4.4	Trabalhos futuros	44
	REFERÊNCIAS	45

Capítulo 1

INTRODUÇÃO

1.1 Motivação e Contextualização

O cultivo de bactérias em placas de Petri é um dos processos fundamentais da microbiologia. Este processo consiste em colocar amostras de algum material previamente coletado em ágar (um caldo de nutrientes específico para o micrório) e esperar o micro-organismo se reproduzir, que pode demorar de horas até dias. Observando a morfologia e aparência da colônia formada, um profissional pode inferir características para identificar bactérias existentes no material e suas propriedades (FERRARI; SIGNORONI, 2014).

Em um dia no laboratório, um clínico trabalha com um grande número de placas, realizando a identificação visual delas, descartando as que não são mais necessárias e dando uma atenção maior para as placas mais críticas ou que precisam de mais esforço para a identificação. Essa grande carga de trabalho, que aumenta a probabilidade de erros, é o que incentiva buscar formas de automatizar o processo (BOURBEAU; LEDEBOER, 2013; NAUGLER; CHURCH, 2019). A automatização se dá em dois pontos: padronizar as fases pré-analíticas (preparação da amostra, processamento e incubação), aumentando a confiabilidade e reproduzibilidade do processo; e digitalizar as culturas, onde a identificação passa a ser por imagens em alta resolução, como apresentado na Figura 1.

Até recentemente, o processo de digitalizar e processar imagens não era eficiente, pois o preço das câmeras era alto, a qualidade das imagens deixava a desejar e o processamento dessas imagens para classificação eram consideravelmente demorados, tornando ineficaz o custo-benefício de se utilizar este método. No entanto, atualmente existem aparelhos de captura de imagens baixo de custo e o advento de GPUs aumentou de forma significativa a capacidade de processamento de imagens (AFFONSO *et al.*, 2017). Assim, métodos para a classificação de imagens usando inteligência artificial, como aprendizado de máquina, têm se tornado mais populares também para apoiar automatização de processos na área de microbiologia (AFFONSO *et al.*, 2017; NAUGLER; CHURCH, 2019).

Dentre diferentes estratégias para classificação de imagens, as técnicas de Aprendizado Profundo com Redes Neurais são o estado-da-arte na classificação de imagens (AFFONSO *et al.*, 2017; LIU *et al.*, 2020). As Redes Neurais Convolucionais, por exemplo, possuem várias camadas para a criação de representações intermediárias da imagem original, de maneira a facilitar a construção de conceitos complexos pela rede (RAWAT; WANG, 2017). Em contrapartida, por



Figura 1 – Exemplo de uma cultura em placa de Petri.

causa de sua estrutura complexa, possuem um grande custo computacional para o treino e precisam de uma grande quantidade de dados para gerar um modelo com boa performance ([LIU et al., 2020](#)).

Uma rede neural profunda possui várias camadas, onde as primeiras podem ser responsáveis por detectar bordas e as próximas identificar formas e objetos, dependendo do problema ([LITJENS et al., 2017](#)). Dessa forma, esses tipos de redes conseguem aprender atributos em diversos níveis de abstração, de forma hierárquica. Enquanto métodos tradicionais de aprendizado de máquina exigem uma etapa de pré-processamento para extração de características das imagens, os diferentes níveis de abstração de redes neurais profundas substituem essa etapa — permitindo identificar características relevantes da imagem dentro do próprio processo de aprendizado ([LIU et al., 2020](#)). Entretanto, adicionar muitas camadas na rede neural apresenta o “problema de degradação”, quando o modelo passa a ter um desempenho pior pela dificuldade inerente ao treinamento da rede ([RAWAT; WANG, 2017](#)).

As redes neurais residuais existem para mitigar o problema de degradação ([HE et al., 2016](#)). Elas mapeiam a rede de forma indireta por meio de “atalhos” entre blocos de camadas, de modo que algumas camadas podem ser “puladas” durante o treinamento da rede. Em geral, a própria estratégia de gerar atalhos entre as camadas (e sua matriz de pesos desses atalhos) faz parte do processo de treinamento das redes residuais profundas ([HE et al., 2016](#)).

Redes residuais profundas também são úteis para lidar com um outro desafio na área, que é a quantidade de dados necessários para o treinamento de um modelo. Nesse caso, diversos estudos disponibilizam redes pré-treinadas em um grande conjunto de imagens de diferentes categorias. Utilizando um modelo já treinado, no qual foi usado um grande conjunto de dados, é esperado que o modelo já consiga identificar padrões também relevantes para o problema

em questão, em especial, conceitos como detecção de bordas, texturas e formatos (HE *et al.*, 2019). Por exemplo, neste trabalho é utilizado a ResNet50, uma rede neural residual profunda de 50 camadas, pré-treinada com a base de dados ImageNet, da competição *Large Scale Visual Recognition Challenge*¹, contendo 1.281.167 imagens de treinamento em 1000 categorias. Desta maneira, é possível aproveitar camadas intermediárias da ResNet50 e treinar um modelo robusto para um novo conjunto de dados, amenizando problemas relacionadas à falta de um grande conjunto de treinamento que costuma ser exigido para classificação de imagens com redes neurais profundas.

1.2 Objetivos

O objetivo geral deste Trabalho de Conclusão de Curso é avaliar o desempenho de redes neurais residuais profundas para a tarefa de classificação de bactérias, em que as bactérias são representadas por imagens de colônias cultivadas em Placas de Petri. Para atingir o objetivo geral, são apresentados os seguintes objetivos específicos:

- Investigar e desenvolver um módulo computacional para classificação de imagens de Placas de Petri por meio de redes neurais residuais profundas. O módulo deve ter capacidade para utilizar modelos pré-treinados e realizar o treinamento da rede neural usando GPU; e
- Avaliar o desempenho de modelo pré-treinado, denominado ResNet50, como forma de transferência de conhecimento. Nesse caso, a ideia é explorar a habilidade de identificar estruturas como bordas, texturas, formas e objetos aprendida em outros domínios. Novas camadas são adicionadas à ResNet50 para considerar o problema de classificação de bactérias e, então, uma nova etapa de treinamento é realizada a partir de imagens de placas de Petri.

1.3 Principais Contribuições

As principais contribuições resultantes desse trabalho são apresentadas a seguir:

- Foi desenvolvido um módulo computacional para treinamento de classificadores de imagem de placas de Petri. As imagens representam colônias de bactérias, nas quais algumas imagens são anotadas por especialistas e o objetivo é identificar a bactéria em novas imagens não anotadas. O módulo utiliza como base a ResNet50, uma rede residual profunda pré-treinada em outros domínios, e transfere padrões dessa rede para a classificação de bactérias.

¹ <<http://www.image-net.org/challenges/LSVRC/>>

- Foi gerado um modelo de classificação de bactérias a partir de imagens de Placas de Petri de suas respectivas colônias cultivadas. O modelo proposto obteve acurácia média de 0.935. O mesmo conjunto de dados, quando avaliado em outros trabalhos usando modelos tradicionais de aprendizado de máquina, obtém acurácia menor, por exemplo, 0.73 com o método Random Forests, 0.49 com o método SVM e 0.67 com o método KNN. Dessa forma, há evidências de que a proposta investigada neste trabalho é competitiva.
- Para mitigar uma das principais críticas aos modelos de aprendizado profundo, que é a explicabilidade do modelo, o módulo computacional desenvolvido também incorpora o método chamado LIME (*Local Interpretable Model-agnostic Explanations*), que basicamente faz um realce na imagem da entrada indicando quais partes da imagem da colônia foram mais consideradas pelo modelo para determinar a classe (tipo de bactéria). Dessa forma, especialistas podem validar o modelo e identificar possíveis melhorias para o treinamento.

1.4 Organização do Trabalho

O trabalho está organizado da seguinte maneira:

- No Capítulo 1 foi apresentada uma motivação e contextualização sobre classificação de imagens de Placas de Petri, apresentando as vantagens obtidas em automatizar o processo de identificação de micro-organismos. Também foi discutido que redes neurais profundas, em especial, redes residuais são reconhecidas entre os métodos estado-da-arte na área.
- No Capítulo 2 é apresentada uma fundamentação do trabalho. É apresentada uma breve revisão de Redes Neurais Profundas, Redes Neurais Convolucionais e Redes Neurais Residuais; que são os tópicos de investigação deste trabalho.
- No Capítulo 3 é descrito o método desenvolvido durante o trabalho, em particular, detalhes de como uma rede neural residual profunda, ResNet50, pré-treinada em outro domínio, foi utilizada para apoiar o problema de classificação de bactérias. Foram apresentados os critérios de avaliação e uma discussão dos resultados.
- Finalmente, no Capítulo 4 são apresentadas as considerações finais deste trabalho, bem como direções para trabalhos futuros.

Capítulo 2

MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS

2.1 Redes Neurais Profundas

Redes neurais artificiais foram propostas com a motivação de simular o comportamento de um cérebro humano. Desta forma, uma rede neural consiste em um conjunto de neurônios (também chamados de nós), onde cada um está conectado à outros nós e cada uma dessas ligações possui um peso associado, demonstrando o quanto forte dois neurônios artificiais estão conectadas. Além disso, cada nó possui uma característica chamada função de ativação, que é uma função matemática que determina quando (ou quanto) um neurônio transmite seu sinal para os outros (HAYKIN, 1994).

A unidade mais simples para demonstrar esse funcionamento é o Perceptron (Figura 2). Ele é um modelo utilizado para classificar dois conjuntos linearmente separáveis, que são aqueles que podem ser separados por um hiperplano. Os inputs x_i são multiplicados pelos seus devidos pesos w_i e somados em conjunto com um valor bias multiplicado pelo seu peso w_b . Depois dessa soma passa pela função de ativação. No exemplo ilustrado na Figura 2 é usada a função de ativação ReLU, em que valores positivos são mantidos iguais, enquanto os negativos são zerados; que é uma função de ativação muito usada em redes neurais profundas (GOODFELLOW; BENGIO; COURVILLE, 2016).

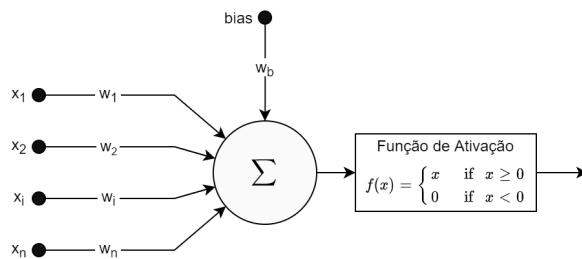


Figura 2 – Esquema geral de um Perceptron. Fonte: Autor.

Existem diferentes arquiteturas de redes neurais. Dentre as principais, existem aquelas de uma única camada e as de múltiplas. As Figuras 3a e 3b ilustram os modelos conceituais dessas redes, respectivamente. Note que as diferenças entre elas são a quantidade de camadas ocultas (nome dado às camadas entre a de entrada e saída). Enquanto as redes de uma única camada são

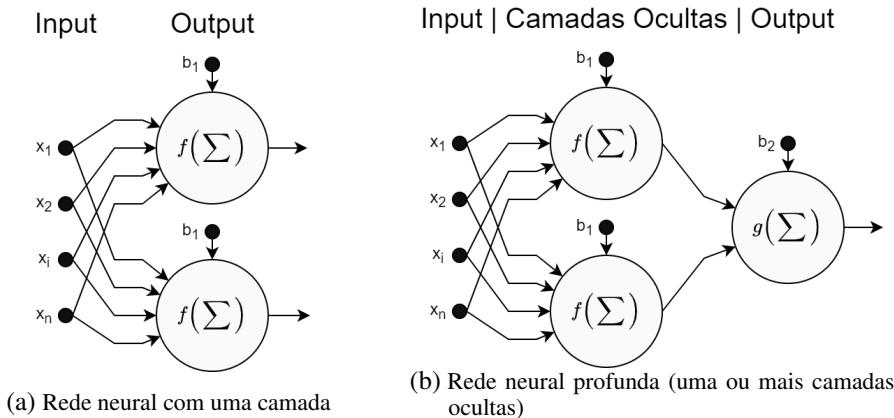


Figura 3 – Dois tipos de arquiteturas de redes neurais. Fonte: Autor.

restritas ao tamanho da saída, as rede de múltiplas camadas podem possuir qualquer número de nós e 1 ou mais camadas ocultas. Neste projeto, o interesse é explorar as redes neurais de múltiplas camadas, também conhecidas como MLP (*Multi-layer Perceptrons*) (HAYKIN, 1994).

Uma rede MLP é composta por várias camadas de perceptrons, possuindo uma ou mais camadas ocultas. Para cada nó de uma camada, com exceção da de entrada, um neurônio (perceptron) pode ser definido pela Equação 2.1:

$$X_j^l = f\left(\sum_i X_i^{l-1} W_{ij} + b^l\right) \quad (2.1)$$

onde X_j^l é o j-ésimo nó da camada l ; $f()$ é uma função de ativação (exemplos mais abaixo); X_i^{l-1} é o i-ésimo nó da camada $l-1$; W_{ij} é o peso que liga o X_i^{l-1} com o X_j^l e b^l é um valor denominado bias que é o mesmo para cada nó da camada l ;

O aprendizado de uma rede neural se deve ao ajustar os parâmetros do modelo para um melhor mapeamento entrada-saída. Esses parâmetros, no exemplo de uma MLP, são os pesos que multiplicam as entradas e o bias, modificados através de um processo de treinamento da rede neural chamado *backpropagation* (HAYKIN, 1994). Outros parâmetros a serem explorados são a quantidade de camadas, o número de nós em cada uma e como eles estão conectados, ou seja, a arquitetura da rede neural também possui impacto na sua performance.

O *backpropagation* é um processo que altera os pesos sinápticos da rede de maneira a minimizar a diferença entre a saída esperada e atual. Ele é fundamentado no Método de Descida do Gradiente e consiste em calcular a diferença entre as saídas, através da chamada função de custo, para depois atualizar os pesos anteriores baseado na Equação 2.2. As Equações 2.3, 2.4 e 2.5 apresentam detalhes dessa atualização sináptica (HAYKIN, 1994).

$$w^{[c+1]} = w^{[c]} + (1 - \alpha)\Delta w^{[c]} + \alpha\Delta w^{[c-1]} \quad (2.2)$$

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (2.3)$$

$$E = \frac{1}{2} \sum_{k=1}^K [e_k]^2 \quad (2.4)$$

$$e_k = d_k - z_k \quad (2.5)$$

onde w é o valor de um peso sináptico, c é o ciclo de treinamento (ou época), η é o parâmetro de momento, α é a taxa de aprendizado, E é a função de custo (ou função de erro), d_k é a saída esperada para o neurônio k da última camada do modelo e z_k é a saída atual daquele nó. Os parâmetros momento e aprendizado definem o comportamento do aprendizado da rede e devem ser ajustados conforme cada base de dados.

Sem confundir com a função de custo, as funções de ativação são aquelas que no fim das contas determinam o valor de saída de um neurônio, representando o quão ativo ele está. As utilizadas diretamente neste projeto foram a ReLU e a Sigmoid (fórmulas pela Figura 4). A ReLU normalmente é utilizada entre camadas ocultas, simulando a ativação de um neurônio: quanto mais forte o valor, mais forte a ativação, e sem ativação negativa; a Sigmoid é utilizada na camada de saída e em problemas de classificação, indicando a probabilidade da entrada ser da classe correspondente àquele nó (GOODFELLOW; BENGIO; COURVILLE, 2016).

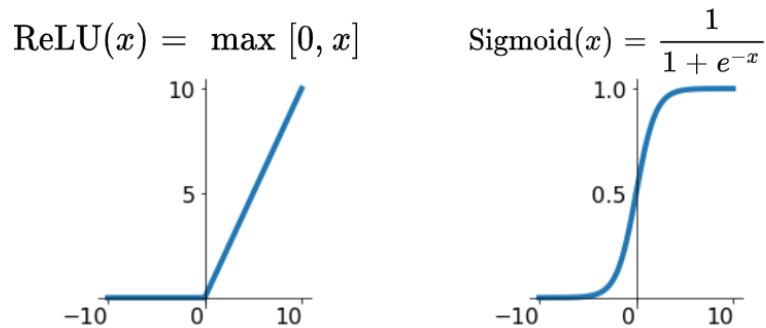


Figura 4 – Exemplos de funções de ativação. Fonte: Autor.

Conforme comentado anteriormente, o aprendizado da rede neural consiste em ajustar os pesos entre neurônios. Para isso, o algoritmo *backpropagation* é o mais utilizado. De forma geral, primeiro se propaga um valor de entrada até que sua saída seja calculada, depois é aplicado a função de custo, que representa o quão próximo o resultado esteve do valor esperado. Algumas funções comuns são o erro quadrático médio (MSE) para problemas de regressão e a entropia cruzada (cross-entropy) para classificação (GOODFELLOW; BENGIO; COURVILLE, 2016). Com o resultado da função de custo, é calculado, a partir das derivadas dos cálculos dos nós e da função de ativação, o quanto tal peso ou bias influenciou o resultado, ajustando o valor dele de maneira adequada para aproximar o resultado da predição ao valor esperado.

Um avanço recente nesta área são as Redes Neurais Profundas ([ZHANG et al., 2018](#)). Tais redes exploram muitas camadas ocultas para aprender representações intermediárias do problema. Para o contexto de imagens, as camadas das redes neurais profundas podem representar diferentes estruturas da imagem, como detecção de bordas e objetos; o que torna tais redes muito úteis para problemas em que a representação dos dados é complexa ([GOODFELLOW; BENGIO; COURVILLE, 2016](#); [LIU et al., 2020](#)).

Em comparação com as redes MLP, uma rede neural profunda pode ser formulada pela Equação 2.6, em que a saída de uma rede pode ser representada por composições de funções ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

$$f(\mathbf{x}) = f_L(\dots f_2(f_1(\mathbf{x}))) \dots \quad (2.6)$$

onde x é a entrada e $f_n(\cdot)$ é a n -ésima camada, que recebe os valores das camadas anteriores.

Essa propriedade é claramente visível nas redes neurais convolucionais (discutidas na Seção 2.2), onde existem camadas e camadas de filtros transformando as imagens de entrada.

Entretanto, um problema de se utilizar muitas camadas é a degradação, onde quantidade de camadas atrapalha o treino com o *backpropagation*. Para também minimizar esse aspecto, é utilizado as redes residuais (discutidas na Seção 2.3), que são conexões de "atalho" entre diferentes camadas.

Por esses motivos, as redes neurais convolucionais residuais foram escolhidas como foco principal para esse trabalho. Nas próximas seções, são apresentados mais detalhes sobre esse modelo de Rede Neural Profunda.

2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNN) se tornaram populares com imagens e problemas de visão computacional ([LIU et al., 2017](#)). A ideia desse tipo de rede é aplicar vários filtros diferentes, criando várias imagens transformadas para serem analisadas por uma Rede Neural Profunda. As CNNs são caracterizadas por:

- Convolução (Figura 5): dada uma matriz K com tamanho nxn (também chamado de filtro) e uma matriz I (imagem de entrada, por exemplo), o produto dessa operação leva cada região nxn da matriz I a ser aplicada o filtro K (soma da multiplicação ponto a ponto). Fazer isso para cada região gera-se uma outra matriz, de tamanho normalmente menor que a matriz I . Quando não se quer que a matriz resultante tenha tamanho reduzido, a matriz I é inflada e completada com zeros ou o valor vizinho. No ponto de vista de uma rede neural, o filtro é uma matriz de pesos.

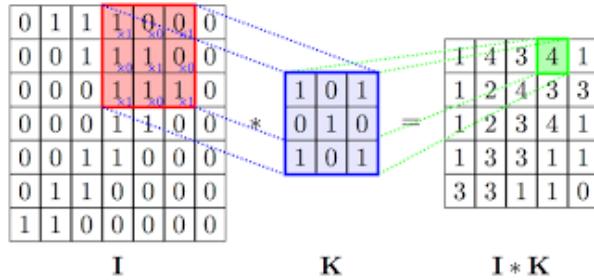


Figura 5 – Exemplo de convolução. Composto por um filtro 3x3. Fonte: ([PRAKASH, 2018](#))

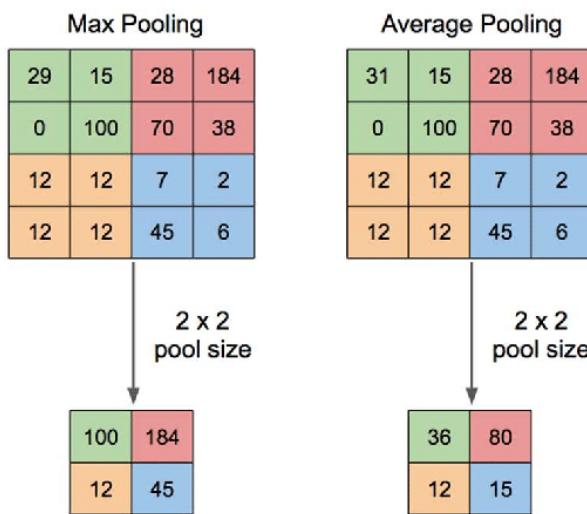


Figura 6 – Exemplo de pooling. Este é composto por: tamanho do filtro 2x2, tipo de filtro (max e média) e valor de passo (2). Fonte: ([YANI; IRAWAN; S.T., 2019](#))

- Pooling (Figura 6): no geral, *pooling* é usado após uma camada de convolução. A função dele é diminuir o tamanho de uma matriz e é composto por um filtro $n \times n$, um valor de passo (*stride*) e o tipo de filtro. O filtro $n \times n$ é parecido com a convolução, na questão de ser aplicada em regiões $n \times n$ da matriz original. Porém, uma diferença é o *stride*, em que ao invés de aplicar em todas as regiões $n \times n$, o *stride* "pula" uma quantidade de regiões vizinhas. Outra diferença é o tipo de filtro, ao invés da convolução, o *pooling* resulta em separar o valor máximo da região, ou a média, ou o mínimo, etc.

Um exemplo para ilustrar o funcionamento do CNN é a classificação de números manuscritos (Figura 7), conforme discutido em ([PONTI; COSTA, 2017](#)). A imagem de entrada (*input*) é, neste caso, o número 2 escrito à mão e possui tamanho (28x28x1). A primeira camada de convolução é composta por n_1 filtros 5x5, gerando uma camada de tamanho (24x24xn1). Essa redução de tamanho (28 -> 24) se deve aos filtros 5x5. Após essa convolução, existe uma camada de *pooling*, com filtros 2 x 2, tipo de filtro como máximo e *stride* não explicitado como 2, onde o tamanho é reduzido pela metade (12x12xn1). Em seguida, existe outra camada de convolução, também com filtros 5x5, mas com n_2 desses filtros, gerando a camada com tamanho (8x8xn2). Depois, mais uma camada de *pooling*, com a mesma configuração, diminuindo o

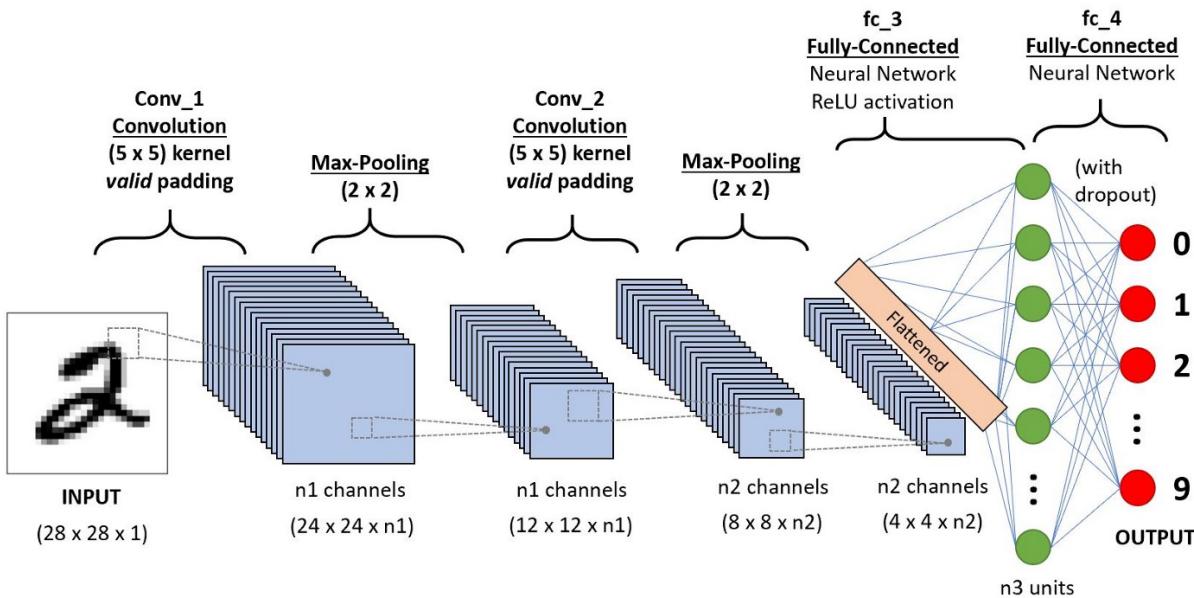


Figura 7 – Um exemplo de Rede Neural Convolucional para classificação de números manuscritos. A primeira parte é composta de filtros convolucionais seguidos por processos de *pooling*. A segunda parte é uma rede neural profunda, no caso da imagem, os filtros achados são o *input*, a rede possui uma camada oculta e um vetor de saída com 10 espaços, um para cada dígito. Fonte: ([MARCHILDON, 2017](#))

tamanho para $(4 \times 4 \times n_2)$, que é achatada para um vetor de uma dimensão e aplicado a função de ativação ReLU. Esse estágio foi apenas a parte convolucional da rede. Em seguida a saída do estágio convolucional é apresentada à MLP, que é composta com uma camada oculta de tamanho n_3 , totalmente conectada com a entrada (última camada achatada da parte convolucional) e com a saída, onde existe 1 neurônio correspondente a cada dígito.

2.3 Redes Neurais Residuais

Um dos problemas das redes neurais profundas explicadas anteriormente é a dificuldade em treinar modelos com muitos parâmetros, o que inclui redes profundas com muitas camadas ([ZHANG et al., 2018](#)). Com o intuito de resolver esse problema e aumentar consideravelmente a profundidade dos modelos, mas sem perder acurácia, as redes neurais residuais (ResNet) foram criadas ([HE et al., 2016](#)). Essas redes incorporam uma estratégia para evitar a degradação do treinamento, o desaparecimento do gradiente no algoritmo *backpropagation*, devido à quantidade de camadas.

Um bloco de aprendizado residual (Figura 8) consiste em fazer um "atalho" entre o começo do bloco e o fim dele, utilizando uma matriz identidade. Considerando a entrada de um bloco como x , a saída desse bloco numa rede não-residual seria $F(x)$. Agora, considerando uma rede neural residual, a saída é $F(x) + x$ (assumindo que possuem o mesmo tamanho). Em outras palavras, essa estrutura permite que as camadas mais profundas recebam diretamente as informações das camadas superficiais, sem a interferência das camadas intermediárias ([HE et al.,](#)

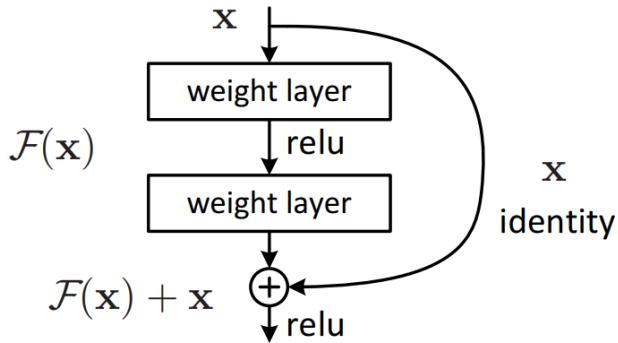


Figura 8 – Um bloco de aprendizado residual. Fonte: ([HE et al., 2016](#))

[2016](#)).

Dessa maneira, essa técnica garante que uma rede com N blocos tenha o mesmo espaço de resolução que uma rede de tamanho menor, permitindo aumentar consideravelmente a profundidade de uma rede em busca do melhor resultado.

A ResNet em si é um modelo de CNN residual. Enquanto os estados-da-arte anteriores eram redes cada vez mais profundas e cada vez mais propensas ao problema de degradação, a ResNet veio para resolver esse obstáculo e acabou se tornando a estratégia com maior número de camadas. Para efeito de ilustração, a arquitetura de uma ResNet com 34 camadas é mostrada na Figura 9.

Neste projeto, foi utilizado a ideia de aprendizado por transferência, que consiste em utilizar um modelo com pesos já treinados, retirar a última camada de classificação e adicionar outra referente às classes específicas da nova base de dados, retrainando o modelo com os novos dados ([HE et al., 2019](#)). Para isso foi escolhido a ResNet50, um modelo com 50 camadas para a classificação de imagens da ImageNet (um grande repositório de imagens, divididos em 1000 categorias, como teclado, controle remoto, lápis e muitos tipos de animais). Utilizando-a com o aprendizado por transferência, espera-se que esse modelo possua informações importantes para a classificação de outros tipos de imagens.

2.4 Considerações Finais

Redes neurais profundas, em especial, redes residuais têm se tornado populares para classificação de imagens. Essa popularidade se dá pela habilidade em lidar com redes com muitas camadas, mas ao mesmo tempo evitando o problema de degradação do treinamento. Assim, tais redes permitem gerar representações abstratas e hierárquicas, com muitos níveis, das imagens de entrada, que são úteis para classificação em problemas complexos que exigem reconhecimento de bordas, texturas e formatos.

A classificação de imagens em placas de Petri é considerado um problema complexo, pois identificar as colônias envolvem exatamente tais características: bordas, texturas e formatos.

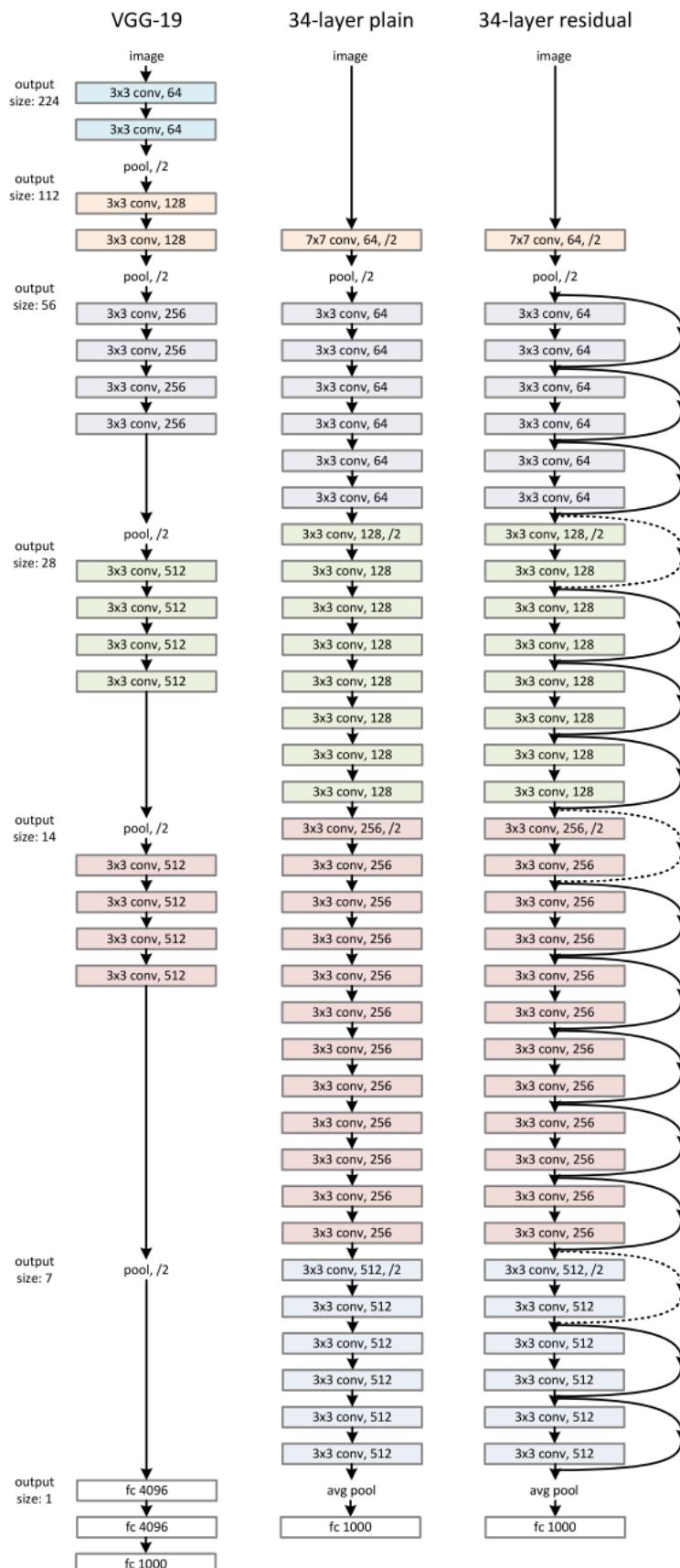


Figura 9 – Comparação entre 3 modelos de redes neurais profundas: VGG-19, composta de 19 camadas; uma rede não-residual com 34 camadas; uma rede residual com 34 camadas. As camadas "fc" são as "fully connected", mesmo conceito das MLPs, e as setas pontilhadas na rede residual significam que há diferença nas dimensões dos inputs, portanto existe um pooling de um bloco para outro. Fonte: ([HE et al., 2016](#))

Por outro lado, não é trivial obter uma base grande imagens de placas Petri, digitalizadas e pré-anotadas por especialistas. Assim, modelos pré-treinados, como a ResNet50 parecem ser uma caminho promissor para apoiar essa tarefa.

Nesse contexto, no próximo capítulo é apresentado o método desenvolvido neste projeto, que explora redes residuais profundas e redes pré-treinadas para a classificação de bactérias a partir de imagens de suas respectivas colônias cultivadas em placas de Petri.

Capítulo 3

DESENVOLVIMENTO

3.1 Classificação de Bactérias a partir de Imagens de Placa de Petri

A cultura de micro-organismos é um método utilizado para identificar e descrever características de certa colônia de bactérias, que são cultivadas em um recipiente cilíndrico e achatado, geralmente feito de vidro ou plástico, chamada placa de Petri ou caixa Petri. Esses recipientes são então preenchidos com ágar, um caldo líquido cheio de nutrientes necessários para a bactéria em questão se desenvolver (FERRARI; SIGNORONI, 2014).

Na Figura 10 são ilustrados alguns exemplos de morfologias de colônias, que podem possuir diferenças enormes no seu formato, tamanho, cor e textura, todos dependentes do tipo de bactéria e quantidade/composição do ágar inserido na placa. Assim, identificar um organismo com base nessas características é importante, porém os métodos são extremamente trabalhosos, caros e demorados, dependentes de um especialista humano para a análise, o que torna o processo trabalhoso, além de que na maioria dos casos, essa identificação pode destruir a cultura. Portanto o estudo e desenvolvimento de um processo para automatizar a identificação do micro-organismo a partir de visão computacional, para identificar e caracterizar espécies, seria uma grande inovação na área da saúde (BOURBEAU; LEDEBOER, 2013; NAUGLER; CHURCH, 2019).

Com isso em mente, o objetivo deste projeto é investigar a classificação de imagens por meio de uma rede neural residual profunda, que são consideradas estado-da-arte para problemas similares, como identificação de objetos em imagens (LIU *et al.*, 2020). Ainda, nesse trabalho foi explorada a ideia de utilizar modelos neurais pré-treinados com o intuito de transferir o conhecimento desses domínios para o problema de classificação de imagens de placas de Petri. É esperado que esse modelo seja capaz de identificar bordas, formas, cor e outras características que também são importantes para distinguir colônias de bactérias.

3.2 Conjunto de Dados

O conjunto de dados é composto por imagens de colônias de diferentes bactérias disponibilizadas pelo Projeto Capstone¹, de pesquisadores da *Foundation School of Engineering and*

¹ <<https://github.com/deibyrios/bacteria-classification>>

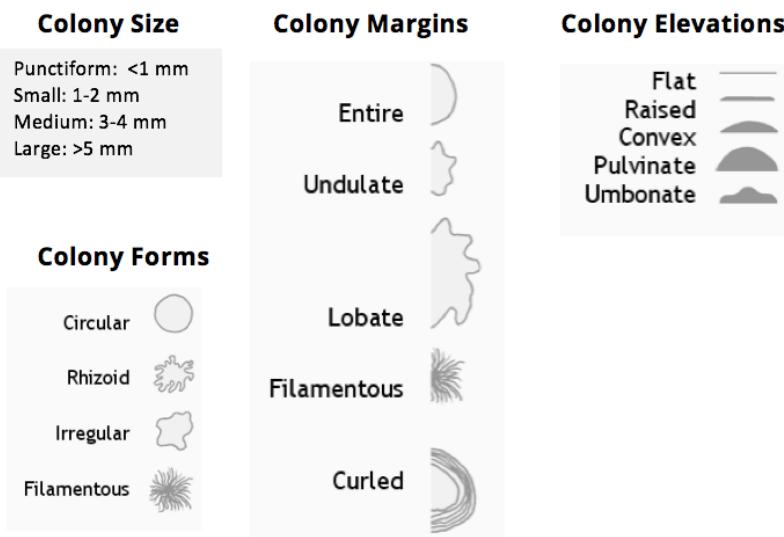


Figura 10 – Exemplos das morfologias das colônias. Fonte: ([AHERN, 2015](#))

Applied Science, Columbia University, USA. Devido à diferenças no solo, essas bactérias podem não parecer da mesma espécie, mas elas possuem características em comum que podem ser identificadas, o que torna esse problema desafiador. A base de dados utilizada foi pré-classificada por biólogos especialistas nesses organismos e, assim, é possível verificar o desempenho de forma objetiva usando métricas de precisão, revocação e acurácia.

A Figura 11 apresenta um exemplo de cada tipo de bactéria existente no conjunto de dados, em que cada classe possui 20 imagens e cada imagem possui uma versão original e pré-processada. Na versão pré-processada, a bactéria em si é isolada da imagem, retirando a placa de Petri e outras informações indesejáveis para o aprendizado de máquina. Neste trabalho, as imagens utilizadas para treino e teste dos modelos são apenas as pré-processadas.

Nas Figuras 12-21 é apresentado um exemplo de cada tipo de bactéria. É possível observar que elas são bem diferentes entre si, cada tipo possui imagens de placas usando diferentes tipos de cultivo. Existem aquelas com um único ponto inicial (Figuras 12, 14, 17 e 20), aquelas com vários pontos (Figuras 13, 16 e 19) e aquelas espalhadas (Figuras 15, 18 e 21). Além disso, também existem diferenças no substrato de cada placa, onde a Figuras 12, 13 e 15 são menos desenvolvidas que as Figuras 17, 18 e 20, por exemplo.

3.3 Método

Os passos utilizados neste projeto são: preparação dos dados, construção da rede neural, realização do treino e a avaliação. Esses passos estão explicitados na Figura 22 e serão explicados nas subseções a seguir.

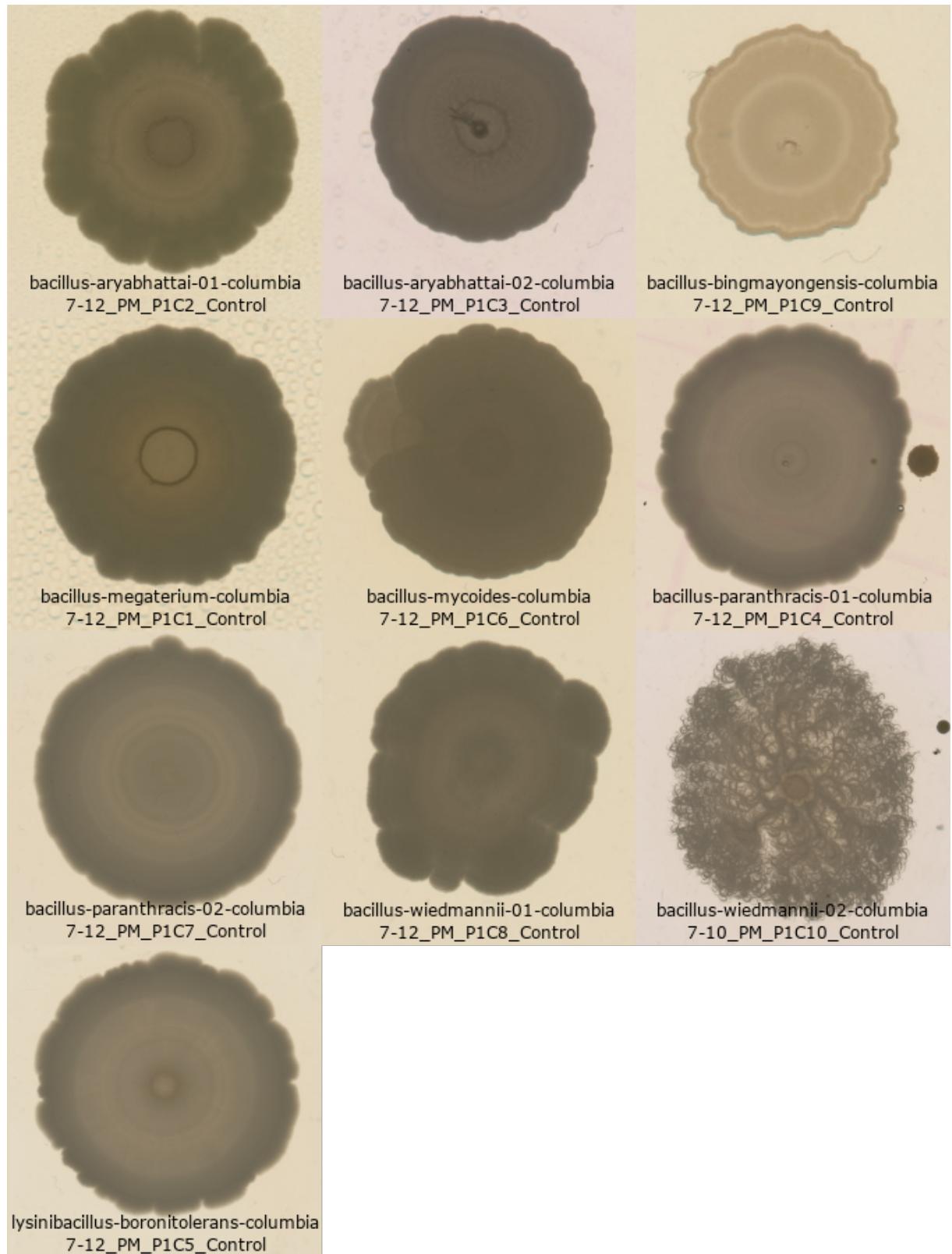


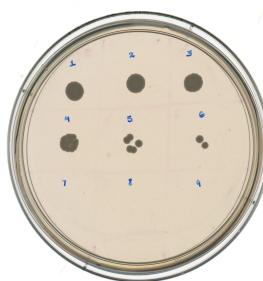
Figura 11 – Imagens de exemplo das diferentes bactérias do conjunto. Fonte: Autor.



(a) Original



(b) Pré-processada

Figura 12 – Bactéria *bacillus-aryabhattai-01-columbia*

(a) Original



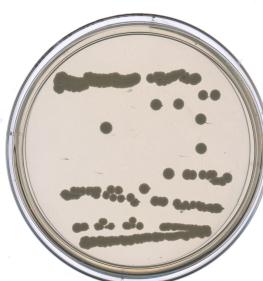
(b) Pré-processada

Figura 13 – Bactéria *bacillus-aryabhattai-02-columbia*

(a) Original



(b) Pré-processada

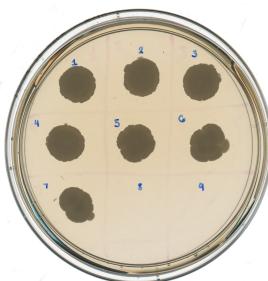
Figura 14 – Bactéria *bacillus-bingmayongensis-columbia*

(a) Original

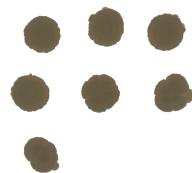


(b) Pré-processada

Figura 15 – Bactéria *bacillus-megaterium-columbia*



(a) Original



(b) Pré-processada

Figura 16 – Bactéria bacillus-mycooides-columbia



(a) Original



(b) Pré-processada

Figura 17 – Bactéria bacillus-parantracis-01-columbia

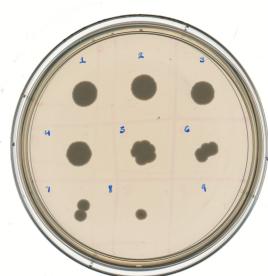


(a) Original

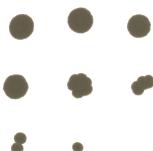


(b) Pré-processada

Figura 18 – Bactéria bacillus-parantracis-02-columbia

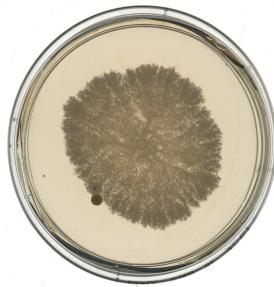


(a) Original



(b) Pré-processada

Figura 19 – Bactéria bacillus-wiedmannii-01-columbia



(a) Original



(b) Pré-processada

Figura 20 – Bactéria bacillus-wiedmannii-02-columbia



(a) Original



(b) Pré-processada

Figura 21 – Bactéria lysinibacillus-boronitolerans-columbia

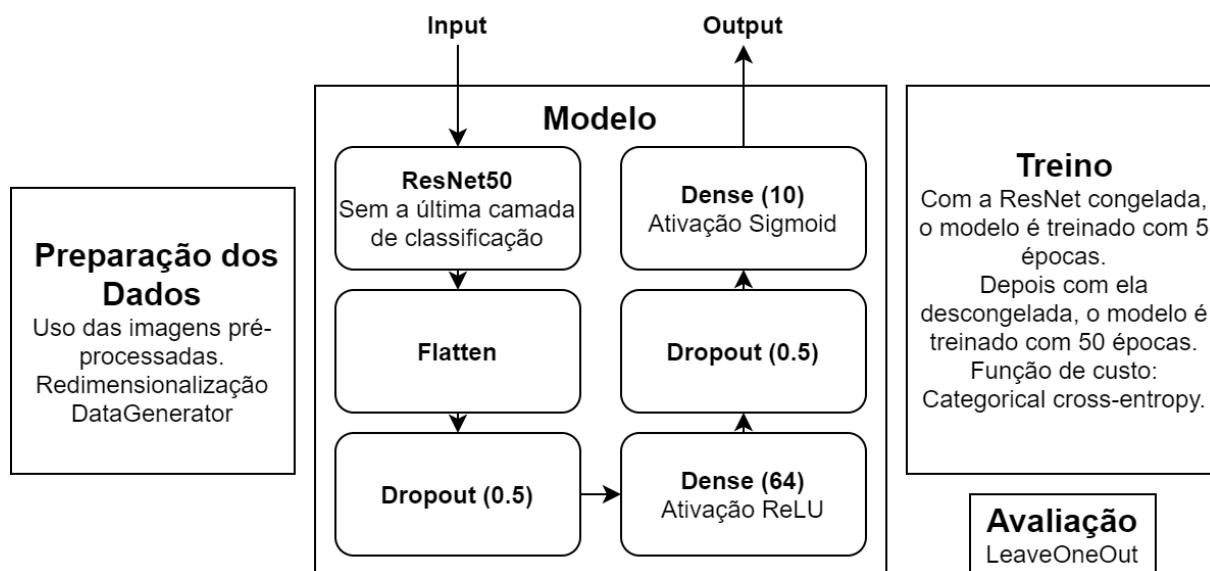


Figura 22 – Resumo dos métodos utilizados. Fonte: Autor.

Parâmetro	Valor	Explicação
rescale	1./255	Reescala as imagens para 0 a 1.
rotation_range	40	Ângulo limite para a rotação da imagem.
width_shift_range	0.2	Quando em um float de 0 a 1, é a porcentagem de translação baseado na largura.
height_shift_range	0.2	Quando em um float de 0 a 1, é a porcentagem de translação baseado na altura.
shear_range	0.2	Límite em porcentagem para a aplicação do shear, uma espécie de inclinação artificial.
zoom_range	0.2	Límite em porcentagem para a aplicação do zoom.
horizontal_flip	True	Flags para inversão horizontal da imagem.
vertical_flip	True	Flags para inversão vertical da imagem.
fill_mode	'nearest'	Indicar qual algoritmo utilizar ao completar espaços não existentes na imagem transformada.

Tabela 1 – Parâmetros utilizados no *Data Augmentation*.

3.3.1 Leitura de imagens e preparação dos dados

Das imagens exemplificadas nas seções anteriores, o mais interessante é utilizar as versões pré-processadas, que já são as que possuem menos informações irrelevantes para a classificação das bactérias. Para isso, é preciso redimensionar as imagens, dado que elas não possuem um tamanho padrão. Elas então foram transformadas em imagens de tamanho (224, 224, 3), que é o padrão utilizado pela Rede Neural Residual (Resnet50).

Outro ponto é a quantidade de imagens, pois cada classe possui apenas 20 amostras, um número relativamente baixo. Nesse trabalho foi empregada a estratégia de *Data Augmentation* (GU; PEDNEKAR; SLATER, 2019), que envolve gerar novas imagens a partir de transformações realizadas na original, como translação e rotação. Na Tabela 1 são apresentados os parâmetros utilizados na estratégia de *Data Augmentation*. Com isso, foi possível gerar 200 novas imagens para cada época de treino.

3.3.2 Modelo e treinamento da Rede Residual ResNet50

A ResNet50² é uma rede neural convolucional residual de 50 camadas treinada no conjunto de imagens denominado ImageNet, contendo 1.281.167 imagens de treinamento em 1000 categorias. Um uso típico desse modelo é o aprendizado por transferência, na qual acredita-se que este modelo contém padrões reaproveitáveis, como bordas, texturas e formatos.

Nesse trabalho foi utilizada a biblioteca Keras, em que possui métodos prontos para inserir a ResNet50 em um modelo, carregando a arquitetura e pesos da rede. Como a ResNet50 é treinada para 1000 classes de imagens diferentes, é necessário retirar a última camada de classificação, pois não há interesse em usar as classes originais do problema. Assim, a ResNet50

² <<https://keras.io/api/applications/resnet/#resnet50-function>>

foi complementada com novas camadas específicas para este problema de classificação de imagens de placas de Petri.

Na Figura 22 é apresentado o esquema da rede neural residual proposta neste trabalho. Na construção do modelo, após a ResNet50 sem a camada de classificação, é inserido uma camada denominada *Flatten*, que simplesmente concatena a saída da ResNet50 um vetor que será utilizado como entrada para as novas camadas. Dessa forma, os padrões embutidos na rede pré-treinada ResNet50 podem ser transferidos como entrada para uma rede neural MLP. Depois vem uma camada Dropout com probabilidade 50%, significando que no treino, um peso possui 50% de chance de ser "ignorado", resultando em um modelo mais robusto e não dependente em neurônios específicos. Após o Dropout vem uma camada Densa, que é uma camada totalmente conectada, assim como uma MLP, com 64 nós e ativação ReLU. Depois outro Dropout com 50% e outra Densa, dessa vez com 10 neurônios e ativação Sigmoid, em que cada neurônio é associado a uma das classes (camada de saída), ou seja, a um tipo de bactéria. Essa estratégia é frequentemente utilizada na literatura para outros problemas de classificação (HE *et al.*, 2019).

Como um treino inicial, os pesos da camada da ResNet50 são congelados (não poderão ser mudados) e a rede é treinada por 5 épocas. Isto é feito para que os pesos das camadas Densas que foram adicionadas, que são iniciados aleatoriamente, sejam inicializados de forma a obter uma convergência mais rápida no próximo estágio de treinamento.

Para avaliar o desempenho de classificação obtido pelo modelo, na rede neural foi utilizada a função de custo Categorical Cross-Entropy, dada pela Equação 3.1,

$$CE = - \sum_{i=1}^M t_i \log(s_i) \quad (3.1)$$

onde M é a quantidade de classes para classificar; t_i é 1 se a imagem atual pertence à classe i e 0 caso contrário; s_i é o valor predito pela rede.

Depois do treino inicial, o último bloco residual da ResNet50 é descongelado e o modelo é treinado novamente, desta vez com 50 épocas. Na Figura 23 é apresentado um gráfico com o histórico de treinamento (Função de Custo (Loss) vs Épocas), demonstrando a convergência da rede.

3.3.3 Avaliação com o Leave-One-Out

Ao avaliar um modelo de aprendizado de máquina, é necessário mensurar sua performance em dados não existentes no treino do modelo. Como a quantidade de dados no conjunto é relativamente baixa, apenas 200 imagens totais, foi escolhido o *Leave-One-Out* como método de avaliação.

O Leave-One-Out (Figura 24) consiste em treinar N modelos, sendo N a quantidade de dados no conjunto. Em cada etapa de treinamento é separada uma imagem diferente para o

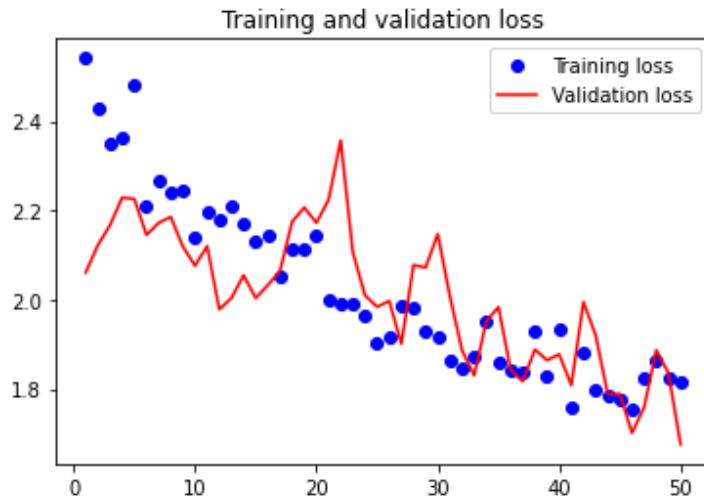


Figura 23 – Erro (Loss) do treino (eixo y) (em função da época (eixo x)).

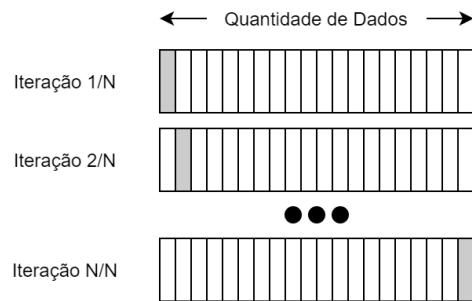


Figura 24 – Funcionamento do *Leave-One-Out*. A parte cinza indica um exemplo de teste e a parte branca os exemplos de treino. Fonte: Autor.

teste/validation e as outras são utilizados como treino. Desta maneira, pode-se gerar estimativas de acurácia, precisão e revocação do modelo de classificação, mesmo com um número baixo de dados.

3.4 Resultados

3.4.1 Desempenho da Classificação

Para avaliar o desempenho, foi realizada uma análise da matriz de confusão obtida da classificação e suas métricas derivadas. A matriz de confusão consiste em uma matriz $M \times M$, sendo M a quantidade de classes na classificação. Cada linha representa a classe real de uma imagem e a coluna a classe em que ela foi classificada pelo modelo, ou seja, a diagonal principal representa a quantidade de vezes em que um dado foi classificado em sua classe real.

A partir da matriz é possível calcular várias métricas. As utilizadas neste projeto foram: accuracy (Equação 3.2), f1 score (Equação 3.3), precision (Equação 3.4) e recall (Equação 3.5). A accuracy é a porcentagem de itens classificados corretamente dentre todos os dados. Precision é a porcentagem de itens corretos dentre aqueles classificados como a classe em questão. Recall

é a porcentagem de itens classificados corretos dentre aqueles da classe em questão. F1 score é a média harmônica do recall e precision. Nessas equações, TP representa *True Positive*, TN representa *True Negative*, FP representa *False Positive* e FN representa *False Negative*.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.2)$$

$$f1_score = \frac{2TP}{2TP + FP + FN} \quad (3.3)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (3.4)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (3.5)$$

Na Tabela 2 são apresentados os resultados obtidos para as métricas: accuracy, f1_score, precision e recall das predições de cada classe.

Classe	Accuracy	F1 Score	Precision	Recall
bacillus-aryabhattai-01-columbia	0.93	0.708	0.607	0.85
bacillus-aryabhattai-02-columbia	0.925	0.634	0.619	0.65
bacillus-bingmayongensis-columbia	1	1	1	1
bacillus-megaterium-columbia	0.915	0.540	0.588	0.5
bacillus-mycoides-columbia	0.935	0.606	0.769	0.5
bacillus-parantracis-01-columbia	0.905	0.536	0.523	0.55
bacillus-parantracis-02-columbia	0.895	0.461	0.473	0.45
bacillus-wiedmannii-01-columbia	0.925	0.615	0.631	0.6
bacillus-wiedmannii-02-columbia	0.995	0.974	1	0.95
lysinibacillus-boronitolerans-columbia	0.925	0.651	0.608	0.7
Média	0.935	0.6725	0.6818	0.675

Tabela 2 – Resultado do treino, por classe. Métricas provenientes da Matriz de Confusão: accuracy, f1 score, precision e recall.

De forma geral, a acurácia de classificação foi considerada satisfatório, com menor acurácia de 0.895 e maior acurácia de 1.0, e acurácia média de 0.935. Por outro lado, ao analisar o desempenho do modelo usando métricas da área de recuperação de informação (F1, precision e recall), são identificadas classes que precisam de mais atenção durante o treinamento, como a bactéria Bacillus-Parantracis. Em geral, seria interessante adquirir mais amostras (imagens) dessas classes.

Para efeito de comparação, um estudo conduzido por ([RIOS AKHIL PUNIA, 2019](#)) usando métodos tradicionais de aprendizado de máquina neste mesmo conjunto de imagens, obteve acurácia média de 0.49 com SVM, 0.73 com Random Forests, e 0.67 com KNN. Por outro lado, eles usaram outros tipos de atributos e métodos de pré-processamento de imagens para extrair informação de textura e formato de objetos na imagem. Os autores de ([RIOS AKHIL PUNIA, 2019](#)) também avaliaram uma Rede Profunda Convolucional, obtendo acurácia

de 0.87. Em uma base de dados com mais imagens (364 no total), os autores também reportaram avanço na acurácia usando uma adaptação da ResNet18 (18 camadas), com acurácia de 0.93; porém com outra distribuição de classes (8 classes de bactérias). Dessa forma, é importante ressaltar que tais comparações são limitadas, uma vez que as técnicas de pré-processamento (Data Augmentation) e distribuição dos folds (validação cruzada) não são as mesmas nos respectivos trabalhos.

3.4.2 Interpretando a Rede Neural

Um modelo normalmente é considerado bom quando suas métricas ultrapassam certo limiar. O motivo de um modelo resultar naquelas métricas é normalmente ignorado, já que ele pode ser tão complexo que se torna inviável sua interpretação. Na prática, o modelo se torna basicamente uma caixa preta de algoritmos e cálculos.

Uma maneira de tentar visualizar o que acontece em uma rede neural, em saber como as decisões afetam as métricas, é utilizando o método LIME (*Local Interpretable Model-agnostic Explanations*), que basicamente explicita quais partes da entrada são mais importantes para a classificação (RIBEIRO; SINGH; GUESTRIN, 2018).

Utilizando o LIME para identificar como as imagens são classificadas, foi escolhido para cada classe um exemplo classificado corretamente e um incorretamente. Na Figura 25 é apresentada uma dessas classes. No geral, é possível observar que o modelo utiliza formas e texturas para classificar corretamente cada imagem, o que era o esperado. Em situações em que o modelo erra, essas formas e texturas não foram utilizadas. O modelo utilizou, nesse caso, espaços vazios na imagem ou partes irrelevantes. Esse resultado é importante para gerar alertas aos especialistas verificarem manualmente esses casos. Também é possível verificar essas situações para modificar algum critério da etapa de treinamento.

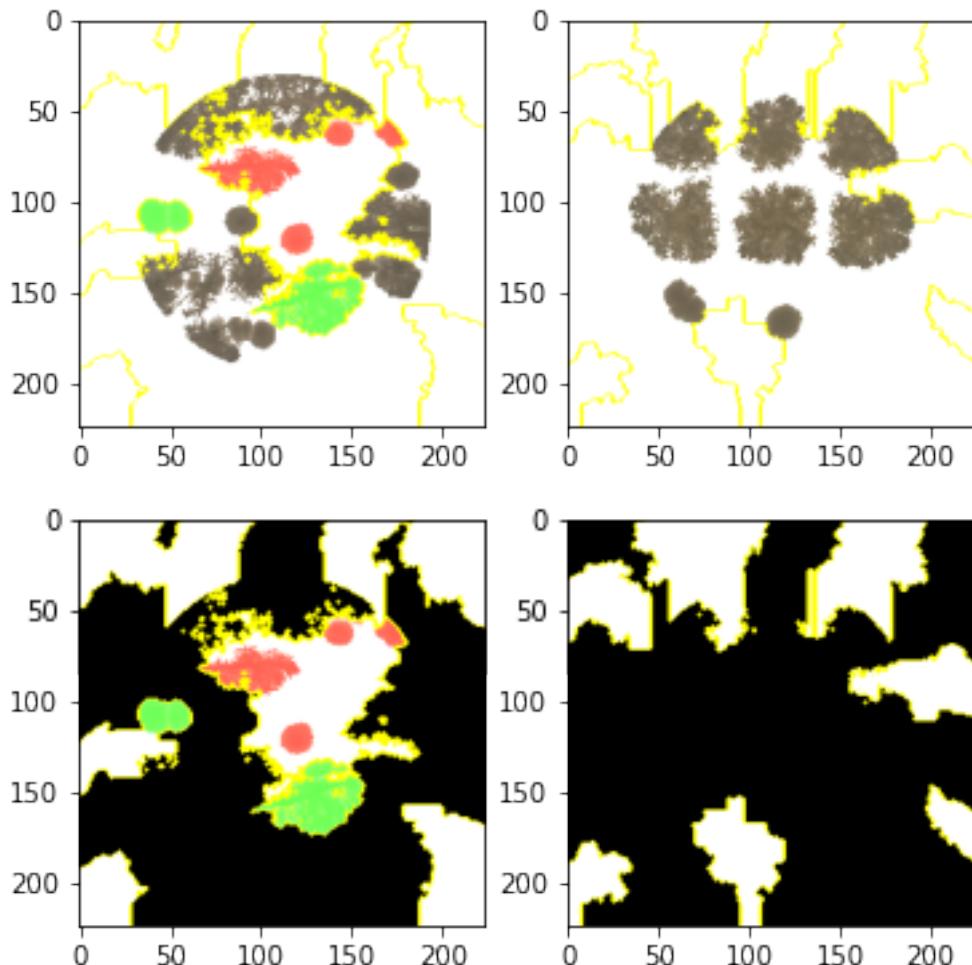


Figura 25 – Exemplo da utilização do LIME na classe *bacillus-wiedmannii-02-columbia*, a imagem esquerda possui um exemplo onde a classificação foi correta, já a direita possui um classificado errado.

Capítulo 4

CONCLUSÃO

4.1 Considerações finais

Redes Residuais Profundas foram investigadas nesse trabalho para classificação de imagens de placas de Petri. O objetivo é avaliar o desempenho desses modelos visando apoiar automatização em análise de colônias de bactérias.

Os resultados alcançados nesse trabalho de conclusão de curso serão utilizados para apoiar um outro projeto em andamento no Labic (Laboratório de Inteligência Computacional do ICMC/USP) que, em colaboração com o Instituto de Química (IQ ICMC/USP), visa a classificação de imagens de placas de Petri para apoiar a construção de laboratórios automatizados. Essa é uma área de pesquisa recente e promissora, especialmente para identificar bactérias (ou outros micro-organismos) que podem ser úteis para fabricação de medicamentos.

Os resultados obtidos foram satisfatórios, uma vez que são competitivos com o estado-da-arte. Além disso, do ponto de vista computacional, o módulo desenvolvido está preparado para ser treinado com novos conjuntos de imagens de maneira facilitada e faz uso de GPU para agilizar o treinamento de modelos.

As principais limitações do trabalho envolvem a interpretabilidade dos resultados, dado que redes neurais são consideradas caixa-preta. Nesse sentido, esse trabalho também avançou em implementar modelos de explicação das classificações realizadas, algo que não era discutido em outros trabalhos relacionados à classificação de bactérias usando Placas de Petri.

4.2 Considerações sobre o curso de graduação

De modo geral, o curso de graduação me ensinou a ter independência, a pesquisar e aprender de maneira própria. A base ensinada pelas disciplinas iniciais, como Introdução a Ciência de Computação (ICC), foram essenciais para meu interesse nesse universo de programação, já que entrei na faculdade sem saber nada sobre.

Em específico para o projeto, as matérias de Aprendizado de Máquina e Redes Neurais, bem como meu Projeto de Conclusão de Curso anterior, foram fundamentais o andamento deste projeto. Os trabalhos das disciplinas ensinaram a como organizar, tratar os dados e principalmente meu último projeto me ensinou a trabalhar com redes neurais.

4.3 Sugestões para o curso de graduação

Assim como meu projeto anterior, reitero o quanto importante foi a disciplina de ICC, a extensa lista de exercícios se transformou em um gosto por descobrir coisas novas e resolver problemas de programação. Portanto espero que o cuidado em como essas disciplinas iniciais são ministradas seja mantido.

4.4 Trabalhos futuros

O desenvolvimento deste trabalho permitiu identificar diferentes frentes para trabalhos futuros. O primeiro ponto de trabalho futuro é aumentar o conjunto de imagens anotadas com colônias em placas de Petri. Idealmente, seria interessante um repositório público que possa ser compartilhado e mantido entre diferentes organizações interessadas em automatizar essa análise.

Uma segunda direção de trabalho futuro é investigar outros modelos de profundas pré-treinadas. Nos últimos três anos houve um grande avanço nessa área, na qual foram disponibilizados outros modelos para imagens como GoogLeNet, Inception e AlexNet. Assim, além da ResNet, esses modelos podem ser utilizados em uma estratégia de comitê de classificadores, permitindo obter uma classificação consensual e mais robusta.

REFERÊNCIAS

- AFFONSO, C.; ROSSI, A. L. D.; VIEIRA, F. H. A.; FERREIRA, A. C. P. de L. *et al.* Deep learning for biological image classification. **Expert Systems with Applications**, Elsevier, v. 85, p. 114–122, 2017. Citado na página 17.
- AHERN, H. **Bacteriological Culture Methods**. 2015. Acessado em Abril de 2020. Disponível em: <<https://milnepublishing.geneseo.edu/suny-microbiology-lab/chapter/bacteriological-culture-methods/>>. Citado 2 vezes nas páginas 11 e 32.
- BOURBEAU, P. P.; LEDEBOER, N. A. Automation in clinical microbiology. **Journal of clinical microbiology**, Am Soc Microbiol, v. 51, n. 6, p. 1658–1665, 2013. Citado 2 vezes nas páginas 17 e 31.
- FERRARI, A.; SIGNORONI, A. Multistage classification for bacterial colonies recognition on solid agar images. In: IEEE. **2014 IEEE International Conference on Imaging Systems and Techniques (IST) Proceedings**. [S.l.], 2014. p. 101–106. Citado 2 vezes nas páginas 17 e 31.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016. Citado 3 vezes nas páginas 21, 23 e 24.
- GU, S.; PEDNEKAR, M.; SLATER, R. Improve image classification using data augmentation and neural networks. **SMU Data Science Review**, v. 2, n. 2, p. 1, 2019. Citado na página 37.
- HAYKIN, S. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice Hall PTR, 1994. Citado 2 vezes nas páginas 21 e 22.
- 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. Citado 5 vezes nas páginas 11, 18, 26, 27 e 28.
- HE, T.; ZHANG, Z.; ZHANG, H.; ZHANG, Z.; XIE, J.; LI, M. Bag of tricks for image classification with convolutional neural networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 558–567. Citado 3 vezes nas páginas 19, 27 e 38.
- LITJENS, G.; KOOI, T.; BEJNORDI, B. E.; SETIO, A. A. A.; CIOMPI, F.; GHAFOORIAN, M.; LAAK, J. A. V. D.; GINNEKEN, B. V.; SÁNCHEZ, C. I. A survey on deep learning in medical image analysis. **Medical image analysis**, Elsevier, v. 42, p. 60–88, 2017. Citado na página 18.
- LIU, L.; OUYANG, W.; WANG, X.; FIEGUTH, P.; CHEN, J.; LIU, X.; PIETIKÄINEN, M. Deep learning for generic object detection: A survey. **International journal of computer vision**, Springer, v. 128, n. 2, p. 261–318, 2020. Citado 4 vezes nas páginas 17, 18, 24 e 31.
- LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. E. A survey of deep neural network architectures and their applications. **Neurocomputing**, Elsevier, v. 234, p. 11–26, 2017. Citado na página 24.

MARCHILDON, R. P. **Building Neural Networks in TensorFlow**. 2017. Acessado em Abril de 2020. Disponível em: <<http://rjmparchildon.com/ai-cnn-digits/>>. Citado na página 26.

NAUGLER, C.; CHURCH, D. L. Automation and artificial intelligence in the clinical laboratory. **Critical reviews in clinical laboratory sciences**, Taylor & Francis, v. 56, n. 2, p. 98–110, 2019. Citado 2 vezes nas páginas 17 e 31.

PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. In: **Tópicos em Gerenciamento de Dados e Informações**. SBC. [S.l.: s.n.], 2017. Citado na página 25.

PRAKASH, A. **Machine Learning - Convolution for image processing**. 2018. Acessado em Abril de 2020. Disponível em: <<https://blog.francium.tech/machine-learning-convolution-for-image-processing-42623c8dbec0>>. Citado na página 25.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. **Neural computation**, MIT Press, v. 29, n. 9, p. 2352–2449, 2017. Citado 2 vezes nas páginas 17 e 18.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. Anchors: High-precision model-agnostic explanations. In: **Thirty-Second AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2018. Citado na página 41.

RIOS AKHIL PUNIA, C. P. Z. D. P. S. X. C. A. **Capstone Project - Bacteria Image Classification and Segmentation**. 2019. Acessado em Abril de 2020. Disponível em: <<https://github.com/deibyrios/bacteria-classification/>>. Citado na página 40.

YANI, M.; IRAWAN, S.; S.T., M. Application of transfer learning using convolutional neural network method for early detection of terry's nail. **Journal of Physics: Conference Series**, v. 1201, p. 012052, 05 2019. Citado na página 25.

ZHANG, Q.; YANG, L. T.; CHEN, Z.; LI, P. A survey on deep learning for big data. **Information Fusion**, Elsevier, v. 42, p. 146–157, 2018. Citado 2 vezes nas páginas 24 e 26.