

LARISSA FERREIRA RODRIGUES

COMPARAÇÃO ENTRE REDES NEURAIS  
CONVOLUCIONAIS E TÉCNICAS DE  
PRÉ-PROCESSAMENTO PARA CLASSIFICAR  
CÉLULAS *HEP-2* EM IMAGENS DE  
IMUNOFLUORESCÊNCIA

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS – BRASIL  
2018

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

R696c  
2018  
Rodrigues, Larissa Ferreira, 1994-  
Comparação entre redes neurais convolucionais e técnicas  
de pré-processamento para classificar células *HEp-2* em imagens  
de imunofluorescência / Larissa Ferreira Rodrigues. – Viçosa,  
MG, 2018.  
xiii, 65 f. : il. (algumas color.) ; 29 cm.

Inclui apêndice.

Orientador: Murilo Coelho Naldi.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 55-62.

1. Redes neurais (Computação). 2. Diagnóstico por  
imagem. 3. Processamento de imagens - Coloração. 4. Células  
epiteliais. 5. Doenças autoimunes - Diagnóstico.  
6. Imunofluorescência. I. Universidade Federal de Viçosa.  
Departamento de Informática. Programa de Pós-Graduação em  
Ciência da Computação. II. Título.

CDD 22. ed. 006.32


LARISSA FERREIRA RODRIGUES

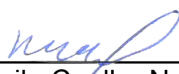
**COMPARAÇÃO ENTRE REDES NEURAIS CONVOLUCIONAIS E  
TÉCNICAS DE PRÉ-PROCESSAMENTO PARA CLASSIFICAR  
CÉLULAS *HEP-2* EM IMAGENS DE IMUNOFLUORESCÊNCIA**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 10 de dezembro de 2018.

  
\_\_\_\_\_  
André Ricardo Backes

  
\_\_\_\_\_  
Alcione de Paiva Oliveira

  
\_\_\_\_\_  
Murilo Coelho Naldi  
(Orientador)

*À minha família*

*“Os teus olhos viram o meu corpo ainda informe.  
Todos os dias que foram ordenados para mim, no teu  
livro foram escritos quando nenhum deles havia ainda.  
Quão preciosos me são, ó Deus, os teus pensamentos!  
Quão vasta é a soma deles!”  
(Salmos 139.16-17)*

# Agradecimentos

Agradeço a Deus, autor da vida, “*em quem estão escondidos todos os tesouros da sabedoria e da ciência*” (Cl 2:3) por me abençoar com coragem, força e perseverança em todos os momentos.

Aos meus pais, Joaquim Roberto Rodrigues e Maria Abadia Ferreira Rodrigues, por todo o amor, apoio, confiança e por nunca pouparem esforços para que eu pudesse alcançar os meus sonhos.

Ao meu irmão Leonardo Gabriel Ferreira Rodrigues, por todo o amor, incentivo e por me fortalecer com suas orações, as quais foram essenciais para a realização deste trabalho.

Ao meu namorado Rodrigo Moreira, por todo o amor, apoio e companheirismo.

Aos meus orientadores Prof. Dr. Murilo Coelho Naldi e Prof. Dr. João Fernando Mari, pela eminente orientação, conversas construtivas e valiosos ensinamentos.

Aos membros da banca avaliadora, Prof. Dr. André Ricardo Backes e Prof. Dr. Alcione de Paiva Oliveira, pelas correções e importantes contribuições para este trabalho.

Ao médico Prof. Dr. José Mario S. Mezêncio por apresentar os desafios existentes no diagnóstico de doenças autoimunes e a necessidade de aplicar técnicas computacionais nesta área.

Ao Prof. Dr. Ricardo dos Santos Ferreira, Prof. Dr. Marcus Vinícius Alvim Andrade, Prof. Me. Maurício Gouvêa Gruppi, Me. Keiller Nogueira e Fernando Ferreira Passe pelas contribuições para a realização deste trabalho.

Aos demais professores e funcionários do Departamento de Informática (DPI) por viabilizarem um excelente ambiente de ensino e pesquisa, em especial aos professores Dr. Jugurta Lisboa Filho, Dr. Alcione de Paiva Oliveira e ao secretário William Silva Souza, por toda presteza.

A *NVIDIA Corporation* pela doação da *GPU Titan Xp* e as instituições de fomento à pesquisa: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) por todo suporte financeiro. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Ao Instituto de Ciências Exatas e Tecnológicas da UFV-CRP, aos professores Dr. Lucas M. Guimarães e Dra. Rejane Nascentes, por fornecer os recursos financeiros necessários para a aquisição de equipamentos e para a apresentação deste

trabalho em conferências.

Ao Analista de Sistemas da UFV-CRP, Henrique A. Peluzio, por toda atenção, disponibilidade e excelência ao atender as demandas que surgiram no decorrer desta pesquisa.

Ao professor Dr. Rodrigo Smarzaro da Silva por todo apoio e prontidão para a realização da minha defesa de mestrado por webconferência. Ao Me. Paulo Gustavo Lopes Cândido por todo o auxílio na entrega da versão final da Dissertação.

A todos os meus familiares por acreditarem nesta conquista, em especial ao meu tio José Fernandes Neto, por me apoiar com o equipamento necessário durante o período de disciplinas do mestrado. Aos meus primos Dorinato Fernandes Ferreira, Márcia Melo (Pina) e Clara Fernandes Melo, pela graciosa companhia de São Gotardo/MG até Viçosa/MG, sendo uma viagem marcada por construtivas conversas e boas risadas.

Aos estimados amigos: Rodrigo Molinari da Costa, Luana G. E. Borges da Costa, Isabella Molinari Borges da Costa e Renata Molinari Borges da Costa, por todo o apoio e carinho.

Ao Prof. Dr. Eduardo Euclides de Lima e Borges e Prof<sup>a</sup>. Dra. Rita de Cassia Gonçalves Borges por todo o carinho, hospitalidade e por me receberem como uma filha durante o período que estive em Viçosa/MG.

Aos colegas de laboratório no DPI e na UFV-CRP, por todo o aprendizado, parcerias e boas conversas.

Aos amigos da Batista da Lagoinha em Viçosa, pela amizade, carinho e por todos os momentos de comunhão.

Ao Rev. João Ricardo Sampaio, Miss. Andreza Moraes Sampaio e todos da Presbiteriana Pentecostal de São Gotardo, por todo apoio, carinho e orações.

A todos que colaboraram de alguma forma, direta ou indiretamente, com o desenvolvimento deste trabalho. Obrigada!

# Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Resumo	x
Abstract	xii
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos	3
1.2 Contribuições	3
1.3 Organização da Dissertação	3
<b>2 Fundamentação Teórica e Trabalhos Correlatos</b>	<b>5</b>
2.1 Doenças Autoimunes	5
2.1.1 Teste de Fator Antinúcleo (FAN) por fluorescência	6
2.2 Processamento e Análise de Imagens	8
2.2.1 Pré-processamento e Correção de Imagens	8
2.2.2 Aumento do Conjunto de Dados	11
2.3 Redes Neurais	11
2.3.1 Tipos de Função Ativação	14
2.3.2 Estrutura Profunda do Neocórtex	15
2.3.3 Aprendizado Profundo ( <i>Deep Learning</i> )	16
2.4 Redes Neurais Convolucionais ( <i>CNNs</i> )	17
2.4.1 Camada Convolucional	17
2.4.2 Camada de <i>Pooling</i>	18
2.4.3 Camada Totalmente Conectada	19
2.4.4 Arquiteturas <i>CNNs</i>	19
2.4.5 Estratégias de Treinamento das <i>CNNs</i>	22
2.5 Otimização de Parâmetros	23
2.6 Otimização de Hiperparâmetros	24
2.6.1 <i>Tree of Parzen Estimators (TPE)</i>	25
2.7 Avaliação do Desempenho das <i>CNNs</i>	26
2.7.1 Validação Cruzada com <i>k-fold</i>	26
2.7.2 Matriz de Confusão	27

2.8	Trabalhos Correlatos . . . . .	28
<b>3</b>	<b>Material e Métodos</b>	<b>31</b>
3.1	Conjunto de Imagens . . . . .	32
3.2	Particionamento do Conjunto de Imagens . . . . .	33
3.3	Pré-processamento das Imagens . . . . .	33
3.4	Estimativa de Hiperparâmetros . . . . .	34
3.5	<i>CNNs</i> e Estratégias de Treinamento . . . . .	35
3.6	Critério de Avaliação . . . . .	37
<b>4</b>	<b>Resultados e Discussões</b>	<b>38</b>
4.1	Impacto do Pré-processamento . . . . .	38
4.2	Impacto do Aumento do Conjunto de Dados . . . . .	42
4.3	Comparação entre diferentes arquiteturas . . . . .	43
4.4	Treinamento Completo <i>Vs Fine-Tuning</i> . . . . .	49
4.5	Comparação com Trabalhos Relacionados . . . . .	52
<b>5</b>	<b>Conclusões</b>	<b>53</b>
	<b>Referências Bibliográficas</b>	<b>55</b>
	<b>Apêndice A Lista de Publicações</b>	<b>63</b>

# Lista de Figuras

2.1	Imagem obtida pelo método IFI . . . . .	6
2.2	Seis padrões de coloração mais comuns em células <i>HEp-2</i> . . . . .	7
2.3	Representação de uma imagem digital . . . . .	8
2.4	Realce de contraste em imagem de célula <i>HEp-2</i> . . . . .	10
2.5	Diagrama esquemático de um neurônio biológico . . . . .	12
2.6	Modelo de um perceptron . . . . .	13
2.7	Exemplo de Rede <i>MLP</i> . . . . .	13
2.8	Funções de ativação mais populares . . . . .	15
2.9	Representação das seis camadas do neocórtex humano . . . . .	16
2.10	Ilustração da estrutura de uma típica camada de convolução. . . . .	18
2.11	Ilustração das computações na camada de <i>pooling</i> . . . . .	19
2.12	Arquiteturas <i>CNNs</i> avaliadas neste trabalho . . . . .	22
2.13	Visão geral do método <i>k-fold</i> . . . . .	26
2.14	Visão geral de uma matriz de confusão . . . . .	27
3.1	Etapas do método proposto . . . . .	31
3.2	Representação gráfica da validação cruzada com <i>k-fold</i> . . . . .	33
3.3	Melhoria de contraste na imagem de célula <i>HEp-2</i> . . . . .	34
3.4	Arquiteturas <i>CNNs</i> em que aplicou-se <i>fine-tuning</i> . . . . .	36
4.1	Acurácia da validação com <i>5-fold</i> para cada modelo <i>CNN</i> . . . . .	40
4.2	Evolução dos valores de perda ( <i>loss</i> ) e acurácia para o conjunto de validação	41
4.3	Acurácia da validação com <i>5-fold</i> para cada modelo <i>CNN</i> considerando o aumento do conjunto de dados . . . . .	42
4.4	Exemplos de imagens classificadas de forma correta e incorreta de acordo com as matrizes de confusão . . . . .	48

# Lista de Tabelas

3.1	Informações sobre cada classe do conjunto de dados ICPR 2014 . . . . .	32
3.2	Espaço de pesquisa de hiperparâmetro considerado para otimização . . .	35
3.3	Resultado dos hiperparâmetros otimizados para cada CNN com o <i>TPE</i> .	35
4.1	Acurácia da validação e tempo médio de treinamento completo para cada <i>CNN</i> . . . . .	39
4.2	Acurácia da validação e tempo médio de treinamento para cada <i>CNN</i> considerando aumento do conjunto de dados . . . . .	43
4.3	Acurácia média do <i>5-fold</i> e acurácia do teste para cada modelo <i>CNN</i> , considerando a melhor estratégia de pré-processamento. . . . .	44
4.4	Matriz de confusão do melhor resultado para cada <i>CNN</i> . . . . .	45
4.5	Acurácia da validação e tempo médio de treinamento para cada <i>CNN</i> considerando <i>fine-tuning</i> sem aumento do conjunto de dados . . . . .	49
4.6	Acurácia da validação e tempo médio de treinamento para cada <i>CNN</i> considerando <i>fine-tuning</i> e aumento de dados . . . . .	50
4.7	Acurácia da validação e tempo médio de treinamento para cada <i>CNN</i> considerando a melhor estratégia de pré-processamento . . . . .	50
4.8	Sumarização de todos os experimentos . . . . .	51
4.9	Comparação com Trabalhos Relacionados . . . . .	52

# Resumo

RODRIGUES, Larissa Ferreira, M.Sc., Universidade Federal de Viçosa, dezembro de 2018. **Comparação entre Redes Neurais Convolucionais e técnicas de pré-processamento para classificar células *HEp-2* em imagens de imunofluorescência.** Orientador: Murilo Coelho Naldi. Coorientador: João Fernando Mari.

As doenças autoimunes são a terceira causa de mortalidade no mundo. A identificação de autoanticorpos nucleares por meio do teste de imunofluorescência indireta (IFI) em células epiteliais humanas (*HEp-2*) é um método convencional para auxiliar no diagnóstico e monitoramento de tais doenças. Entretanto, análises manuais de células *HEp-2* por IFI foram feitas ao longo dos anos, mas são subjetivas e demoradas. Portanto, a necessidade de métodos automatizados e padronizados é reconhecida e investigada há muito tempo e o desenvolvimento de sistemas de diagnóstico médico auxiliado por computador baseados em processamento de imagens e aprendizado de máquina se tornaram fundamentais. O principal objetivo deste trabalho consiste em avaliar cinco modelos de Redes Neurais Convolucionais (*CNNs*): LeNet-5, AlexNet, Inception-V3, VGG-16 e ResNet-50, para a tarefa de classificação automática de células *HEp-2* e o impacto de seis estratégias de pré-processamento baseadas em alargamento de contraste, centralização de dados e aumento do conjunto de dados. Adicionalmente, a otimização de hiperparâmetros utilizando *Tree of Parzen Estimators (TPE)* é estudada junto a duas possíveis estratégias para explorar o poder das *CNNs* em diferentes cenários: treinamento completo e *fine-tuning*. Todos os experimentos foram realizados utilizando o conjunto de dados *HEp-2* do *ICPR 2014* composto por 13.596 imagens categorizadas em 6 classes. O desempenho da classi-

ficação foi avaliado por meio da validação cruzada estratificada com *5-fold* sobre o conjunto de treino e a comparação entre os modelos *CNN* considerou o conjunto de teste. O melhor resultado, em termos de acurácia, foi alcançado treinando o modelo Inception-V3 a partir do zero, sem pré-processamento e com aumento de dados, com valores situados em 98,28%, que supera os resultados apresentados por outros trabalhos na literatura. Os resultados apontaram que a maioria das *CNNs* tem melhor desempenho sobre imagens não pré-processadas quando treinadas a partir do zero e com aumento de dados. Embora a técnica de *fine-tuning* pareça promissora, ela não obteve resultados melhores do que as *CNNs* treinadas a partir do zero. Talvez, isso seria considerado em cenários em que o tempo de treinamento é um problema, pois a técnica exige menor tempo computacional. Os métodos desenvolvidos neste trabalho podem auxiliar os especialistas em saúde a escolherem corretamente um método de classificação para identificar e acompanhar condições autoimunes e as doenças que as causam.

# Abstract

RODRIGUES, Larissa Ferreira, M.Sc., Universidade Federal de Viçosa, December, 2018. **Comparison between Convolutional Neural Networks and pre-processing techniques to classify HEp-2 cells in immunofluorescence images.** Advisor: Murilo Coelho Naldi. Co-advisor: João Fernando Mari.

Autoimmune diseases are the third cause of mortality in the world. The identification of antinuclear antibody via indirect immunofluorescence (IIF) test in Human Epithelial-2 (HEp-2) cells is a conventional method to support the diagnosis and monitoring of such diseases. However, manual analyses of HEp-2 cells by IIF have been done along years, but it is subjective and time-consuming. The need for automated and standardized methods is known and persecuted for a long time, and the development of computer-aided diagnosis systems based on image processing and machine learning techniques are fundamental. The main objective of this research resides in assess five Convolutional Neural Networks (CNNs) models: LeNet-5, AlexNet, Inception-V3, VGG-16 and ResNet-50, for this task of automatic classification of HEp-2 cells and the impact of six different pre-processing strategies based on contrast improvements, data centralization, and data augmentation. Additionally, the hyperparameters optimization using Tree of Parzen Estimators (TPE) is applied, and two possible strategies for exploiting the power of existing CNNs in different scenarios are analyzed : from scratch and fine-tuning. All experiments were performed using the HEp-2 dataset from ICPR 2014 composed by 13,596 images classified in six different classes. The classification performance was evaluated using stratified 5-fold cross-validation over the training set, and the comparison among the CNN models considered the test set. The best result, in terms of accuracy, was

achieved by training the Inception-V3 model from scratch without preprocessing and with data augmentation, with values lying on 98.28%, which outperforms the results presented in other works in literature. The results pointed out that most of CNNs perform better over non-preprocessed images when trained from scratch and data augmentation. Although fine-tuning technique sounds promising, it did not achieve better results than the CNNs trained from scratch. Maybe, it would be considered in scenarios where training time is an issue, as its computational needs are lower. The methods developed in this research can help health agents to correctly choose a classification method to identify and manage autoimmune conditions and the diseases that cause them.

# Capítulo 1

## Introdução

A aplicação de técnicas de visão computacional, especificamente de reconhecimento de padrões, estão na base do diagnóstico assistido por computador em um número considerável de aplicações que apoiam e facilitam a tomada de decisão por especialistas em medicina. Estudos em diversos contextos estão sendo desenvolvidos em diferentes áreas: detecção e classificação de microcalcificações mamárias em imagens de mamografia [Cheng et al., 2006; Jalalian et al., 2013]; análise de eritrócitos para identificação de infecção pelo parasita da malária [Díaz et al., 2009; Loddo et al., 2018]; doença falciforme e outras anemias [González-Hidalgo et al., 2015; Gual-Arnau et al., 2015; Rodrigues et al., 2016a]; classificação automática de leucócitos para diagnóstico de leucemia [Scotti, 2005; Agaian et al., 2014; Neoh et al., 2015; Rodrigues et al., 2016b; Faria et al., 2018]; identificação de anormalidades em imagens de endoscopia [Cong et al., 2015; Moccia et al., 2018], entre outros.

Há um certo interesse em aplicar tais técnicas para identificação de doenças autoimunes. Essas doenças causam uma condição na qual o sistema imunológico ataca e destrói por engano as células e tecidos saudáveis do paciente [Mackay & Rose, 2013]. Um aumento significativo no número de casos de doenças autoimunes ocorreu nas últimas décadas, o que as tornam a terceira causa mais comum de mortalidade, depois das doenças cardiovasculares e câncer [Chatenoud, 2016]. Não há cura para as doenças autoimunes e o tratamento consiste em aliviar os sintomas por meio de drogas anti-inflamatórias e imunossupressoras [Bach, 2002; Chatenoud, 2016].

Um dos procedimentos disponíveis para a identificação de doenças autoimunes é a análise visual dos padrões de coloração nas imagens de células epiteliais humanas (do inglês *Human Epithelial-2 - HEp-2*) submetidas ao método de imunofluorescência indireta (IFI). A análise de células *HEp-2* por IFI tem sido feita manualmente ao longo dos anos, mas é subjetiva e demorada [Solomon et al., 2002; Fritzler, 2011; Tozzoli et al., 2013]. A necessidade de métodos automatizados e padronizados é conhecida e investigada há um bom tempo e o desenvolvimento de sistemas de diagnóstico auxiliado por computador baseados em processamento de imagens e técnicas de aprendizado de máquina são fundamentais [Foggia et al., 2014; Manivannan et al., 2016]. Geralmente, soluções baseadas em análise automática de imagens têm menor custo financeiro do que os métodos tradicionais de diagnóstico e os dados coletados

podem facilmente ser compartilhados e processados em qualquer lugar [Doi, 2007] [Dougherty, 2009].

Todos os dias, novas imagens IFI de células *HEp-2* são obtidas em todo o mundo, criando conjuntos crescentes de dados que se complementam. A geração de enormes quantidades de dados, especialmente imagens, permitiu o sucesso das técnicas de aprendizado profundo. Em particular, as Redes Neurais Convolucionais (do inglês *Convolutional Neural Networks - CNN*) vêm demonstrando bons resultados em tarefas gerais de reconhecimento visual, motivando sua ampla aplicação em vários campos de pesquisa [Krizhevsky et al., 2012; Sharif Razavian et al., 2014]. A base do sucesso da *CNN* está na sua capacidade de processamento de imagens em múltiplos estágios para extrair representações hierárquicas de alto nível dos dados. A combinação de *CNNs* com estratégias de pré-processamento, permite revelar várias doenças autoimunes como, mas não se limitando a: lúpus eritematoso sistêmico, artrite reumatóide, esclerose múltipla, hepatite autoimune e diabetes [Egerer et al., 2010].

Recentemente, Gao et al. [2017] apresentaram um método para classificar células *HEp-2* baseado em uma *CNN* profunda inspirada na clássica rede LeNet-5 [Lecun et al., 1998]. Este método foi o primeiro avanço em direção às técnicas estado da arte para classificação de células *HEp-2* com *CNN*.

No entanto, a metodologia de validação usada em Gao et al. [2017] consiste em uma variação de *hold-out* simples [Devijver & Kittler, 1982], que consiste em atribuir aleatoriamente pontos de dados aos conjuntos de treinamento e teste. Essa abordagem pode criar aleatoriamente dois conjuntos tendenciosos, o que pode resultar em modelos de *CNN* anormais, isto é, *CNNs* que não se comportam com a normalidade das *CNNs* treinadas com o mesmo tipo de dados. Apesar de possíveis, tais resultados são improváveis de ocorrer em uma aplicação real do classificador nos dados, pois o erro de classificação pode se distanciar do erro real. Além disso, os valores dos hiperparâmetros também influenciam os resultados das *CNNs* e devem ser ajustados para cada tipo de dado e aplicação.

Da ciência da autora, esta dissertação é o primeiro trabalho a classificar células *HEp-2* utilizando diferentes modelos de *CNNs*, além da LeNet-5, são comparados os modelos: AlexNet [Krizhevsky et al., 2012], Inception-V3 [Szegedy et al., 2016], VGG-16 [Simonyan & Zisserman, 2014a] e ResNet-50 [He et al., 2016], utilizando seis diferentes estratégias de pré-processamento. Para uma melhor estimativa de erro, é adotado um processo de validação cruzada por meio da técnica *k-fold*, um estimador mais confiável sobre a acurácia das arquiteturas *CNNs* avaliadas. Além disso, a análise do impacto da otimização de hiperparâmetros e avaliação de estratégias de treinamento a partir do zero e *fine-tuning* são realizadas para comparar as cinco arquiteturas *CNNs*.

Acredita-se que este estudo pode ajudar os especialistas em medicina a escolher adequadamente um método de classificação para identificar e acompanhar as condições autoimunes e as doenças que as causam.

## 1.1 Objetivos

O objetivo principal deste trabalho foi avaliar diferentes abordagens baseadas em *CNNs* para classificar automaticamente células *HEp-2* em imagens de imunofluorescência e analisar como diferentes estratégias de pré-processamento e otimização automática de hiperparâmetros influenciam no desempenho da classificação.

Especificamente pretende-se:

- Avaliar o impacto de diferentes estratégias de pré-processamento de imagens com base em melhorias de contraste, centralização dos dados e aumento do conjunto de dados.
- Comparar cinco diferentes arquiteturas de *CNNs*: LeNet-5, AlexNet, Inception-V3, VGG-16 e ResNet-50, em termos de custo computacional de treinamento e acurácia da classificação.
- Avaliar e analisar as estratégias de treinamento completo e *fine-tuning* para explorar as *CNNs*.
- Definir os melhores hiperparâmetros utilizando o método de otimização *Tree of Parzen Estimators (TPE)* para cada *CNN* avaliada.
- Alcançar resultados superiores ou iguais aos obtidos pelos trabalhos mais recentes presentes na literatura, sendo uma atualização para o estado da arte na classificação automática de imagens de células *HEp-2*.

## 1.2 Contribuições

Ao final dessa pesquisa foram desenvolvidos métodos para classificar automaticamente imagens de células *HEp-2* utilizando diferentes modelos *CNNs* e otimização de hiperparâmetros com o método *TPE*. Foram avaliadas seis estratégias de pré-processamento baseadas em alargamento de contraste, normalização, aumento dos dados e o impacto sobre a performance das *CNNs*. Para verificar a eficiência dos métodos propostos, todos os experimentos foram realizados com validação cruzada *k-fold*, gerando resultados mais robustos e confiáveis.

## 1.3 Organização da Dissertação

Essa dissertação está organizada em 5 capítulos. Este capítulo introduz o tema, objetivos e principais contribuições. O Capítulo 2 é composto pela fundamentação teórica necessária para compreensão do trabalho e descrição dos principais trabalhos correlatos. No Capítulo 3 são descritos os materiais utilizados, os métodos desenvolvidos e como foram conduzidos e avaliados os experimentos. Os resultados e discussões são relatados no Capítulo 4. Por último, as conclusões obtidas durante o desenvolvimento da dissertação e possíveis temas para trabalhos futuros são apresentados no Capítulo 5.

As publicações decorrentes deste trabalho de forma direta ou indireta são listadas no Apêndice A.

## Capítulo 2

# Fundamentação Teórica e Trabalhos Correlatos

Este capítulo apresenta os aspectos teóricos que embasam este trabalho, bem como os trabalhos correlatos. Inicialmente são conceituadas as doenças autoimunes e seus diagnósticos por meio da análise de imagens de imunofluorescência. É feita uma revisão das principais técnicas e algoritmos de processamento de imagens, ferramenta chave para o diagnóstico médico auxiliado por imagens. São introduzidos os conceitos básicos de Redes Neurais Artificiais e uma visão geral sobre Aprendizado Profundo, detalhando as Redes Neurais Convolucionais e as técnicas aplicadas para avaliar os modelos construídos. Após, é discutida a importância da otimização de hiperparâmetros. Por fim, são apresentados os principais trabalhos relacionados.

### 2.1 Doenças Autoimunes

As doenças denominadas autoimunes (como: lúpus eritematoso sistêmico, artrite reumatoide, esclerose múltipla, hepatite autoimune e diabetes) são enfermidades crônicas e de causas desconhecidas [Egerer et al., 2010]. Essas doenças ocorrem quando o sistema imunológico, que é responsável por defender o organismo, começa a reagir contra os próprios órgãos, tecidos ou células saudáveis equivocadamente, levando à sua destruição ou prejudicando sua função adequada, o que é resultado da produção de autoanticorpos.

Não existe teste laboratorial ou manifestação clínica que seja característica exclusiva de uma única doença autoimune, o que torna o diagnóstico ainda mais difícil [Mackay & Rose, 2013] e as causas de tais doenças ainda não está completamente elucidada, mas acredita-se que envolva a interação de fatores genéticos, agentes infecciosos e alterações em processos de regulação do sistema autoimune [Benoist & Mathis, 2001; Rioux & Abbas, 2005; Chervonsky, 2010; Mackay & Rose, 2013]. Também há a hipótese de que as doenças autoimunes podem ser atribuídas a alterações intrínsecas nas células-tronco presentes na medula óssea [Ikehara, 2008]. O tratamento se concentra em aliviar os sintomas por meio de medicamentos anti-inflamatórios e imunossupressores, sendo que ainda não há cura para as doenças

autoimunes [Chatenoud, 2016].

### 2.1.1 Teste de Fator Antinúcleo (FAN) por fluorescência

A pesquisa de anticorpos contra antígenos celulares pela técnica de Imunofluorescência Indireta (IFI), tradicionalmente denominada como teste de Fator Antinúcleo (FAN), é um excelente exame para triagem de autoanticorpos indicado para identificar doenças autoimunes [Foggia et al., 2014]. O teste de FAN, em geral, utiliza como substrato as células *HEp-2* (do inglês *Human Epithelial-2*) que originam-se de células tumorais de carcinoma de laringe humana, cultivadas em monocamadas sobre lâminas de vidro e que apresentam antígenos bem expressos nas diversas fases do ciclo celular [Moore et al., 1955; Dellavance et al., 2002].

O substrato de células *HEp-2* é usado para testar o soro diluído e inclui as seguintes etapas: i) aquisição das imagens; ii) detecção de mitoses; iii) classificação da intensidade de fluorescência; e iv) reconhecimento do padrão de coloração. Sendo esta última a mais importante, porque dependendo da história clínica do paciente, cada padrão de coloração pode se relacionar com uma doença autoimune específica [Meroni & Schur, 2010; Wiik et al., 2010; Egerer et al., 2010].

A análise de células *HEp-2* por IFI tem sido feita manualmente ao longo dos anos e caracteriza-se como uma tarefa demorada e suscetível a subjetividade [Solomon et al., 2002; Fritzler, 2011; Tozzoli et al., 2013]. A Figura 2.1, mostra um exemplo de amostra obtida pelo método IFI; células mitóticas (destacadas com círculos vermelhos), alguns artefatos decorrentes do processo de aquisição (destacados com círculos brancos) e células normais (objetos restantes), revelam alguns padrões de coloração de fluorescência que são relevantes no diagnóstico.

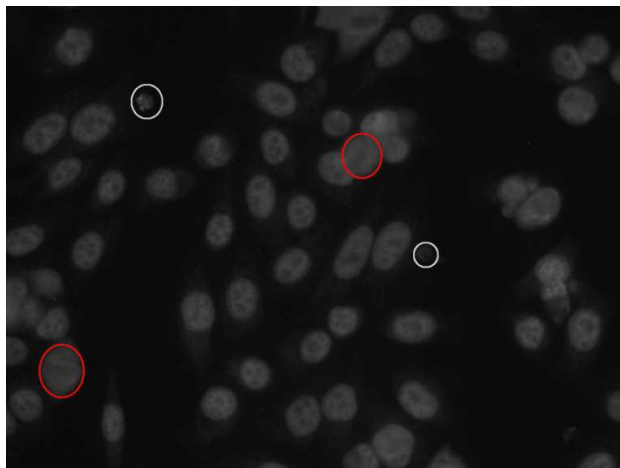


Figura 2.1: Imagem obtida pelo método IFI. Disponível no conjunto de dados ICPR 2014.

Podem ser vistos diversos padrões de coloração, o presente trabalho se concentra em classificar seis padrões mais comuns: *centromere* (centromérico), *golgi*, *homogeneous* (homogêneo), *nucleolar*, *nuclear membrane* (membrana nuclear) e *speckled*

(pontilhado), os quais são brevemente descritos a seguir [Wiik et al., 2010; Hobson et al., 2014] e ilustrados na Figura 2.2.

- *Centromere*: apresenta discretos e uniformes pontilhados em todo o núcleo da célula *HEp-2*; o nucléolo normalmente é não fluorescente. A célula em divisão apresenta concentração dos pontos na placa metafásica e citoplasma não fluorescente.
- *Golgi*: o núcleo da célula *HEp-2* é parcialmente envolvido por uma coloração na organela polar adjacente. O citoplasma das células em divisão apresenta uma coloração difusa e grânulos grandes e irregulares.
- *Homogeneous*: apresenta nucleoplasma fluorescente e o núcleo é considerado não reagente. A placa metafásica cromossômica é intensamente corada, com decoração homogênea e citoplasma, normalmente, não fluorescente.
- *Nucleolar*: os grânulos grandes agrupados nos nucléolos das células em interfase tendem à homogeneidade, com menos de seis grânulos por célula. O citoplasma da célula em divisão pode ser ligeiramente positivo.
- *Nuclear Membrane*: toda a membrana nuclear apresenta fluorescência e em células metafásicas, a fluorescência é difusamente localizada no citoplasma. O material cromossômico não é corado.
- *Speckled*: apresenta nucléolo, célula em divisão e citoplasma não fluorescentes. Esse padrão é dividido em duas categorias: *coarse speckled* (pontilhado grosso) e *fine speckled* (pontilhado fino). *Coarse speckled* consiste em nucleoplasma com grânulos de aspecto grosseiro, heterogêneos em tamanho e brilho. *Fine speckled* consiste em nucleoplasma com granulação fina e os núcleos de células em telófase, ou seja, na fase final da mitose podem ser positivos e algumas vezes mais corados do que os núcleos de células em interfase.

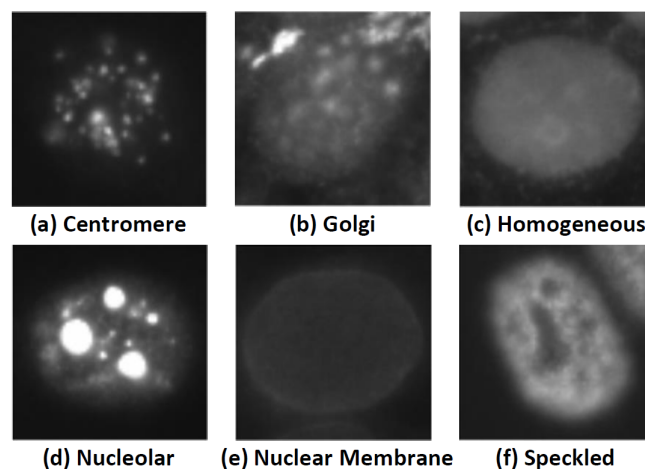


Figura 2.2: Seis padrões de coloração mais comuns em células *HEp-2*. Imagens disponíveis no conjunto de dados ICPR 2014.

## 2.2 Processamento e Análise de Imagens

Os métodos de processamento digital de imagens decorrem de duas áreas principais de aplicação: melhoria de informação visual para a interpretação humana e o processamento de dados de cenas para percepção automática por meio de máquinas. Uma imagem digital é definida como uma função bidimensional  $(x, y)$  em que  $x$  e  $y$  são coordenadas espaciais e o valor de  $f$  para qualquer par de coordenadas  $(x, y)$  representa a intensidade ou nível de cinza da imagem [Gonzalez & Woods, 2007]. Dessa forma, uma imagem digital é composta por um número finito de elementos, denominados *pixels*, conforme ilustrado na Figura 2.3.

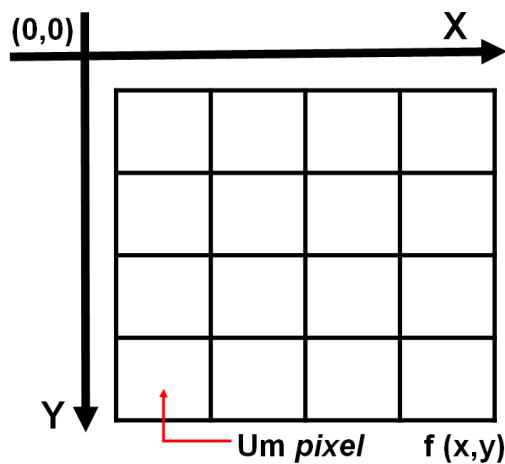


Figura 2.3: Representação de uma imagem digital.

Com o advento do uso de microscópios digitais, o processamento digital de imagens tornou-se essencial para o sucesso da análise e interpretação de dados, sendo que há modalidades de imagens de microscopia que estão disponíveis para visualização somente após a realização de um processamento digital. Dessa forma, o uso de técnicas de processamento digital de imagens proporcionam avanços na medicina e a possibilidade de diagnóstico clínico avançado [Wu et al., 2008].

### 2.2.1 Pré-processamento e Correção de Imagens

As imagens de microscopia podem exigir a aplicação de uma etapa de pré-processamento porque durante a aquisição da imagem, mesmo que o microscópio encontra-se perfeitamente estável, pode ocorrer movimentos indesejáveis e gerar imagens com ruídos ou imperfeições [Wu et al., 2008]. A ausência dessa etapa pode gerar um processamento de imagens difícil ou impossível, além de resultados imprecisos.

O pré-processamento tem como principal objetivo melhorar a qualidade da imagem para a execução das etapas subsequentes. Operações realizadas nessa etapa são consideradas de baixo nível, pois trabalham diretamente sobre os valores de intensidade dos *pixels*. Portanto, nessa etapa o melhoramento das imagens de mi-

croscopia exige o uso de métodos capazes de aproveitar todas as informações multi-dimensionais e multi-espectrais a fim de obter uma imagem digitalizada com melhor qualidade que a original.

Os principais métodos para melhorar imagens de microscopia podem ser classificados como: métodos de domínio espacial e métodos do domínio de transformação, os quais são brevemente descritos a seguir [Gonzalez & Woods, 2007; Wu et al., 2008].

- **Métodos do domínio de transformação:** consiste no uso de métodos que manipulam informações em domínios de transformação, como as Transformadas de Fourier e *Wavelet*. Esses métodos permitem isolar importantes informações (que não podem ser separadas no domínio espacial) no domínio de transformação.
- **Métodos de domínio espacial:** incluem operações que são realizadas em toda a imagem ou em uma região escolhida com base nas estatísticas da imagem. As técnicas pertencentes a essa categoria incluem: equalização de histograma, subtração pela imagem média, detecção de bordas e filtragem não linear. O presente trabalho aplica métodos de processamento de imagem dessa categoria.

Vários métodos para realçar imagens, pertencentes ao domínio espacial, são baseados na manipulação de histograma. O histograma de uma imagem digital em níveis de cinza é computado como uma função  $p(r_k)$ , definida pela Equação 2.1,

$$p(r_k) = \frac{n_k}{n} \quad (2.1)$$

em que  $r_k$  é o  $k$ -ésimo nível de cinza,  $n_k$  é o número de *pixels* na imagem com nível de cinza  $r_k$  e  $n$  é o número total de *pixels*. O histograma de uma imagem fornece informações sobre o contraste da imagem e descreve a distribuição estatística dos níveis de cinza nos *pixels* [Gonzalez & Woods, 2007].

A normalização de histograma é a forma mais simples de melhorar o contraste de uma imagem, pois permite a distribuição de intensidade dos *pixels* ao longo de toda a faixa de valores válidos para a imagem. Por exemplo,  $[0, 255]$  em imagens de 8 bits [Telea, 2014]. Isso é definido pela Equação 2.2,

$$f(x) = \frac{x - I_{min}}{I_{max} - I_{min}} \quad (2.2)$$

em que  $f(x)$  é a função de transferência que normaliza as intensidades da imagem e  $I_{min}$  e  $I_{max}$  representam a amplitude da luminosidade da imagem, sendo as intensidades mínima e máxima, respectivamente.

Neste trabalho são propostas uma série de estratégias de pré-processamento baseadas em combinações de realce de contraste e normalização de imagem. Essas estratégias combinam alargamento de contraste, equalização de histograma e subtração pela imagem média, as quais são descritas a seguir.

- **Alargamento de Contraste:** é uma função de transformação linear simples que mapeia as intensidades mínimas em uma imagem para 0 e os valores máximos para 1, enquanto os valores intermediários são linearmente mapeados no intervalo  $[0,1]$  de seus valores originais [Gonzalez & Woods, 2007].
- **Equalização de Histograma:** a equalização de histograma constrói uma função linear por partes baseada no histograma da imagem normalizada, o que implica em uma imagem com um histograma uniforme [Gonzalez & Woods, 2007].
- **Subtração pela Média da Imagem:** a subtração pela média da imagem consiste no cálculo da imagem média do conjunto de treinamento e essa imagem média é subtraída de cada imagem no conjunto de dados. O resultado é um conjunto de treinamento transformado com média zero e, se o conjunto de treinamento for representativo, o conjunto de dados apresentará média próxima de zero [Lecun et al., 1998].

A Figura 2.4 mostra uma imagem de célula *HEp-2*, processada por alargamento de contraste e equalização de histograma, todas acompanhadas dos respectivos histogramas.

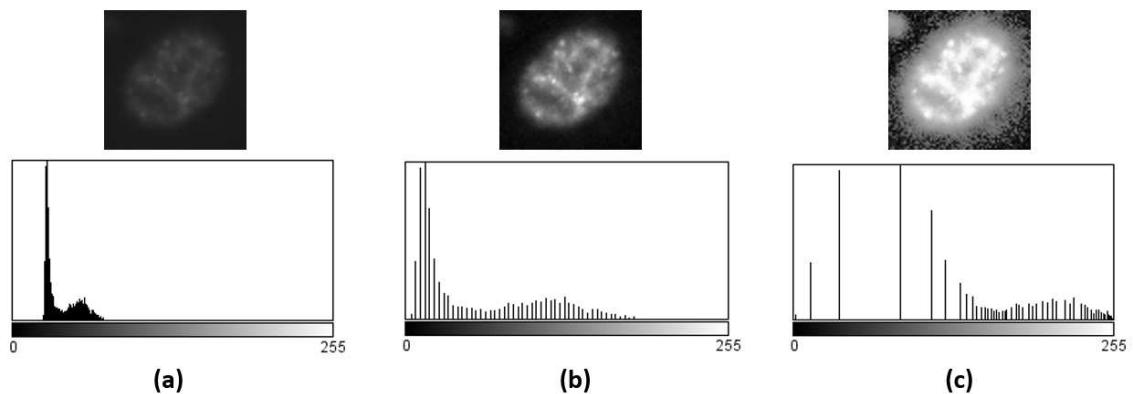


Figura 2.4: Realce de contraste em imagem de célula *HEp-2* por alargamento de contraste e equalização de histograma. A linha superior mostra a imagem e a linha inferior o respectivo histograma. (a) Imagem Original; (b) Alargamento de Contraste; (c) Equalização de Histograma.

### 2.2.2 Aumento do Conjunto de Dados

O aumento do conjunto de dados é uma estratégia utilizada para aumentar artificialmente os dados de treinamento sem a introdução de custos de rotulação dos dados [Krizhevsky et al., 2012]. A geração da base de dados aumentada inclui os seguintes métodos [Chatfield et al., 2014]:

- **Rotações:** a imagem pode ser rotacionada horizontalmente e também verticalmente, caso faça sentido no contexto das imagens de entrada. Gao et al. [2017], por exemplo, utilizaram essa estratégia para aumentar o conjunto de treinamento de células *Hep-2*, rotacionando cada imagem em  $360^\circ$  em torno do seu centro com etapas de  $36^\circ$ ,  $18^\circ$  e  $9^\circ$ .
- **Cortes:** as imagens de entrada, em geral, possuem resolução superior à entrada comumente esperada pelas *CNNs* (por exemplo,  $224 \times 224$  *pixels*), assim invés de redimensionar as imagens para esse tamanho, tem-se como opção cortar a imagem original em diversas subimagens do tamanho de entrada esperado pela *CNN* [Ponti et al., 2017].
- **Flips:** permite inverter as imagens horizontalmente e/ou verticalmente. Essa estratégia faz com que a rede não assuma que determinadas características estão disponíveis apenas em um lado específico [Chatenoud, 2016; Ponti et al., 2017].
- **Adição de Ruídos:** consiste em adicionar ruídos nas imagens de entrada, sendo que utilizar modelos treinados com dados ruidosos pode ser interessante quando a qualidade das imagens de teste estão sujeitas a variações [Nazaré et al., 2018].
- **Criação de novas imagens com PCA:** consiste em criar imagens por meio da Análise de Componentes Principais (PCA) nos canais de cor, como é feito no método Fancy PCA [Krizhevsky et al., 2012].

Nesta dissertação o aumento do conjunto de dados foi feito rotacionando cada imagem do conjunto de treinamento com valores aleatórios entre  $0^\circ$  e  $360^\circ$ . As demais técnicas não foram utilizadas e estão presentes nessa seção pelo caráter informativo.

## 2.3 Redes Neurais

Os principais conceitos abordados no paradigma de Aprendizado Profundo (*Deep Learning*) foram herdados de Redes Neurais, que tem como objetivo desenvolver programas de computador capazes de resolver problemas difíceis de serem solucionados por meio de regras formais. As Redes Neurais começaram a ser introduzidas na Inteligência Artificial na década de 1950 [Rosenblatt, 1962], tendo como inspiração o funcionamento do cérebro humano.

O córtex humano pode ser visto como uma complexa rede, cujos nós são os neurônios. Cada neurônio recebe sinais de outros por meio dos dendritos e conexões são estabelecidas por meio de sinapses. Os sinais de entrada são determinados pelo estímulo recebido e estes são combinados dentro do neurônio gerando um sinal de saída, que é transmitido para outros neurônios por meio do axônio. A amplitude do sinal define o potencial da ação, caso este seja maior do que um valor pré-determinado. Estima-se que o córtex humano tenha 10 bilhões de neurônios e 60 trilhões de sinapses [Kandel et al., 2000]. A Figura 2.5 mostra um diagrama esquemático com as principais partes de um neurônio biológico.

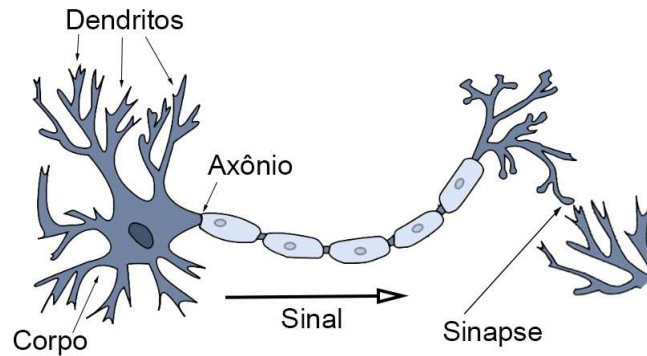


Figura 2.5: Diagrama esquemático de um neurônio biológico.

Uma Rede Neural Artificial (RNA) consiste em um processador massivamente distribuído e paralelo feito de unidades de processamento simples, também conhecidos como neurônios ou nós, que possuem uma capacidade de armazenar e utilizar conhecimento [Russell & Norvig, 2009]. A RNA adquire conhecimento a partir de exemplos por meio de um processo de aprendizagem. Esse processo de aprendizagem ocorre mediante aos ajustes dos parâmetros da rede (pesos sinápticos) em um processo conhecido como treinamento. Dessa forma, o tipo de aprendizado é determinado de acordo com as mudanças ocorridas nos parâmetros [Haykin, 2007].

A modelagem matemática do neurônio artificial é dada de acordo com a Equação 2.3:

$$y_k = f \left[ \left( \sum_{i=1}^n x_i \cdot w_{ki} \right) + b_k \right] \quad (2.3)$$

em que  $y_k$  representa a saída resultante do neurônio  $k$ . O vetor de entrada dado por  $x_i$  e os pesos sinápticos  $w_i$  interagem multiplicativamente, gerando a saída do combinador linear que é somado ao termo *bias* representado por  $b$ . O termo *bias* permite aumentar ou diminuir a entrada da função de ativação  $f$ , que é responsável por restringir a amplitude do sinal de saída ao condicionar a ativação do sinal ao ultrapassar um determinado limiar pelo valor da soma ponderada das entradas [Tan et al., 2005; Russell & Norvig, 2009]. A Figura 2.6 ilustra o modelo matemático do neurônio artificial, conhecido como modelo perceptron.

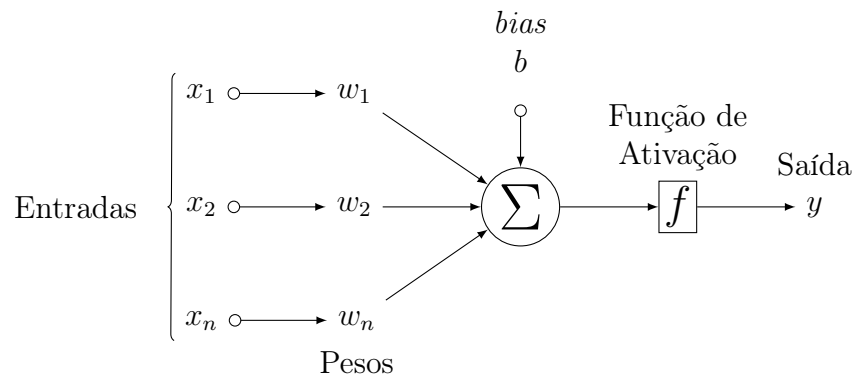


Figura 2.6: Modelo de um perceptron.

A generalização do modelo perceptron é dada pela rede Perceptron de Multicamadas (do inglês *Multilayer Perceptron - MLP*) [McClelland et al., 1986; Lippmann, 1987], que consiste em um algoritmo de aprendizado supervisionado que considera a existência de um conjunto de treinamento contendo objetos previamente classificados e que serão utilizados como protótipos para classificar os demais objetos [Duda et al., 2000]. A rede *MLP* é conhecida por ser *feedforward*, ou seja, é uma rede unidirecional distribuída em camadas, sendo formada por um conjunto de nós que só enviam informações para as camadas posteriores, não permitindo a ocorrência de ciclos ou repetições dentro da rede [Tan et al., 2005; Russell & Norvig, 2009]. Várias camadas intermediárias, denominadas de nós ocultos, estão presentes entre as camadas de entrada e saída de uma rede *MLP*, melhorando a eficácia da rede para extrair características complexas e resolução de problemas não linearmente separáveis [Haykin, 2007]. As funções de ativação permitem definir um componente não linear aos neurônios e ao mesmo tempo restringir o valor das saídas resultantes da combinação linear das entradas e pesos dos neurônios, as principais funções de ativações não lineares que podem ser utilizadas são descritas Seção 2.3.1.

A Figura 2.7 ilustra um exemplo de uma rede *MLP* com 1 camada oculta, 4 atributos ( $x_1$ ,  $x_2$ ,  $x_3$  e  $x_4$ ) e dois estados de saída ( $\hat{y}_1$  ou  $\hat{y}_2$ ).

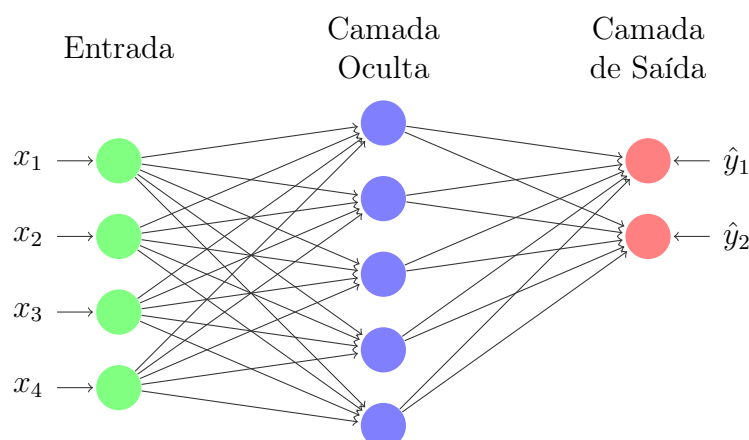


Figura 2.7: Exemplo de Rede MLP.

O mecanismo de aprendizado mais utilizado em uma RNA é o algoritmo de correção de erros, denominado *back-propagation* [Haykin, 2007]. O algoritmo *back-propagation* é caracterizado por realizar a propagação dos sinais e a retropropagação do erro. Durante a propagação, um exemplo de entrada é aplicado aos neurônios de entrada da rede e os sinais produzidos por esses neurônios são propagados para as camadas, até que a rede produza uma saída. Após, realiza-se o cálculo do erro da rede e ele é retro-propagado pela rede camada a camada e os pesos da rede são ajustados para tornar a saída da rede próxima da saída desejada [Haykin, 2007; Negnevitsky, 2011].

O treinamento de uma RNA é iterativo e sua qualidade relaciona-se a escolha da arquitetura da rede e ao número de épocas necessárias para o treinamento a fim de superar os dois desafios principais em aprendizado de máquina: *underfitting* e *overfitting*. *Underfitting* ocorre quando o modelo não é capaz de obter um valor de erro suficientemente baixo no conjunto de treinamento; *overfitting* ocorre quando o intervalo entre o erro de treinamento e o erro de teste é muito grande, ou seja, há pouca generalização [Negnevitsky, 2011; Goodfellow et al., 2016].

### 2.3.1 Tipos de Função Ativação

A função de ativação permite que as redes neurais tenham capacidade representativa ao introduzir um componente de não linearidade. É responsável por decidir se um neurônio deve ou não ser ativado, sendo definida como uma função algébrica [Haykin, 2007]. A seguir é feita uma breve descrição sobre as funções de ativação mais populares [Goodfellow et al., 2016], as quais são ilustradas na Figura 2.8.

- **Identidade ou Linear:** é a função de ativação mais simples e consiste em aplicar a operação de identidade aos dados, sendo os dados de saída proporcionais aos dados de entrada. A ativação do neurônio é exatamente igual à soma ponderada da entrada no neurônio. Em geral, são usadas para neurônios na camada de entrada de uma RNA. É definida pela função  $f(x) = x$ .
- **Sigmóide:** foi bastante utilizada em RNAs pelo fato de gerar resultados semelhantes ao funcionamento dos neurônios biológicos, saídas entre 0 e 1 (ativação e não ativação, respectivamente). Entretanto, é uma função que satura quando seu argumento é muito positivo ou muito negativo, o que a torna muito plana e insensível a pequenas mudanças, causando dificuldades no treinamento. Define-se pela função de ativação  $\sigma(x) = \frac{1}{1+e^x}$ .
- **Tangente Hiperbólica (TanH):** é a função que mais se aproxima da função identidade, sendo uma oportuna alternativa para ativar as camadas ocultas das RNAs invés da função sigmóide. Define-se pela função de ativação  $TanH(x) = 2\sigma(2x) - 1$ .

- **Rectified Linear Unit (ReLU)**: é a função recomendada em redes *feed-forward* devido a sua simplicidade e treinamento mais rápido para a maioria das redes. Resulta em saída 0 se a entrada for menor que 0 e se a entrada for maior que 0, a saída é igual à entrada. Define-se pela função de ativação  $ReLU(x) = \max\{0, x\}$ .

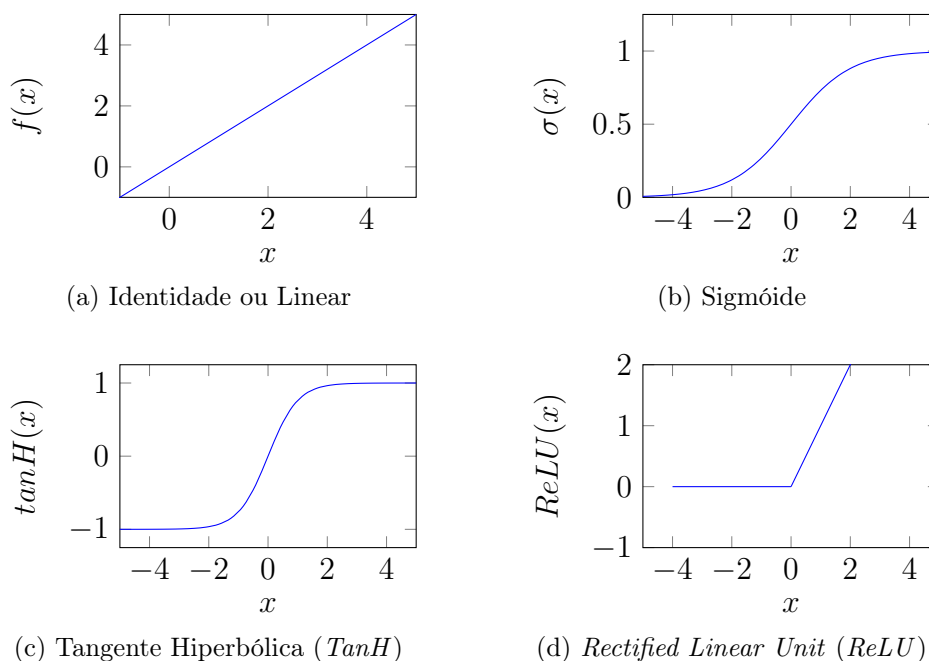


Figura 2.8: Funções de ativação mais populares: (a) Identidade ou Linear; (b) Sigmóide; (c) Tangente Hiperbólica (TanH); e (d) *Rectified Linear Unit (ReLU)*.

### 2.3.2 Estrutura Profunda do Neocórtex

O conceito de hierarquia também está presente em redes neurais profundas e tem como inspiração o funcionamento do córtex humano, pois as ideias e conceitos são organizados hierarquicamente de tal forma que primeiro tem-se a aprendizagem de conceitos simples e, em seguida, estes são compostos para representar conceitos abstratos. O neocórtex humano organiza-se em regiões com um tecido formado por seis camadas de células de neurônios, sendo que as mais profundas, quinta e sexta, apresentam uma maior concentração de neurônios do que as demais camadas. Dessa forma, os neurônios recebem sinais de entrada a partir de outras regiões do córtex e passam as características extraídas para as camadas superiores, que por sua vez, passam a informação para outras regiões do neocórtex [Kandel et al., 2000].

As camadas e regiões superiores também têm conexões de realimentação para as inferiores. Por anos, a maioria dos cientistas ignoravam isso, no entanto, sabe-se hoje que é essencial para que o cérebro seja capaz de realizar a importante função de fazer previsões. As previsões requerem uma comparação entre o que está acontecendo e o que se espera acontecer. Logo, o que está acontecendo vai para a parte

superior da hierarquia e o que espera acontecer vai para a parte inferior [Hawkins & Blakeslee, 2007]. A Figura 2.9 mostra uma representação esquemática da hierarquia das camadas no neocórtex humano.

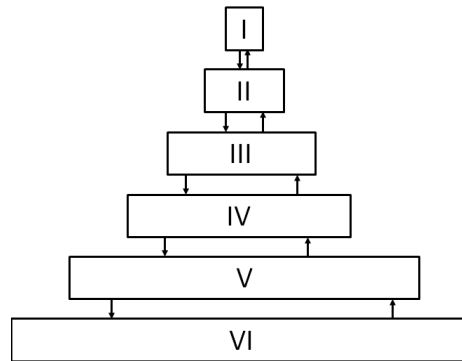


Figura 2.9: Representação das seis camadas do neocórtex humano.

A representação hierárquica com redes neurais foram exploradas nos estudos: Neocognitron de Fukushima [Fukushima, 1988] e rede neural para reconhecimento de dígitos de Lecun [Lecun et al., 1998]. Entretanto, durante o final da década de 2000, os principais esforços dos pesquisadores tiveram foco em outras técnicas como: abordagens baseadas em características invariantes a escala, *Bag-of-Features*, pirâmides espaciais e abordagens baseadas em dicionários [Ponti et al., 2017].

### 2.3.3 Aprendizado Profundo (*Deep Learning*)

O paradigma de Aprendizado Profundo (*Deep Learning*) é uma subárea de aprendizado de máquina que estuda soluções para problemas intuitivos, o que permite que computadores aprendam a partir de experiências anteriores e compreendam o mundo em termos de hierarquias de conceitos. Dessa forma, o computador aprende conceitos mais complexos a partir de conceitos mais simples [Goodfellow et al., 2016; Bengio, 2012].

O foco do aprendizado das arquiteturas que implementam Aprendizado Profundo é identificar abstrações dos dados, dos níveis mais altos para os mais baixos e assim novas representações são obtidas [Bengio, 2012]. Assim, tem-se a composição de camadas de alto nível formada pelas de mais baixo nível, sendo essa uma das principais características da *Deep Learning*.

A disponibilidade de bases de dados com milhões de imagens, modelos complexos e aumento no número de parâmetros foram fatores importantes para que a *Deep Learning* revolucionasse diversas áreas de aprendizado de máquina [Deng et al., 2009; Russakovsky et al., 2015; Ponti et al., 2017].

Outro aspecto interessante é o considerável aumento no poder de processamento paralelo das *GPUs*, o que permite a implementação de modelos profundos com grande quantidade de parâmetros [Goodfellow et al., 2016]. Portanto, a *Deep Learning* apresenta boa performance para descobrir estruturas complexas em dados com alta dimensionalidade [LeCun et al., 2015].

## 2.4 Redes Neurais Convolucionais (CNNs)

As *CNNs* são uma das quatro categorias de métodos de aprendizado profundo, juntamente com as máquinas *Boltzman* (*RBM*s) restritas, *autoencoders* e codificação esparsa. A principal característica de uma *CNN* consiste em ser composta basicamente de camadas convolucionais e tem como principal aplicação, o processamento de informações visuais [Ponti et al., 2017].

A *CNN* é composta basicamente de três camadas neurais: a) camadas convolucionais; b) camadas de *pooling*; e c) camadas totalmente conectadas [Guo et al., 2016]. Essas camadas são estruturadas em arquiteturas hierárquicas, com camadas alternadas com *pooling* e uma totalmente conectada [Lecun et al., 1998]. Os detalhes dessas camadas são descritos a seguir.

### 2.4.1 Camada Convolutiva

As camadas convolucionais executam a convolução dos mapas de características na camada anterior com um conjunto de filtros, no qual cada filtro se destina a detectar um tipo de característica. Sendo  $l$  o índice da camada convolutiva  $C^l$  que é composta por um conjunto de  $K$  filtros,  $W_k^l$  e  $K$  *bias*  $b_k^l$ , para cada  $k \in [1, \dots, K]$ . O tamanho de cada filtro  $W_k^l$  é  $(F \times D)$ , em que  $F$  é o tamanho espacial do filtro e  $D$  é um hiperparâmetro da entrada do volume  $M^{l-1}$ . A entrada do volume,  $M^{l-1}$ , é uma pilha de  $D$  mapas de características resultantes da camada anterior  $l - 1$  e tem o tamanho  $(H \times W \times D)$ . Para gerar o volume de saída  $M^l$ , a camada convolutiva  $C^l$  convoluciona cada parte  $d \in [1, \dots, D]$  do volume de saída  $M^{l-1}$  para cada  $k \in [1, \dots, K]$  filtros  $W_k^l$ . Os resultados das convoluções de cada filtro são somados com o *bias*. O resultado é  $K$  2D mapas de características que são empilhados em um volume de saída  $M^l$ . A Equação 2.4 mostra a formação de uma parte  $M_k^l$ , do volume de saída  $M^l$ .

$$M_k^l = \sum_{d=1}^D M_d^{l-1} * W_d^l + b_k^l \quad (2.4)$$

Após a camada de convolução é aplicada uma função de ativação, por exemplo, uma *REctified Linear Unit* (*ReLU*),  $f(M) = \max(0, M)$  é preferível em relação a outras funções comuns de ativação, como sigmóide e tangente hiperbólica, porque leva a um treinamento muito mais rápido em arquiteturas muito profundas [LeCun et al., 2015]. A Figura 2.10 ilustra a operação na camada de convolução.

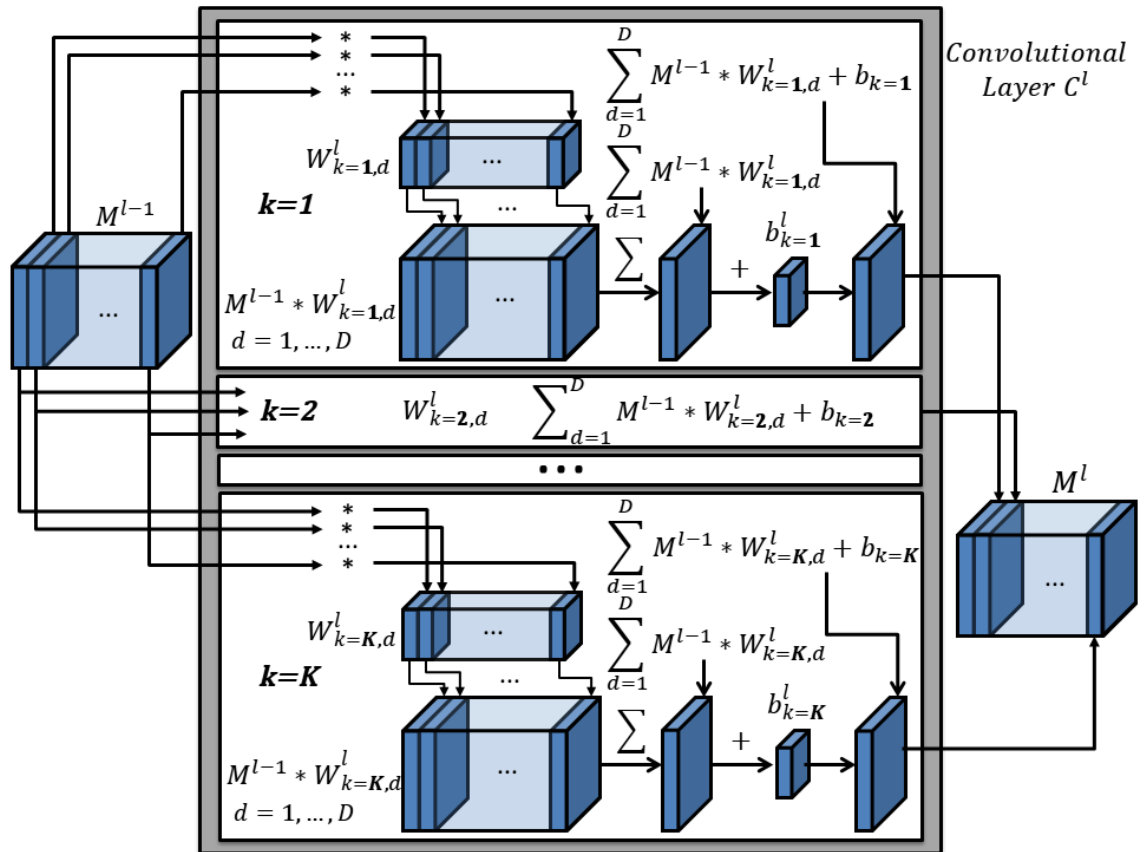


Figura 2.10: Ilustração da estrutura de uma típica camada de convolução.

### 2.4.2 Camada de *Pooling*

As camadas de *pooling* geralmente seguem uma camada de convolução e são responsáveis por reduzir o tamanho dos mapas de características de acordo com alguns critérios, como o *max-pooling* ou *average-pooling*, de acordo com a região de *pooling*. Um conjunto de camadas de *pooling*  $2 \times 2$  reduz o tamanho com  $2 \times 2$  *pixels* para cada um dos *pixels*, escolhendo o *pixel* máximo ou o *pixel* médio para *max-pooling* ou *average-pooling*, respectivamente [Scherer et al., 2010]. O *pooling* máximo resulta em uma convergência mais rápida e uma melhor generalização, na maioria dos casos Scherer et al. [2010]. A Figura 2.11 ilustra o efeito da camada de *pooling* com tamanho  $2 \times 2$  e com critérios máximos e médios, respectivamente.

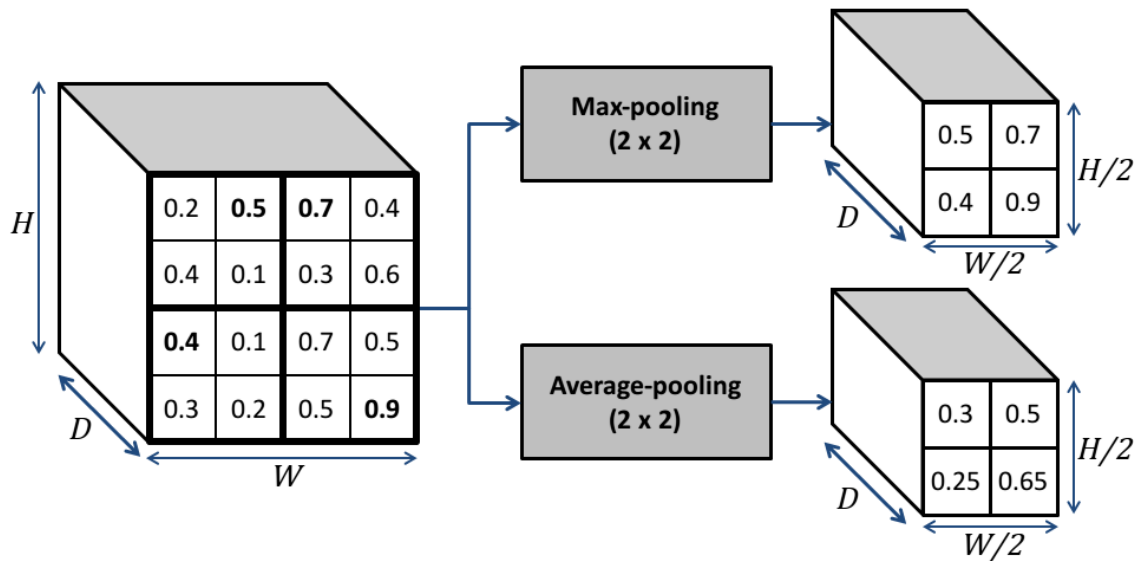


Figura 2.11: Ilustração das computações na camada de *pooling*.

### 2.4.3 Camada Totalmente Conectada

Nas últimas camadas de uma *CNN*, os mapas de características 2D são convertidos em um vetor de características 1D. Este vetor de características é a entrada de uma série de camadas totalmente conectadas, executando a computação de tradicionais produtos internos. A última camada, em geral, consiste em um classificador *softmax*. A camada *softmax*  $h$  tem  $n$  neurônios, em que  $n$  é o número de classes do conjunto de dados, e a saída de cada neurônio  $\tilde{y}_j$  é a probabilidade da imagem de entrada pertencer a classe  $j$  [Gao et al., 2017]. As camadas totalmente conectadas contém cerca de 90% do total de parâmetros (pesos) de uma *CNN* e são responsáveis pela maior parte do custo computacional de treinamento [Guo et al., 2016].

### 2.4.4 Arquiteturas *CNNs*

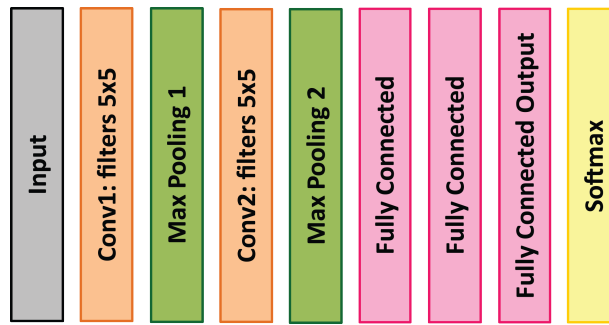
Para este trabalho, cinco arquiteturas *CNN* foram selecionadas para serem comparadas. As *CNNs* descritas a seguir, foram escolhidas tendo como base o desempenho delas em trabalhos de classificação de imagens.

- **LeNet-5:** desenvolvida por Lecun et al. [1998], foi inicialmente projetada para reconhecimento de caracteres e é uma das primeiras aplicações bem sucedidas de *CNNs*. A LeNet-5 é composta por três camadas de convolução, duas camadas de *pooling*, uma camada totalmente conectada e uma camada *softmax*.
- **AlexNet:** arquitetura proposta por Krizhevsky et al. [2012] e venceu a competição ILSVRC 2012 [Deng et al., 2009]. A AlexNet é composta por cinco camadas convolucionais, três camadas de *pooling*, duas camadas totalmente

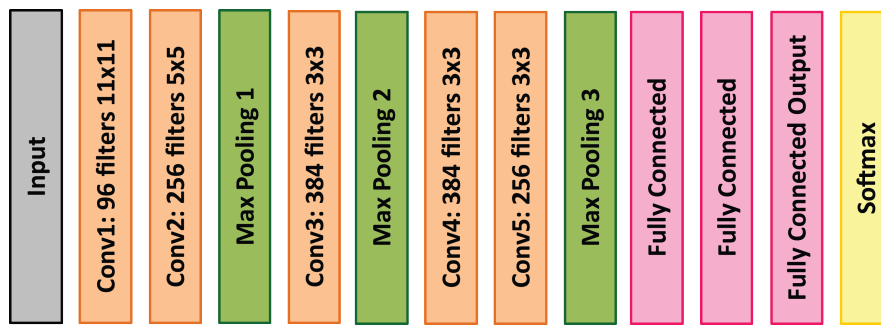
conectadas e uma camada *softmax* [Shin et al., 2016]. Essa arquitetura utiliza regularização *dropout* para reduzir *overfitting* nas camadas totalmente conectadas e *ReLU* como função de ativação nas camadas convolucionais e totalmente conectadas [Srivastava et al., 2014].

- **Inception-V3:** a Inception-V3 é uma versão melhorada da arquitetura GoogLeNet, vencedora da competição ILSVRC-2014 (nas tarefas de classificação e detecção) e foi o primeiro modelo a introduzir a ideia de que as camadas da *CNN* não precisam ser executados sequencialmente. A GoogLeNet é uma instância específica da arquitetura Google Inception. A arquitetura inclui módulos Inception intercalados com camadas de convoluções, *pooling* e totalmente conectada. A GoogLeNet é uma arquitetura muito profunda e apresenta 22 camadas. Os módulos Inception combinam camadas convolucionais com tamanhos  $1 \times 1$ ,  $3 \times 3$  e  $5 \times 5$  com  $3 \times 3$  camadas de *max-pooling*. Como a AlexNet, a GoogLeNet aplica a regularização *dropout* nas camadas totalmente conectadas e as funções de ativação *ReLU* nas camadas de convolução [Szegedy et al., 2015, 2016].
- **VGG-16:** a arquitetura VGG [Simonyan & Zisserman, 2014a] venceu as tarefas de identificação e classificação da competição ILSVRC 2014. Foi a primeira arquitetura a substituir os filtros maiores, ou seja, os filtros que exigem maior custo computacional, por grandes sequências de filtros com tamanhos  $3 \times 3$ . Existem algumas variações do modelo VGG, sendo que as variações VGG-16 e VGG-19 tiveram maior destaque. Nesta dissertação, considerou-se apenas a arquitetura VGG-16, devido à sua boa performance e facilidade de uso.
- **ResNet-50:** a ResNet [He et al., 2016] venceu a competição ILSVRC 2015 e tem como principal característica o uso de uma conexão residual para resolver o problema da degradação ao treinar uma rede muito profunda. É composta por blocos residuais, onde cada um é composto de várias camadas convolucionais empilhadas. A ResNet tem menor perda de convergência sem *overfitting* e a conexão residual pode acelerar a convergência das camadas profundas. Essa arquitetura é cerca de 20 vezes mais profunda que a AlexNet e 8 vezes mais profunda que a VGG [Gu et al., 2018].

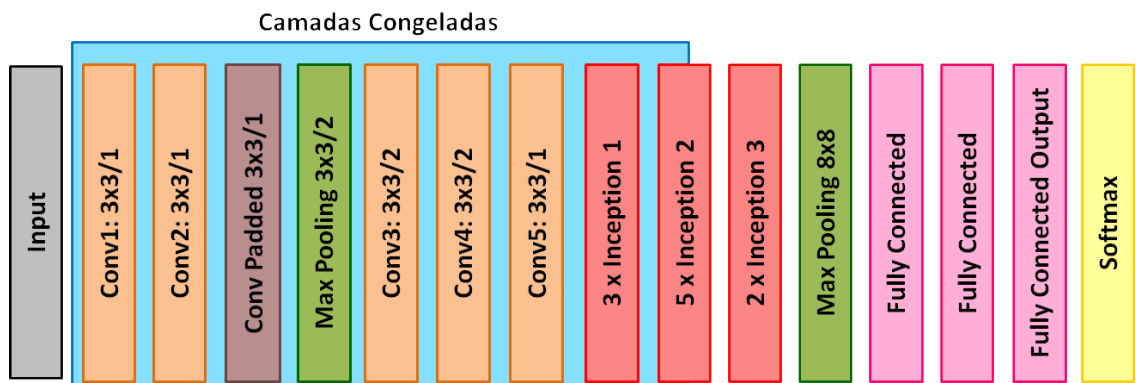
A Figura 2.12 ilustra os principais blocos nos modelos *CNNs* considerados neste trabalho: (a) LeNet-5; (b) AlexNet; (c) Inception-V3; (d) VGG-16; e (e) ResNet-50.



(a) LeNet-5



(b) AlexNet



(c) Inception-V3



(d) VGG-16

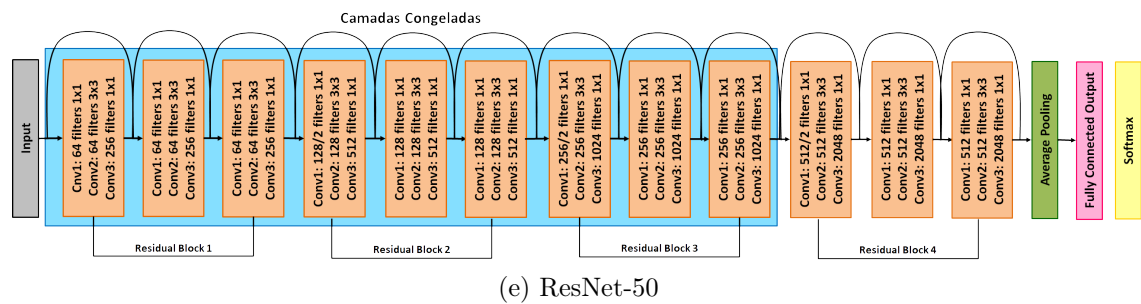


Figura 2.12: Arquiteturas de diferentes *CNNs* avaliadas neste trabalho. Caixas azuis indicam as camadas congeladas nas *CNNs* que aplicou-se *fine-tuning*.

### 2.4.5 Estratégias de Treinamento das *CNNs*

O treinamento de uma *CNN* a partir do zero requer uma quantidade considerável de dados, além de muito poder computacional. Também há situações em que o número de dados rotulados disponíveis é pequeno, portanto, treinar uma nova rede às vezes pode ser uma tarefa desafiadora. Logo, é comum usar uma rede pré-treinada como uma inicialização para o ajuste dos parâmetros. Essas estratégias são descritas a seguir, incluindo suas vantagens e desvantagens, nas próximas subseções.

#### 2.4.5.1 Treinamento Completo

O treinamento completo consiste em treinar a *CNN* a partir do zero, inicializando todos os parâmetros aleatoriamente. Nesse processo, as características são aprendidas diretamente no conjunto de dados em todas as camadas, resultando em um treinamento completo. Essa abordagem tende a ser mais robusta e eficiente, pois permite o controle total da arquitetura e dos parâmetros. Entretanto, requer conjuntos de dados suficientemente grandes para estimar um classificador preciso e exige um alto tempo de treinamento [Bengio, 2009; Goodfellow et al., 2016].

Existem basicamente duas opções de treinamento completo de uma rede profunda. A primeira consiste em projetar e treinar completamente uma nova arquitetura incluindo, por exemplo: o número de camadas, neurônios, funções de ativações, número de épocas, taxa de aprendizado e taxa de decaimento. A outra opção é usar uma arquitetura existente sem modificar sua estrutura e hiperparâmetros, de forma a treinar completamente os pesos para um determinado conjunto de dados.

Espera-se que o treinamento completo alcance melhores resultados quando comparado com outras estratégias, pois a *CNN* aprende características específicas para o conjunto de dados de interesse.

#### 2.4.5.2 *Fine-Tuning*

A estratégia de treinamento baseada em *fine-tuning* consiste em uma abordagem comum e altamente eficaz em aprendizado profundo [Goodfellow et al., 2016], em que a rede é previamente treinada para uma tarefa de classificação usando um

conjunto de dados muito grande, como o ImageNet [Deng et al., 2009] ou CIFAR-10 [Krizhevsky & Hinton, 2009]. Os valores dos parâmetros (pesos) aprendidos para as camadas iniciais da rede são mantidos e as camadas superiores são treinadas sobre o conjunto de interesse, que se destinam a aprender as estruturas mais complexas dos dados.

Particularmente, neste trabalho, aplicou-se *fine-tuning* em três modelos de *CNNs*: Inception-V3, VGG-16 e ResNet-50. Depois disso, as camadas iniciais dos modelos foram congeladas para manter os parâmetros aprendidos durante o treinamento sobre o conjunto de dados do ImageNet, enquanto as camadas superiores têm seus parâmetros ajustados usando uma taxa de aprendizado muito pequena. Foram congeladas para a Inception-V3 as 197 primeiras camadas, para a VGG-16 os três blocos iniciais de camadas convolucionais e para a ResNet-50 todas as camadas iniciais, exceto a última camada do bloco residual. A Figura 2.12 mostra as camadas de treinamento congeladas e livres para cada modelo *CNN*.

## 2.5 Otimização de Parâmetros

Definir empiricamente um nível de desempenho “satisfatório” para uma *CNN* é subjetivo e requer conhecimento prévio sobre o algoritmo e o cenário da aplicação. Alternativamente, é possível representar o valor de escolha dos parâmetros como um problema de otimização, em que estes são considerados como variáveis de decisão e o erro de validação do modelo treinado é a função objetivo a ser minimizada.

Um conjunto de algoritmos de otimização pode ser empregado para executar o treinamento de modelos de *CNNs*, ou seja, para ajustar os parâmetros de conexão  $W$  para minimizar a função que computa a qualidade da predição realizada, denominada função de custo (*loss function*). Dentre as funções de custo mais utilizadas em aprendizado profundo, tem-se a função de entropia cruzada (*cross-entropy*), a qual é considerada neste trabalho [Goodfellow et al., 2016; Ponti et al., 2017].

A função de entropia cruzada é computada sobre um conjunto de amostras de treinamento  $x_j$ , baseada na classificação feita considerando os parâmetros  $W$ ,  $f(x_j, W)$ , os rótulos reais e a classificação correta  $y_j$ , como mostrado na Equação 2.5:

$$\mathcal{L}(W) = \frac{1}{n} \sum_{j=1}^N \ell(y_j, f(x_j; W)) \quad (2.5)$$

A minimização de  $\mathcal{L}(W)$  é feita por meio de algoritmos de otimização, como o Gradiente Descendente (GD) [Cauchy, 1847], que consiste em calcular derivadas parciais no espaço de parâmetros a fim de encontrar o conjunto de valores de parâmetros  $W$  que minimiza a função de custo [Ponti et al., 2017].

Entretanto, usar o GD com *back-propagation* é inviável em aplicações de aprendizado profundo, devido à necessidade de calcular o gradiente de cada instância. Para superar esse problema, são propostas variações do GD como o Gradiente Descendente Estocástico (do inglês *Stochastic Gradient Descent - SGD*) [Lecun et al.,

1998], AdaGrad [Duchi et al., 2011], Adam [Kingma & Ba, 2014], RMSprop [Tieleman & Hinton, 2012], entre outros.

O *SGD* é um dos algoritmos mais populares para otimização de parâmetros em modelos de aprendizado profundo [Goodfellow et al., 2016]. Em vez de calcular o gradiente para cada amostra, o *SGD* é baseado aproximações do GD baseada em lotes de amostras selecionadas aleatoriamente. Este trabalho usa *SGD* com *momentum* (tradicional e Nesterov) [Lecun et al., 1998; Goodfellow et al., 2016].

## 2.6 Otimização de Hiperparâmetros

Os hiperparâmetros são variáveis cujos valores são atribuídos antes da etapa de treinamento. Eles definem a topologia da rede, os pesos de suas conexões, a arquitetura, os detalhes de seus algoritmos de treinamento e otimização [Bengio, 2012]. A escolha de valores para os hiperparâmetros é crucial para o desempenho da *CNN* e pode ser feita empiricamente, testando incrementalmente diferentes valores e realizando ajustes, até que o algoritmo apresente um desempenho “satisfatório”.

A seguir, são descritos os hiperparâmetros ajustados para otimizar os resultados de cada uma das *CNNs* comparadas neste trabalho:

- **Número de unidades ocultas ( $n_h$ ):** o número de neurônios ocultos na camada totalmente conectada é um hiperparâmetro relevante para o modelo da *CNN*. Por um lado, um pequeno valor  $n_h$  pode levar o treinamento a *underfitting*. Por outro lado, um  $n_h$  alto pode gerar *overfitting* e aumentar a necessidade de tempo de processamento [Goodfellow et al., 2016]. Quanto maior e complexo o conjunto de dados, maior o número de neurônios necessários para obter informações relevantes [Bengio, 2012].
- **Decaimento de pesos (*Weight Decay*):** consiste em uma penalidade imposta sobre valores de peso que não influenciaram significativamente a classificação correta durante o treinamento. A ideia é tornar esses valores próximos de zero, enquanto outras conexões relevantes mantêm seus valores relativamente altos. Isso influencia diretamente a regularização do coeficiente  $\lambda$  que é responsável por controlar a suavização aplicada à rede e reduzir o *overfitting* [Bengio, 2012].
- **Taxa de Aprendizado (*Learning Rate*):** a taxa de aprendizado define o nível de ajustes de conexões de pesos e topologia da rede, aplicados em cada ciclo de treinamento. Uma pequena taxa de aprendizado permite uma sintonia cirúrgica do modelo com os dados de treinamento, ao custo de maior número de ciclos de treinamento e tempo de processamento. Uma alta taxa de aprendizado permite que o modelo aprenda mais rapidamente, mas pode sacrificar sua acurácia por causa da falta de precisão sobre os ajustes. Este valor é geralmente definido em 0.01, mas em alguns casos é interessante ser otimizado, principalmente quando pretende-se melhorar o tempo de execução ao usar o *SGD* [Bengio, 2012].

- **Momentum:** o coeficiente *momentum* é responsável por remover alguns dos ruídos e oscilações que podem existir nas direções de alta curvatura da função de perda obtidas pelo *SGD*. Em geral, seu valor padrão é  $\mu = 1$ , mas em algumas situações a otimização desse hiperparâmetro pode levar a melhores resultados [Polyak, 1964].
- **Momentum Nesterov:** difere do *momentum* tradicional, pois impõe um fator de correção sobre a velocidade do gradiente atual quando esta é avaliada, o que pode resultar, durante o aprendizado, em uma convergência para o ótimo significativamente mais rápida [Sutskever et al., 2013; Goodfellow et al., 2016].

### 2.6.1 Tree of Parzen Estimators (TPE)

O algoritmo *Tree of Parzen Estimators (TPE)* [Bergstra et al., 2011] é uma abordagem Sequencial de Otimização Baseada em Modelo (*Sequential Model-Based Global Optimization - SMBO*). Algoritmos *SMBO* otimizam um conjunto de hiperparâmetros  $x$  para algum modelo baseado em seu histórico de observações [Hutter et al., 2011]. Eles são adequados para avaliações dispendiosas da função de custo, porque iteram entre os modelos de treinamento para escolher quais configurações investigar [Bergstra et al., 2011].

O *TPE* modela  $p(x|y)$  substituindo as distribuições da configuração anterior por densidades não paramétricas, ou seja, uma distribuição prévia dos hiperparâmetros é realizada e nas primeiras iterações é feita uma pesquisa aleatória. Posteriormente, as observações obtidas na pesquisa aleatória são categorizadas em dois grupos: i) grupo com as observações que deram os melhores resultados após a avaliação; e ii) todas as outras observações. O objetivo é encontrar um conjunto de hiperparâmetros com maior probabilidade de estar no primeiro grupo. A definição de  $p(x|y)$  é dada de acordo com a Equação 2.6 [Bergstra et al., 2011].

$$p(x|y) = \begin{cases} \ell(x), & \text{se } y < y^* \\ g(x), & \text{se } y \geq y^* \end{cases} \quad (2.6)$$

em que  $\ell(x)$  é a densidade das observações que geram os melhores resultados e  $g(x)$  corresponde as observações restantes;  $y^*$  é escolhido para ser um quantil  $\gamma$  dos valores observados de modo que  $p(y < y^*) = \gamma$ , mas nenhum modelo específico para  $p(y)$  é necessário; os hiperparâmetros são representados por  $x$  e  $y$  representa o índice de qualidade associado.

A parametrização de  $p(x, y)$  como  $p(y)p(x|y)$  permite que o algoritmo *TPE* otimize com facilidade o critério *Expected Improvement (EI)* [Jones, 2001], definido pela Equação 2.7:

$$EI(x) = \frac{\ell(x)}{g(x)} \quad (2.7)$$

sendo  $\ell(x)$  a probabilidade do grupo com as observações que deram os melhores resultados após a avaliação e  $g(x)$  corresponde a todas as demais observações. A cada iteração o algoritmo retorna o candidato  $x^*$  com o maior *EI*.

## 2.7 Avaliação do Desempenho das CNNs

A avaliação do desempenho das CNNs é um importante passo em tarefas de reconhecimento de padrões. Por meio de métodos de medição de erro é possível obter valores que indicam a performance obtida por cada CNN ao classificar o conjunto de dados de interesse. Nesta dissertação é considerado o método de validação cruzada com  $k$ -fold e matriz de confusão para analisar e visualizar a performance do reconhecimento.

### 2.7.1 Validação Cruzada com $k$ -fold

O método estatístico de validação cruzada em  $k$ -partes (do inglês *k-fold cross-validation*) [Kohavi, 1995], foi utilizado para particionar os conjuntos de treinamento, validação e teste. Na técnica *k-fold*, a amostra original é dividida aleatoriamente em  $k$  sub-amostras de tamanhos iguais. Para cada uma das  $k$  sub-amostras, uma sub-amostra é usada como validação para testar o modelo, e os  $k - 1$  restantes são usados como dados de treinamento. O processo de validação cruzada é então repetido  $k$  vezes, com cada uma das  $k$  sub-amostras que são usadas uma vez como dados de validação. As múltiplas  $k$  execuções feitas pelo processo de validação cruzada permitem a redução da variabilidade do resultado do estimador, ou seja, o resultado se aproxima mais de um resultado obtido em um cenário real. Por isso, esta técnica exerce mais confiança do que uma técnica de *hold-out* simples [Devijver & Kittler, 1982], onde o treinamento é feito uma única vez sobre uma divisão dos dados.

A estratégia de validação cruzada é mais robusta para *outliers* e eventual *overfitting*, gerando resultados mais confiáveis. Provavelmente, devido ao grande tempo necessário para realizar a validação cruzada, os estudos com CNNs publicados anteriormente não consideram essa estratégia, mas simplesmente dividem o conjunto de dados em grupos de treinamento, validação e teste, como feito no trabalho apresentado por Gao et al. [2017].

A Figura 2.13 ilustra de forma generalizada o método *k-fold cross-validation* em que são realizados  $k$  experimentos e em cada experimento uma sub-amostra (em vermelho) é usada uma vez como validação.

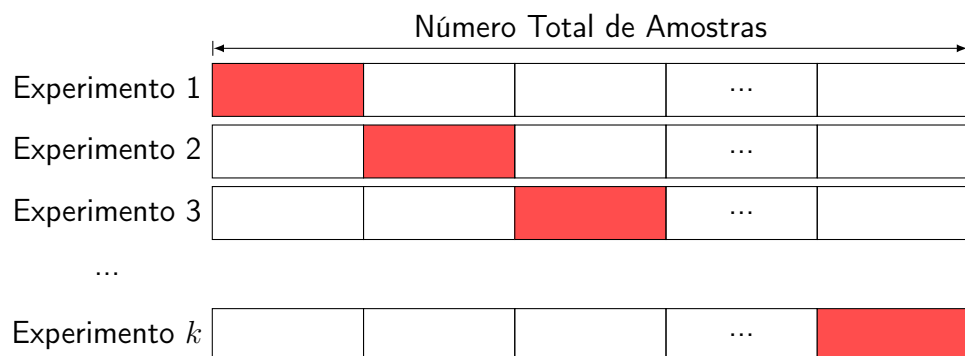


Figura 2.13: Visão geral do método  $k$ -fold.

### 2.7.2 Matriz de Confusão

A matriz de confusão, também conhecida como tabela de contingência, permite verificar o número de classificações corretas em oposição às classificações preditas para cada classe e simplifica o processo de verificar se o algoritmo de aprendizagem está confundindo as classes. Baseia-se no número de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos [Duda et al., 2000].

Considera-se verdadeiro positivo se o elemento de entrada é positivo e o classificador o considera positivo; o falso positivo ocorre quando o elemento de entrada é negativo e o classificador o define como verdadeiro; o falso negativo consiste em se ter como entrada um elemento positivo e para o classificador ele é negativo; considera-se verdadeiro negativo se o elemento de entrada é negativo e o classificador o considera negativo [Costa & Cesar-Jr., 2009].

A Figura 2.14 apresenta uma visão geral de uma matriz de confusão, sendo composta por linhas e colunas que representam, respectivamente, as instâncias da classe real e as instâncias da classe predita.

		Classe Predita	
		Positivo	Negativo
Classe Real	Positivo	Verdadeiro Positivo	Falso Negativo
	Negativo	Verdadeiro Negativo	Falso Positivo

Figura 2.14: Visão geral de uma matriz de confusão.

## 2.8 Trabalhos Correlatos

Uma das primeiras propostas de desenvolvimento de um sistema para classificação de células *HEp-2* foi feita por Perner et al. [2002]. Os autores extraíram uma série de características de textura de doze versões quantizadas das imagens originais e classificaram com Árvore de Decisão. A mesma abordagem foi adotada por Sack et al. [2003] a fim de identificar fluorescência positiva em um conjunto de padrões de imunofluorescência. No entanto, poucas imagens estavam disponíveis para treinamento do modelo baseado em Árvore de Decisão, que era simbólico, mas impreciso se comparado às mais recentes abordagens de aprendizado profundo.

Vários estudos foram propostos com o objetivo de classificar células *HEp-2* com extração de características feitas à mão, ou seja, usando um processo baseado em usuário e não automatizado. Por exemplo, Nosaka & Fukui [2014] propuseram um método baseado em co-ocorrência invariante de rotação entre características de padrões binários locais adjacentes (*RIC-LBP*) e máquinas de vetor e suporte (*Support Vector Machines (SVM)*) para classificar imagens de células *HEp-2*. Em seguida, Stoklasa et al. [2014] fizeram a classificação de células *HEp-2* usando um classificador *K-NN* com descritores de Haralick, padrões binários locais, *SIFT*, descrição de superfícies e um descritor baseado em granularidade.

Shen et al. [2014] apresentaram um modelo *Bag of Visual Words (BoVW)* de características baseados em agrupamento de ordem de intensidade e classificação com *SVM*. Gragnaniello et al. [2016] utilizaram *Scale-Invariant Descriptor (SID)*, que consiste em um descritor local que garante invariância de escala e rotação. As características extraídas foram codificadas com *BoVW* e a classificação foi feita com *SVM*.

Manivannan et al. [2016] normalizaram as imagens de células *HEp-2* e extraíram quatro tipos de características locais utilizando: *Multiresolution Local High-order Statistical, root-SIFT* (uma variante do descritor *Scale-Invariant Feature Transform - SIFT*), projeções aleatórias de vetores de intensidade e histogramas de intensidade. As características extraídas foram codificadas com codificação linear local, que consiste em uma variante eficiente da codificação esparsa e o conjunto máximo é usado para agregar os códigos lineares locais. Além disso, para cada tipo de característica, uma pirâmide de células de dois níveis foi utilizada para capturar a estrutura espacial e a classificação foi feita com *SVM*.

Kastaniotis et al. [2017] propuseram um método denominado *Vector of Hierarchically Aggregated Residuals (VHAR)*, para agregar hierarquicamente o resíduo da representação esparsa dos vetores de características derivados do descritor *SIFT*. Usaram diferentes classificadores, como: *SVM* e *KNN*. Ensafi et al. [2017] utilizaram descritores *SIFT* e *SURF* com um modelo *BoVW* e classificou as imagens de células *HEp-2* usando *SVM*. Ainda assim, a extração de características não automatizada pode ser uma tarefa onerosa, especialmente para grandes conjuntos de dados.

Com o aumento dos recursos computacionais (por exemplo: computação massivamente paralela e distribuída), técnicas aprimoradas de otimização e conjunto de dados crescentes, os modelos *CNNs* superaram significativamente os modelos baseados em extração manual de características para tarefas gerais de reconhecimento

visual. Um dos primeiros trabalhos foi apresentado por Foggia et al. [2014], em que Malon et al. adotaram uma *CNN* para classificar imagens de células *HEp-2*, o alargamento de contraste foi aplicado sobre as imagens e as mesmas foram redimensionadas para  $100 \times 100$  *pixels*, considerando apenas o componente verde.

Bayramoglu et al. [2015] e Phan et al. [2016] classificaram as imagens de células *HEp-2* do conjunto de dados da *International Conference on Pattern Recognition (ICPR) 2012*. Bayramoglu et al. [2015] consideraram uma arquitetura *CNN* inspirada na AlexNet [Krizhevsky et al., 2012] e aumentaram do conjunto de dados por meio de rotações nas imagens, operações de *flipping*, escala e transformação de contraste. Phan et al. [2016] propuseram um modelo *CNN* pré-treinado, selecionaram um conjunto de características baseado no algoritmo de Máxima Relevância e Mínima Redundância (mRMR), para reduzir a dimensionalidade dos dados de entrada com características extraídas pela *CNN* e realizaram classificação com *SMV*.

Entretanto, o aprendizado por transferência aplicado nos trabalhos anteriormente citados concentram-se principalmente no problema de classificação de um pequeno conjunto de dados de células *HEp-2*. Diferentemente, essa dissertação utiliza outros métodos de pré-processamento e outras arquiteturas *CNN* e alcança melhores resultados.

Recentemente, Gao et al. [2017] propuseram um método para classificação automática de células *HEp-2* com uma *CNN* que compartilha a arquitetura básica da LeNet-5 [Lecun et al., 1998]. Diferente dos trabalhos citados anteriormente, a *CNN* foi capaz de aprender características automaticamente durante o processo de treinamento. Experimentos considerando uma instância de cada imagem alcançaram uma acurácia de 88.58%, melhorado para 96.76% com o aumento do conjunto de dados com cópias rotacionadas de suas imagens. Esses experimentos utilizaram uma técnica de validação *hold-out* [Devijver & Kittler, 1982], que consiste em atribuir objetos aleatoriamente aos conjuntos de treinamento e teste (veja a Seção 2.7.1). Essa abordagem pode criar conjuntos de treinamento e validação tendenciosos, o que pode resultar em modelos *CNN* anormais, ou seja, *CNNs* que não se comportam como a maioria das *CNNs* treinadas com o mesmo tipo de dados, se outras distribuições aleatórias forem consideradas.

Seguindo a metodologia proposta por Gao et al. [2017], inicialmente, o trabalho descrito nesta dissertação considerou os modelos AlexNet e GoogLeNet para classificação de imagens de células *HEp-2*, além do modelo LeNet-5. Também considerou-se ajustar alguns hiperparâmetros das *CNNs* empiricamente. Os resultados experimentais permitiram concluir que as estratégias de pré-processamento não melhoraram significativamente os valores de acurácia das *CNNs* quando treinadas com valores fixos (pré-definidos por padrão) de seus hiperparâmetros. Além disso, quando o aumento de dados foi considerado, o alargamento de contraste seguido pela centralização dos dados foi significativo para obter bons resultados. Entretanto, ao seguir o método de validação adotado por Gao et al. [2017], verificou-se que conjuntos tendenciosos podem ser criados aleatoriamente, fazendo com que as *CNNs* se ajustem a amostras não representativas (anormais) e resultem em inesperadas acurácias.

Assim, outras técnicas de validação devem ser devidamente consideradas. Por exemplo, técnica de validação cruzada com *k-fold* ou *leave-one-out* [Devijver & Kit-

tlar, 1982]. Tais técnicas requerem maior poder computacional ou tempo de processamento, mas oferecem maior confiabilidade. Além disso, os valores dos hiperparâmetros também influenciam nos resultados das *CNNs* e devem ser ajustados para cada tipo de dado e aplicação.

Sendo assim, o trabalho resultante descrito nesta dissertação adotou novas técnicas de validação com *k-fold*, otimização de hiperparâmetros utilizando *Tree of Parzen Estimators (TPE)*, estudou a influência de diferentes abordagens de pré-processamento em cada uma das arquiteturas *CNNs* e avaliou e analisou as estratégias de treinamento a partir do zero e com *fine-tuning*.

Mais recentemente, Lei et al. [2018] propuseram um método baseado em aprendizado profundo supervisionado com aprendizagem de transferência intermodal para classificar imagens de células *HEp-2* usando a arquitetura *CNN* ResNet-50. Os autores argumentam que, para uma *CNN* tão profunda, o número de imagens médicas rotuladas disponíveis para eles ainda era pequeno, fazendo com que o problema de *overfitting* ocorresse. Assim, eles pré-treinaram a rede usando o conjunto de dados da *ICPR 2012* para modelá-la na classificação de células *HEp-2* e aplicaram a *CNN* resultante sobre o conjunto de dados ICPR 2016-Task1. A ideia era que o aprendizado de transferência intermodal com ResNet-50 poderia ser bem sucedido por causa que as características expressivas no baixo e alto nível de representação são semelhantes. A técnica proposta apresentou resultados semelhantes ou superiores para a maioria das técnicas experimentalmente comparadas. No entanto, técnicas intermodais exigem similaridade significativa entre conjuntos de dados, o que pode não ser possível para o cenário de aplicação. Por outro lado, as técnicas apresentadas nesta dissertação não requerem múltiplos conjuntos de dados similares e foram consideradas, além da arquitetura LeNet-5, AlexNet [Krizhevsky et al., 2012], Inception-V3 (GoogleNet) [Szegedy et al., 2015, 2016], VGG-16 [Simonyan & Zisserman, 2014b] e ResNet-50 [He et al., 2016]; e explorou seis diferentes estratégias de pré-processamento.

# Capítulo 3

## Material e Métodos

Neste capítulo são descritos os métodos propostos, implementados e testados para classificar células *HEp-2* em imagens de imunofluorescência, tendo como objetivo principal avaliar o desempenho de diferentes *CNNs*.

Especificamente, avaliou-se a influência de diferentes estratégias de pré-processamento, aumento de dados, treinamento completo e *fine tuning* sobre a performance das *CNNs*. Para isso, um extenso estudo foi feito por meio de técnicas de validação cruzada *k-fold* (que oferece maior confiabilidade aos resultados) e otimização dos valores dos hiperparâmetros. A Figura 3.1 mostra as etapas do método proposto.

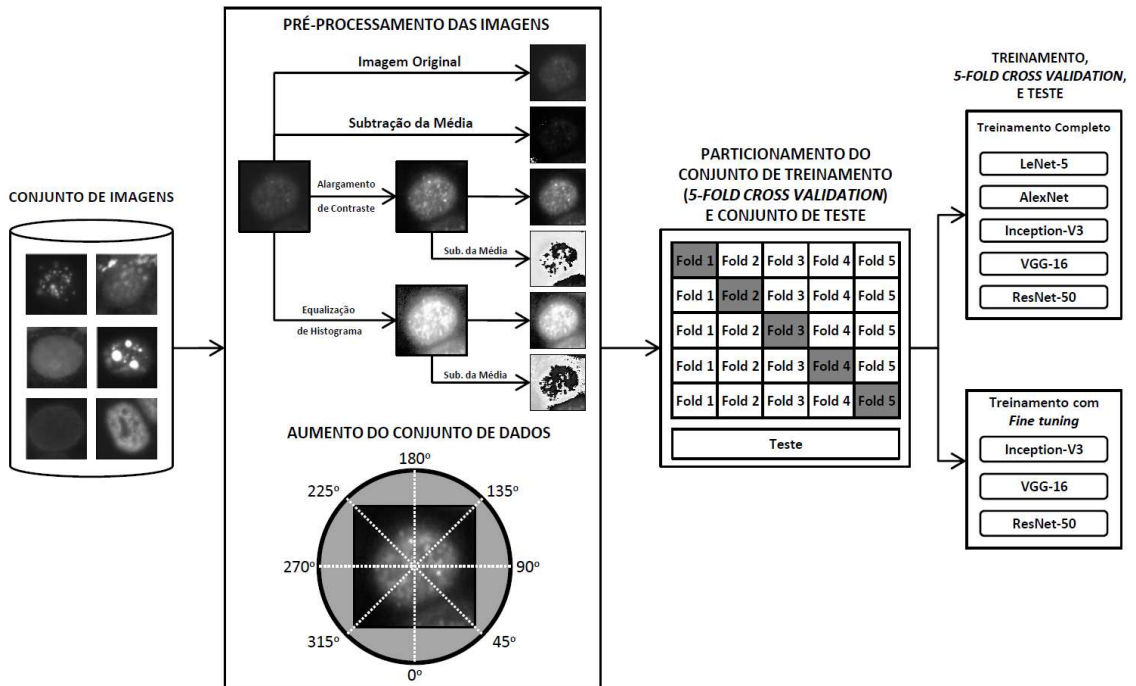


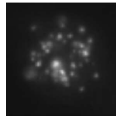
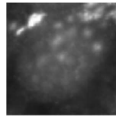
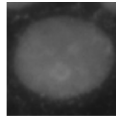
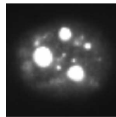
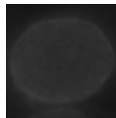
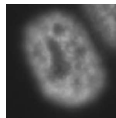
Figura 3.1: Etapas do método proposto.

Experimentos preliminares apresentados em “*Exploiting Convolutional Neural Networks and preprocessing techniques for HEP-2 cell classification in immunofluorescence images*” [Rodrigues et al., 2017] foram desenvolvidos utilizando validação por *hold-out*. Posteriormente, esses experimentos foram refeitos utilizando validação cruzada com *k-fold*. Portanto, a versão preliminar não será considerada na versão final desta dissertação, apenas os experimentos com maior confiança estatística.

### 3.1 Conjunto de Imagens

O conjunto de imagens utilizado foi disponibilizado para competição “*Contest on Performance Evaluation on Indirect Immunofluorescence Image Analysis Systems*” hospedado no ICPR 2014 <sup>1</sup> [Hobson et al., 2014] e contém 13.596 imagens de células *HEp-2* categorizadas em seis classes (descritas na Seção 2.1.1). A Tabela 3.1 traz informações sobre a quantidade de imagens e exemplos de instâncias para cada classe.

Tabela 3.1: Informações sobre cada classe do conjunto de dados ICPR 2014.

Classe	Quantidade	Exemplo
<i>Centromere</i>	2.741	
<i>Golgi</i>	724	
<i>Homogeneous</i>	2.494	
<i>Nucleolar</i>	2.598	
<i>Nuclear Membrane</i>	2.208	
<i>Speckled</i>	2.831	
<b>Total</b>	<b>13.596</b>	

<sup>1</sup>Disponível em: <http://nerone.diem.unisa.it/hep2-benchmarking/dbtools/>

## 3.2 Particionamento do Conjunto de Imagens

As 13.596 imagens do conjunto de dados foram embaralhadas aleatoriamente e particionadas em seis conjuntos estratificados, ou seja, as classes são representadas em cada conjunto na mesma proporção dos dados originais. Aplicou-se a estratégia de validação cruzada, em que cinco conjuntos estratificados (*folds*) foram usados para treinamento e um para teste, como ilustrado na Figura 3.2. O processo de validação é repetido  $k$  vezes, em que cada uma delas um dos *folds* de treinamento é escolhido para a validação do treinamento.

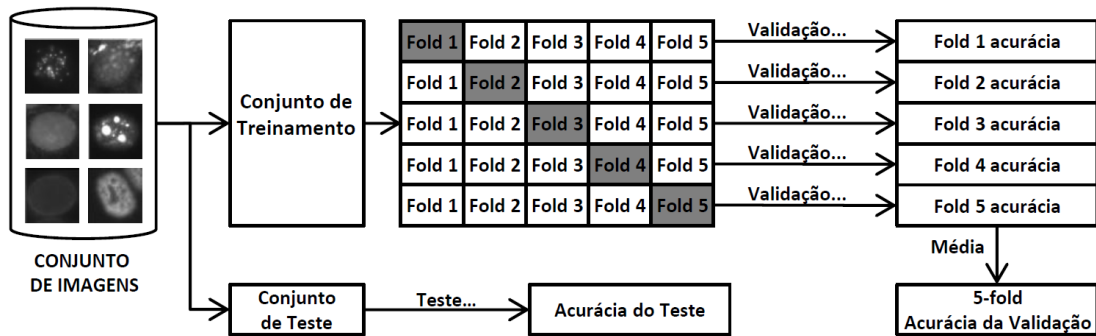


Figura 3.2: Representação gráfica da validação cruzada com  $k$ -fold.

## 3.3 Pré-processamento das Imagens

Todas as imagens foram redimensionadas para  $224 \times 224$  pixels, que é o menor tamanho permitido como entrada dos modelos *CNN* considerados neste trabalho. O redimensionamento é feito considerando a interpolação bilinear, que consiste em utilizar os quatro vizinhos mais próximos para estimar a intensidade de uma dada posição [Gonzalez & Woods, 2007]. Isso é feito, antes de aplicar os métodos de pré-processamento e essas imagens são usadas para treinar e testar todos os modelos *CNN*.

A combinação das técnicas de pré-processamento descritas na Seção 2.2.1 resultaram em seis estratégias de pré-processamento, baseadas em combinações de melhoria de contraste e normalização de imagem, ilustradas na Figura 3.3. A imagem mais à esquerda que é original do conjunto de dados, é definida como estratégia (a) (ou seja, sem pré-processamento). Depois disso, em (b), há um exemplo da estratégia de subtração da imagem média, a partir da média de todas as imagens do conjunto de treinamento. A estratégia (c) é o alargamento de contraste e (d) é o alargamento de contraste com subtração da imagem média. Por último, as estratégias (e) e (f) são equalizações de histograma sem e com subtração da média, respectivamente.

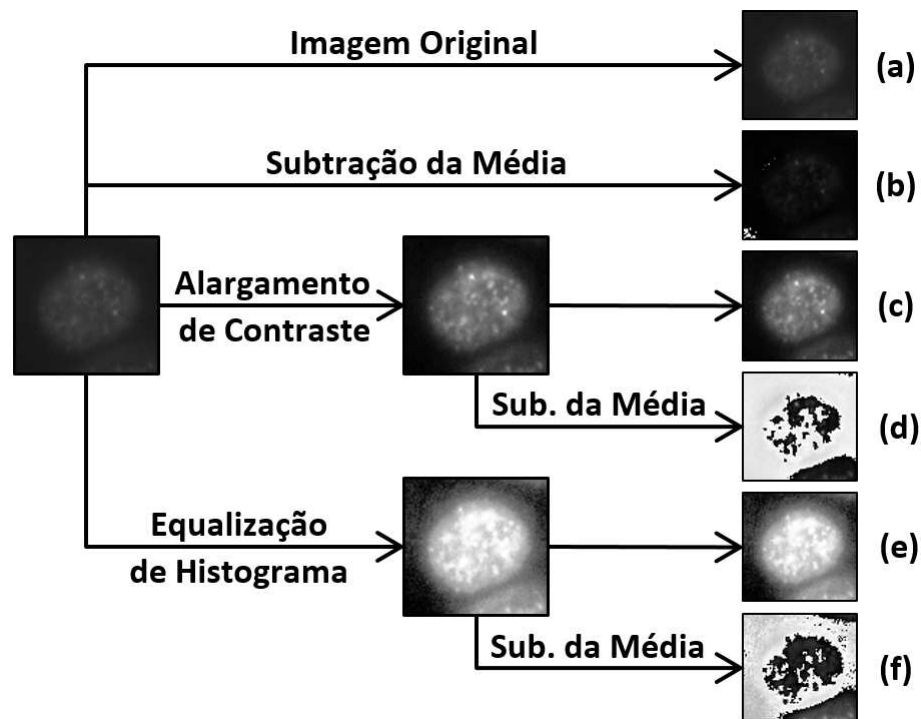


Figura 3.3: Melhoria de contraste na imagem de célula *HEp-2*: (a) imagem original, (b) subtração da média, (c) alargamento de contraste, (d) alargamento de contraste com subtração da média, (e) equalização de histograma e, (f) equalização de histograma com subtração da média.

Os conjuntos de dados gerados a partir da aplicação das estratégias de pré-processamento foram submetidos ao procedimento de aumento de dados com base nas rotações incrementais das imagens com valores aleatórios entre  $0^\circ$  e  $360^\circ$ , conforme descrito na Seção 2.2.2. A investigação feita considerando o aumento do conjunto de treinamento é detalhada na Seção 4.2.

### 3.4 Estimativa de Hiperparâmetros

Para encontrar os valores otimizados para um conjunto de importantes hiperparâmetros, utilizamos o algoritmo *Tree of Parzen Estimators (TPE)*, conforme descrito na Seção 2.6.1. O resultado da pesquisa no espaço de busca pelos melhores hiperparâmetros é mostrada na Tabela 3.2. Os melhores valores para cada hiperparâmetro retornado pelo algoritmo *TPE* são apresentados na Tabela 3.3, para cada arquitetura *CNN* após 30 épocas e 5 avaliações no espaço de busca. Esses valores são usados em todos os experimentos feitos para treinar todos os modelos *CNN* a partir do zero.

Tabela 3.2: Espaço de pesquisa de hiperparâmetro considerado para otimização.

Hiperparâmetro	Distribuição	Valor
Nº. Unid. Ocultas $n_h$	Catagórico	$x \in \{256, 512, 1024, 2048, 4096\}$
<i>Decay</i>	Catagórico	$x \in \{1e^{-6}, 1e^{-5}, 5e^{-4}, 5e^{-3}\}$
<i>Learning Rate</i>	Catagórico	$x \in \{0, 01, 0, 001\}$
<i>Momentum</i>	Uniforme	$x \in [0, 1]$

Tabela 3.3: Resultado dos hiperparâmetros otimizados para cada CNN com o *TPE*.

CNN	Hiperparâmetro			
	Nº. Unid. Ocultas	<i>Decay</i>	<i>Learning Rate</i>	<i>Momentum</i>
<b>LeNet-5</b>	2048 e 2048	$1e^{-6}$	0,01	0,337626380941364
<b>AlexNet</b>	512 e 2048	$5e^{-4}$	0,01	0,437116259431842
<b>Inception-V3</b>	2048	$5e^{-4}$	0,01	0,437116259431842
<b>VGG-16</b>	2048	$1e^{-6}$	0,01	0,337626380941364
<b>ResNet-50</b>	2048	$1e^{-6}$	0,01	0,337626380941364

### 3.5 CNNs e Estratégias de Treinamento

Para a tarefa de classificação automática de células *HEp-2* foram comparadas cinco arquiteturas *CNNs*: LeNet-5, AlexNet, Inception-V3, VGG-16 e ResNet-50 (descritas na Seção 2.4.4). Os conjuntos de imagens de células *HEp-2* resultantes da aplicação das seis estratégias de pré-processamento foram utilizados para avaliar o poder das *CNNs* em dois cenários diferentes: treinamento completo e *fine-tuning*, sendo considerado treinamento com e sem aumento do conjunto de dados.

No treinamento completo, as cinco *CNNs* avaliadas foram treinadas a partir do zero, com todos os parâmetros inicializados aleatoriamente e com as características aprendidas a partir do conjunto de imagens de células *HEp-2*. Para o treinamento baseado em *fine-tuning*, foram avaliados apenas três modelos: Inception-V3, VGG-16 e ResNet-50, com os parâmetros inicializados com valores adquiridos do treinamento com o grande conjunto de dados ImageNet [Deng et al., 2009]. Realizou-se a substituição da última camada e da camada de classificação *softmax*, por uma camada *softmax* com seis neurônios, a fim de adaptá-la ao número de classes presentes no problema de classificação estudado neste trabalho.

Foram congelados os valores dos parâmetros das primeiras 197 camadas da Inception-V3, os três blocos iniciais de camadas convolucionais da *VGG-16* e todas as camadas iniciais, exceto a última camada de blocos residuais da ResNet-50,

conforme descrito na Seção 2.4.5 e visto na Figura 3.4. Depois disso, as *CNNs* foram treinadas atualizando as camadas superiores durante o processo de *fine-tuning*. Para o *fine-tuning*, a maioria dos hiperparâmetros usados para treinamento completo foram mantidos, mas reduziu-se a taxa de aprendizado para 0.001, com base na literatura [Bengio, 2012].

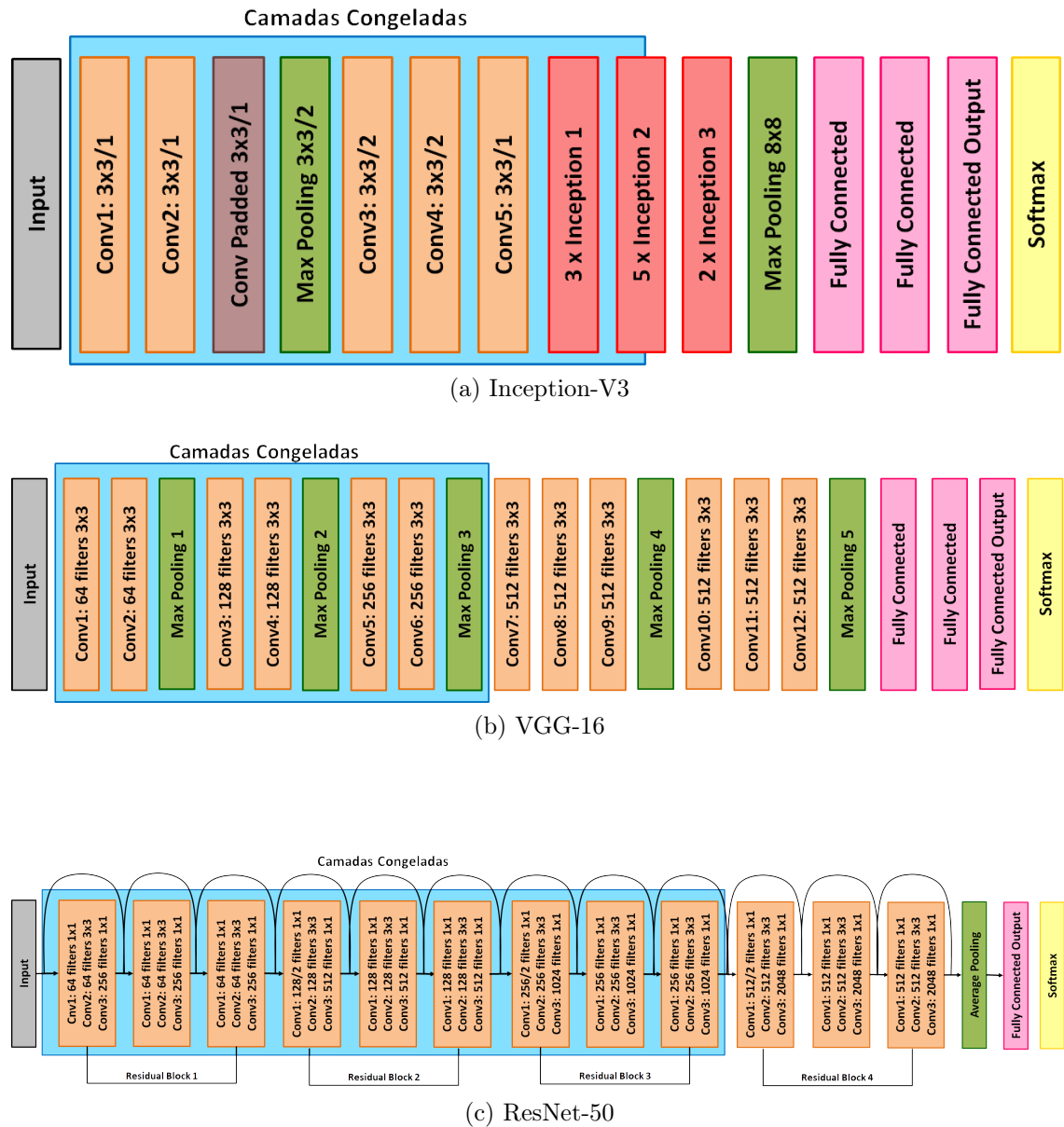


Figura 3.4: Arquiteturas *CNNs* em que aplicou-se *fine-tuning*. As caixas azuis indicam as camadas que foram congeladas.

### 3.6 Critério de Avaliação

O critério de avaliação considerado para classificar células *HEp-2* foi a Acurácia Média por Classe (do inglês, *Mean Class Accuracy - MCA*) [Lovell et al., 2014]. Para cada classe  $k$  calcula-se as classificações corretas (*Correct Classification Rate - CCR<sub>k</sub>*), definida pela Equação 3.1,

$$CCR_k = \frac{1}{n_k}(TP_k + TN_k) \quad (3.1)$$

sendo  $n_k$  o número de imagens pertencentes a  $k$ -ésima classe e  $TP_k$  e  $TN_k$  os índices de desempenho estimados [Hobson et al., 2014] (Veja a Seção 2.7.2). Assim, o *MCA* é calculado como o valor médio do  $CCR_k$ , conforme a Equação 3.2:

$$MCA = \frac{1}{n} \sum_{k=1}^n CCR_k \quad (3.2)$$

em que  $CCR_k$  é a acurácia da classificação calculada para a classe  $k$  e  $n$  é o número de classes. Este é o mesmo critério utilizado para avaliar a competição *HEp-2 ICPR 2014* [Hobson et al., 2014], a partir da qual obteve-se o conjunto de dados considerado no presente estudo.

# Capítulo 4

## Resultados e Discussões

Neste capítulo são apresentados os resultados deste trabalho, alcançados por meio da aplicação dos métodos apresentados no capítulo anterior. Os resultados mostram a efetividade da definição automática de hiperparâmetros, o impacto do pré-processamento e do aumento do conjunto de dados, além de explorar diferentes estratégias de treinamento e avaliar o desempenho da classificação de cada *CNN* em termos de acurácia e tempo de treinamento. No final deste capítulo, na Tabela 4.8 é apresentada a sumarização completa de todos os experimentos realizados.

Todos os experimentos foram realizados em uma máquina Intel i5 3.00 GHz, com 16 GB de memória RAM e uma GPU NVIDIA GeForce GTX Titan Xp com 12 GB de memória. Os experimentos foram implementados em Python 3.5, usando a biblioteca Keras <sup>1</sup> 2.0 [Chollet et al., 2015] com TensorFlow <sup>2</sup> 1.4.1 [Abadi et al., 2015] sob a versão 8.0 do CUDA e cuDNN 6.0. A otimização de hiperparâmetros com o algoritmo *TPE* foi feita com a biblioteca Hyperas<sup>3</sup> versão 0.4 e com a biblioteca Hyperopt<sup>4</sup> [Bergstra et al., 2013b,a], versão 0.1. Foi usado o sistema operacional Ubuntu 16.04.2 LTS.

### 4.1 Impacto do Pré-processamento

A Tabela 4.1 apresenta os resultados obtidos nos experimentos envolvendo todos os modelos *CNN*, variando a estratégia de pré-processamento e a normalização. Cada linha mostra o modelo *CNN* e cada coluna mostra a estratégia de pré-processamento. A acurácia da validação é apresentada considerando a média da validação de 5 *folds* e o tempo médio de treinamento. O gráfico na Figura 4.1 ilustra a acurácia da validação para cada modelo de *CNN* e as estratégias de pré-processamento. Baseado nos resultados da Tabela 4.1 e da Figura 4.1, observou-se que o uso das imagens originais, com ou sem subtração da média, sem técnicas de alargamento de contraste é a melhor opção para todos modelos *CNN* testados.

---

<sup>1</sup><http://keras.io>

<sup>2</sup><http://tensorflow.org>

<sup>3</sup><http://github.com/maxpumperla/hyperas>

<sup>4</sup><http://hyperopt.github.io/hyperopt>



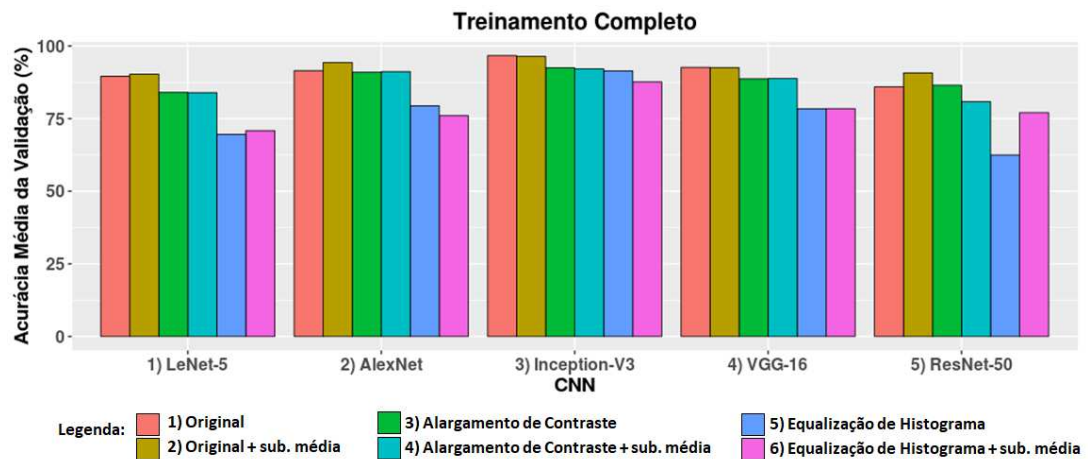
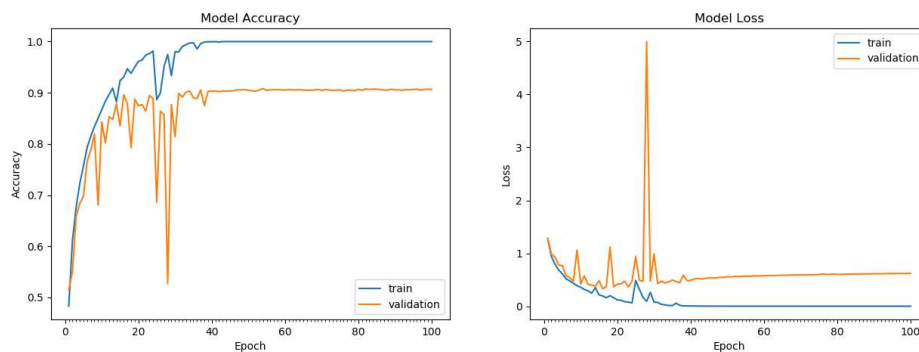
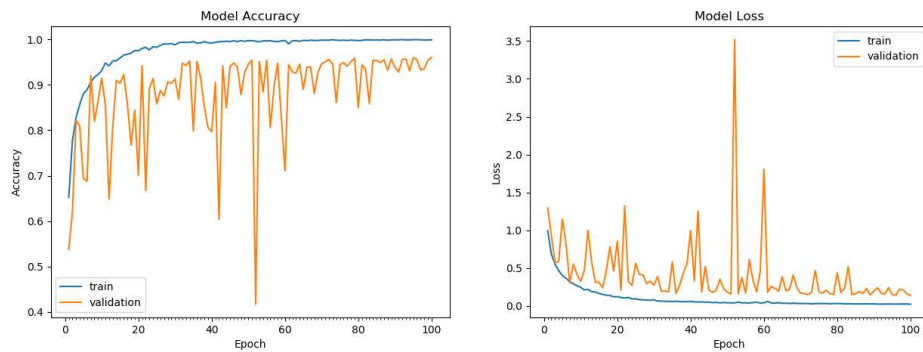


Figura 4.1: Acurácia da validação com  $5$ -fold para cada modelo  $CNN$  considerando as seis estratégias de pré-processamento: 1) imagem original; 2) imagem original + subtração da média; 3) alargamento de contraste; 4) alargamento de contraste + subtração da média; 5) equalização de histograma; e 6) equalização de histograma + subtração da média.

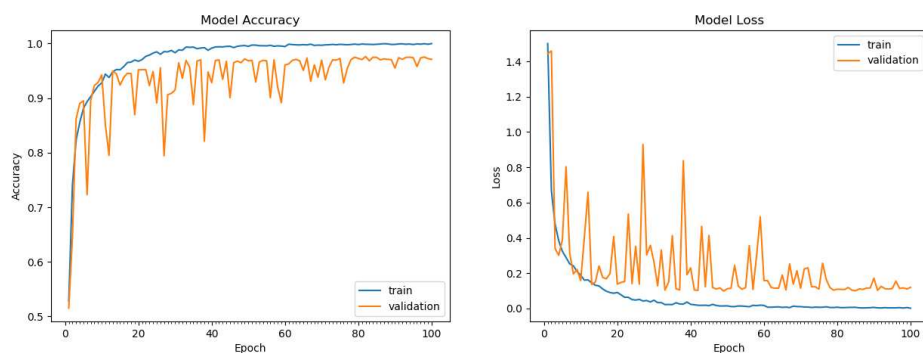
A evolução dos valores de perda ( $loss$ ) e acurácia ( $accuracy$ ) para os conjuntos de treinamento e validação são mostrados na Figura 4.2 considerando o conjunto de imagens original com subtração da média, validação com  $fold$ -1 e treino com  $fold$ s 2, 3, 4 e 5. O uso da validação cruzada permite estimar o erro do classificador, uma vez que há variações durante o treinamento e, portanto, é um método que exerce maior confiabilidade favorecendo a análise dos resultados. Observa-se que o erro de validação e a acurácia seguem os respectivos valores de treinamento, embora algumas variações sejam observadas nas primeiras épocas, especialmente para AlexNet e ResNet-50. Esse padrão é uma evidência importante de que não houve *overfitting* durante o treinamento.



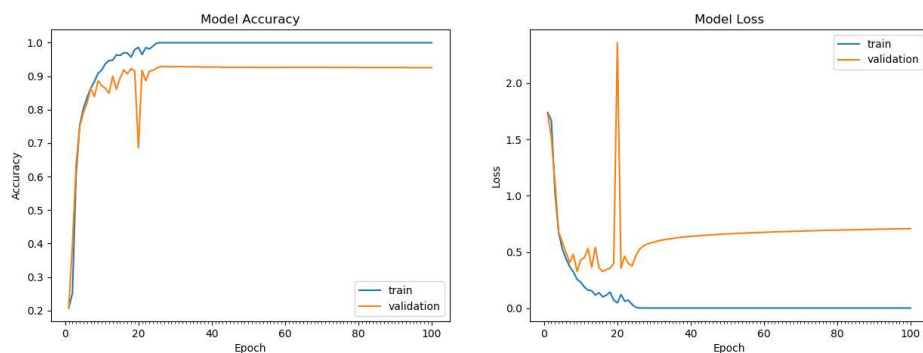
(a) LeNet-5



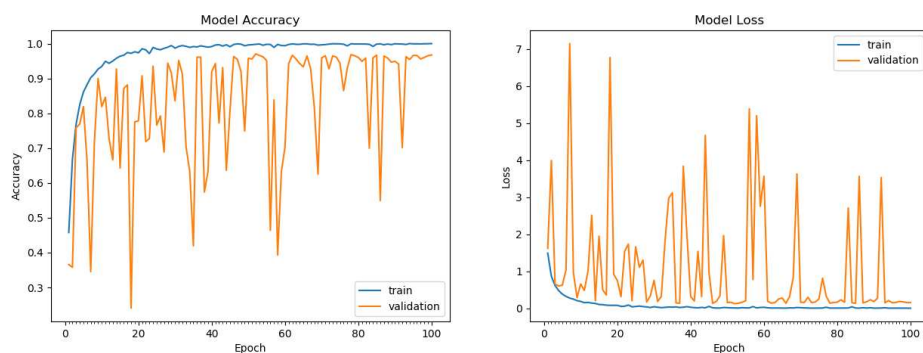
(b) AlexNet



(c) Inception-V3



(d) VGG-16



(e) ResNet-50

Figura 4.2: Gráficos mostrando a evolução dos valores de perda (*loss*) e dos valores de acurácia (*accuracy*) para o conjunto de validação em cada uma das *CNNs* avaliadas: (a) LeNet-5, (b) AlexNet, (c) Inception-V3, (d) VGG-16 e, (e) ResNet-50.

## 4.2 Impacto do Aumento do Conjunto de Dados

Os conjuntos de dados gerados a partir da aplicação de estratégias de pré-processamento foram submetidos ao procedimento de aumentar o conjunto de dados por meio rotações incrementais das imagens com valores aleatórios entre  $0^\circ$  e  $360^\circ$  ao longo da etapa de treinamento, conforme descrito na Seção 2.2.2. Aplicou-se o aumento do conjunto de dados em todas as estratégias de pré-processamento com o objetivo de investigar o seu impacto sobre o desempenho das *CNNs* em termos de acurácia.

O gráfico na Figura 4.3 ilustra a acurácia da validação para cada modelo de *CNN* e as estratégias de pré-processamento ao considerar o aumento do conjunto de dados. Como visto na Tabela 4.2, o aumento do conjunto de dados melhora o resultado em praticamente todos os modelos *CNN* independentemente de qual seja a estratégia de pré-processamento avaliado. Entretanto, ao comparar com os conjuntos de dados não aumentados, o modelo ResNet-50 considerando a equalização de histograma com subtração da média como estratégia de pré-processamento reduz a acurácia em 27,38 pontos percentuais.

Além disso, é possível observar que, exceto para o modelo ResNet-50, o uso das imagens originais com aumento do conjunto de dados é o melhor resultado entre todos os experimentos realizados considerando o treinamento completo das *CNNs*, com destaque para o modelo AlexNet que alcançou 98,00% de acurácia na validação cruzada com 5-*fold*. Para o modelo ResNet-50 o uso de alargamento de contraste como estratégia de pré-processamento é a melhor alternativa, permitindo 96,03% de acurácia.

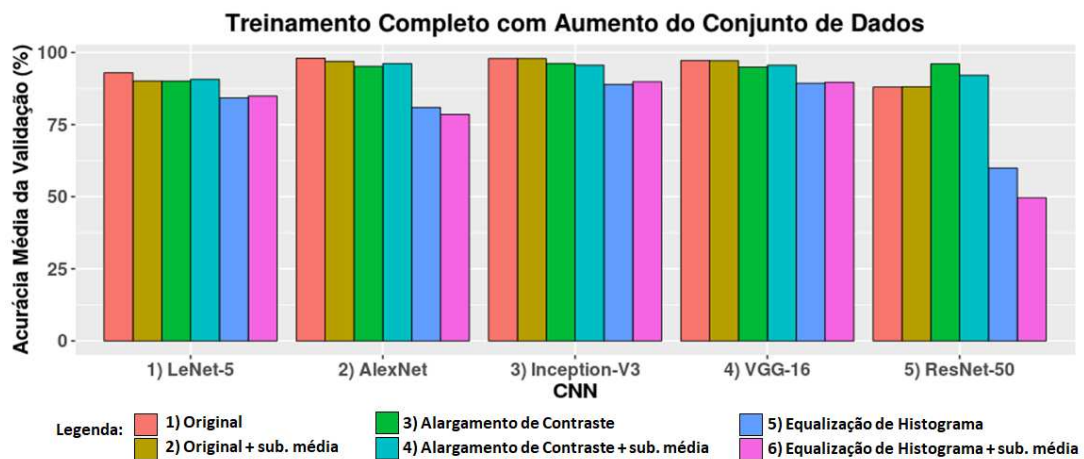


Figura 4.3: Acurácia da validação com 5-*fold* para cada modelo *CNN* considerando as seis estratégias de pré-processamento e aumento do conjunto de dados: 1) imagem original; 2) imagem original + subtração da média; 3) alargamento de contraste; 4) alargamento de contraste + subtração da média; 5) equalização de histograma; e 6) equalização de histograma + subtração da média.

Tabela 4.2: Acurácia da validação e tempo médio de treinamento, com 5-*fold*, para cada modelo *CNN* considerando treinamento com aumento do conjunto de dados.

<i>CNN</i>	Original	Original + sub. média	Alarg. de Contraste	Alarg. de + Sub. Média	Contraste Eq. Hist.	Eq. Hist. + Sub. Média
LeNet-5	<b>92,94</b>	90,09	90,07	90,66	84,22	84,88
	145,23 min.	145,62 min.	145,29 min.	145,80 min.	145,22 min.	145,79 min.
AlexNet	<b>98,00</b>	96,92	95,16	96,16	80,94	78,55
	145,38 min.	146,00 min.	145,51 min.	145,86 min.	145,51 min.	146,01 min.
Inception-V3	<b>97,89</b>	97,88	96,19	95,56	88,92	89,83
	151,08 min.	151,67 min.	151,32 min.	151,86 min.	151,42 min.	151,82 min.
VGG-16	<b>97,17</b>	97,12	94,93	95,57	89,35	89,68
	152,48 min.	152,66 min.	152,43 min.	152,88 min.	152,47 min.	152,75 min.
ResNet-50	88,00	88,13	<b>96,03</b>	92,09	59,94	49,67
	153,00 min.	153,51 min.	153,14 min.	153,56 min.	153,16 min.	153,62 min.

### 4.3 Comparação entre diferentes arquiteturas

Para comparar os cinco modelos *CNN* treinados a partir do zero, escolheu-se a melhor estratégia de pré-processamento para cada modelo, considerando a acurácia 5-*fold* mostrada nas seções anteriores, com e sem aumento de dados (Tabela 4.1 e Tabela 4.2, respectivamente). Os modelos foram treinados novamente considerando todo o conjunto de treinamento (todos os 5-*folds*) e teste final com conjunto de teste (Veja a Figura 3.2). Para LeNet, AlexNet, Inception-V3 e VGG-16, escolheu-se o conjunto de dados original com aumento de dados, mas sem subtração da imagem média, pois essas técnicas alcançaram uma acurácia de 92,94%, 98,00%, 97,89% e 97,17%, respectivamente, considerando a validação com 5-*fold*. Para ResNet-50, escolheu-se o conjunto de dados com alargamento de contraste e aumento de dados, que alcançou uma acurácia de 96,03%, considerando a validação com 5-*fold*. A Tabela 4.3 mostra a acurácia da validação com 5-*fold*, acurácia do teste e o tempo de treinamento para cada modelo *CNN*, treinado com todo o conjunto de treinamento e a predição feita com o conjunto de teste.

Tabela 4.3: Acurácia média do 5-*fold* e acurácia do teste para cada modelo *CNN*, considerando a melhor estratégia de pré-processamento.

<i>CNN</i>	Acurácia Validação (%)	Acurácia Teste (%)
LeNet-5 (original + aumento dos dados)	92,94	94,35
AlexNet (original + aumento dos dados)	98,00	94,35
Inception-V3 (original + aumento dos dados)	97,89	98,28
VGG-16 (original + aumento dos dados)	97,17	96,82
ResNet-50 (alarg. de contraste + aumento dos dados)	96,03	96,38

Considerando as acurácias do teste quando o treinamento é implementado em todo o conjunto de treinamento e as predições são feitas com o conjunto de testes, o melhor resultado foi obtido pelo modelo Inception-V3, com 98,28% de acurácia. Os modelos VGG-16 e ResNet-50, estão na segunda e terceira posição com 96,82% e 96,38%, respectivamente. Os modelos AlexNet e LeNet-5 apresentaram a mesma acurácia para o conjunto de testes, alcançando 94,35%, embora o modelo AlexNet tenha alcançado o melhor resultado de validação de treinamento. A alta acurácia de validação seguida de resultados de teste menores indica que o modelo AlexNet perdeu algum poder de generalização durante o treinamento.

Para ilustrar alguns aspectos de classificação investigados neste trabalho, a Tabela 4.4 mostra as matrizes de confusão para cada modelo *CNN* apresentado na Tabela 4.3. Esses valores na matriz de confusão consideram o treinamento feito com todo o conjunto de treinamento e as predições sobre o conjunto de testes. É possível ver que, com exceção da Inception-V3, as *CNNs* classificaram erroneamente uma parte considerável das células *homogeneous* como *speckled*. Os outros resultados variam de acordo com cada conjunto de dados e cada modelo *CNN*.

Tabela 4.4: Matriz de confusão do melhor resultado para cada *CNN*.

(a) LeNet-5 (original + aumento de dados).

	Cent.	Golgi	Homo.	NuMem.	Nucleo.	Speck.
Cent.	<b>93,22</b>	0,00	0,22	0,00	3,06	3,50
Golgi	0,00	<b>96,69</b>	0,00	1,65	1,65	0,00
Homo.	0,00	0,00	<b>91,57</b>	1,69	0,72	6,02
NuMem.	0,00	0,54	0,54	<b>98,10</b>	0,54	0,27
Nucleo.	0,92	0,00	0,46	0,46	<b>94,69</b>	3,46
Speck.	1,48	0,64	1,69	0,64	1,48	<b>94,07</b>

(b) AlexNet (original + aumento de dados)

	Cent.	Golgi	Homo.	NuMem.	Nucleo.	Speck.
Cent.	<b>94,97</b>	0,22	0,22	0,00	2,41	2,19
Golgi	1,65	<b>94,21</b>	1,65	1,65	0,83	0,00
Homo.	0,00	0,00	<b>95,66</b>	1,45	0,72	2,17
NuMem.	0,00	1,63	2,17	<b>95,38</b>	0,27	0,54
Nucleo.	0,92	0,23	1,62	0,23	<b>94,46</b>	2,54
Speck.	1,91	0,21	4,24	0,85	1,06	<b>91,93</b>

(c) Inception-V3 (original + aumento de dados)

	Cent.	Golgi	Homo.	NuMem.	Nucleo.	Speck.
Cent.	<b>99,12</b>	0,22	0,00	0,00	0,22	0,44
Golgi	2,48	<b>96,69</b>	0,00	0,00	0,83	0,00
Homo.	0,00	0,00	<b>98,55</b>	0,72	0,24	0,48
NuMem.	0,00	0,00	0,00	<b>99,46</b>	0,54	0,00
Nucleo.	0,23	0,23	0,00	0,23	<b>98,61</b>	0,69
Speck.	0,64	0,21	1,91	0,64	0,21	<b>96,40</b>

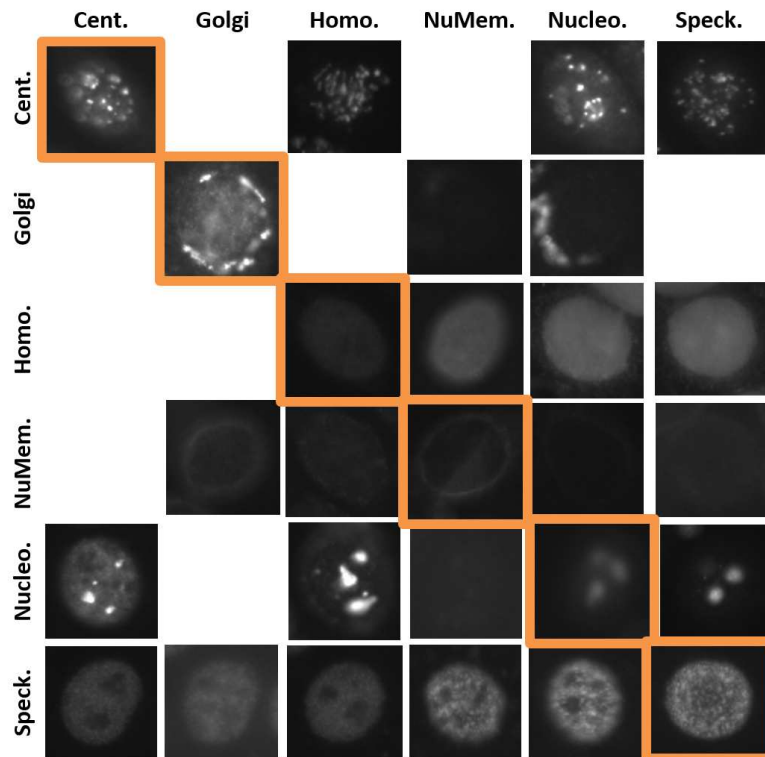
(d) VGG-16 (original + aumento de dados)

	Cent.	Golgi	Homo.	NuMem.	Nucleo.	Speck.
Cent.	<b>98,91</b>	0,00	0,00	0,00	0,66	0,44
Golgi	2,48	<b>94,21</b>	0,00	3,31	0,00	0,00
Homo.	0,00	0,00	<b>94,46</b>	2,41	0,72	2,41
NuMem.	0,00	0,27	0,00	<b>99,73</b>	0,00	0,00
Nucleo.	0,46	0,23	0,23	0,23	<b>98,15</b>	0,69
Speck.	1,69	0,00	2,75	0,85	0,64	<b>94,07</b>

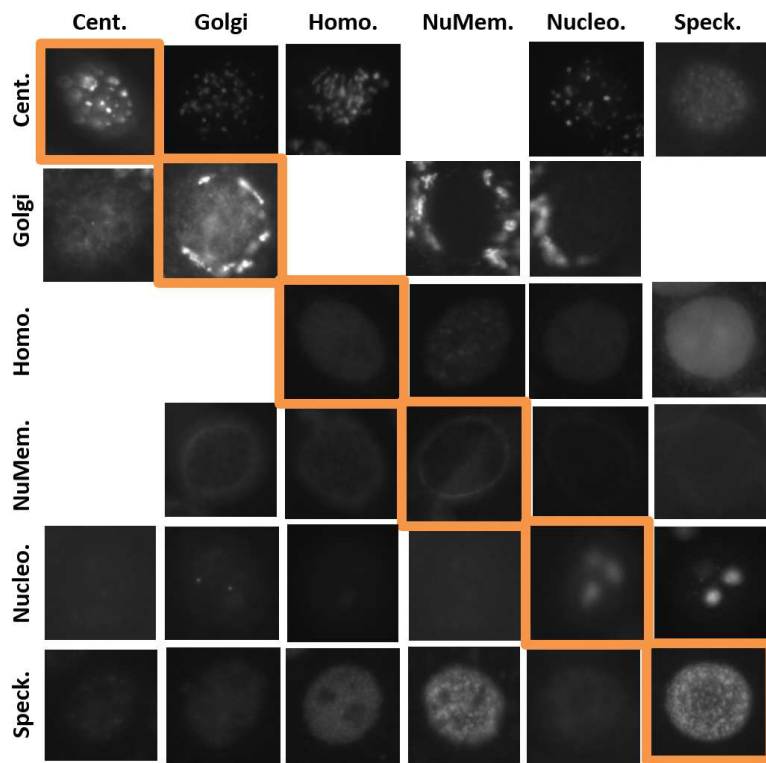
(e) ResNet-50 (alargamento de contraste + aumento de dados)

	Cent.	Golgi	Homo.	NuMem.	Nucleo.	Speck.
Cent.	<b>94,53</b>	0,00	0,00	0,00	4,38	1,09
Golgi	0,83	<b>95,04</b>	0,83	0,00	3,31	0,00
Homo.	0,24	0,00	<b>97,83</b>	0,24	0,24	1,45
NuMem.	0,00	1,36	1,09	<b>96,74</b>	0,27	0,54
Nucleo.	0,00	0,46	0,23	0,00	<b>98,61</b>	0,69
Speck.	0,21	0,00	3,81	0,21	0,85	<b>94,92</b>

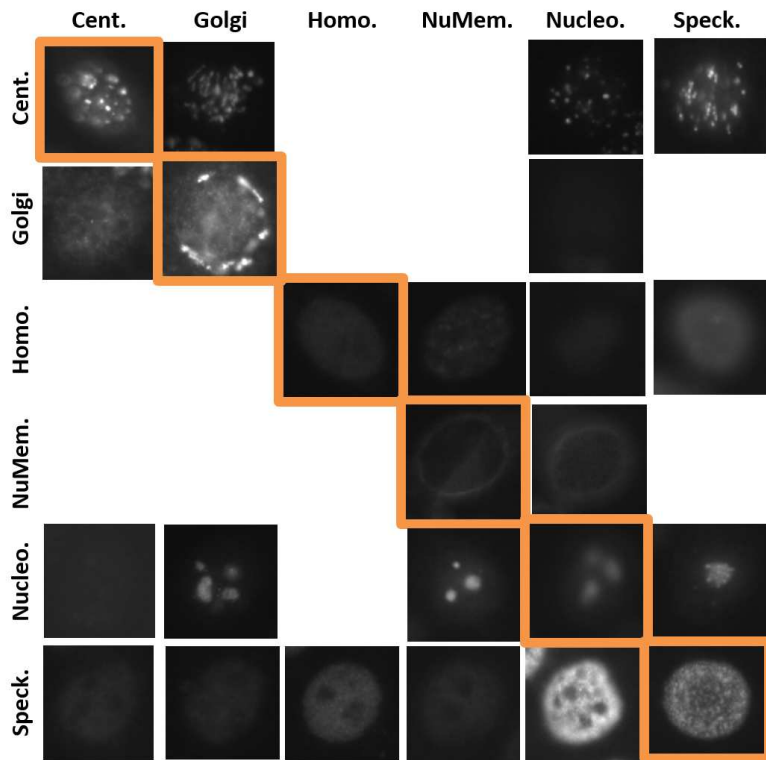
Para uma melhor compreensão e visualização das imagens classificadas erroneamente, a Figura 4.4 apresenta uma amostra de cada classificação celular possível de acordo com as matrizes de confusão apresentadas na Tabela 4.4. A diagonal principal contém amostras de células classificadas corretamente, enquanto as imagens que não estão na diagonal principal (linha, coluna) são amostras de células pertencentes a classe da “linha”, mas classificadas erroneamente como a classe da “coluna”.



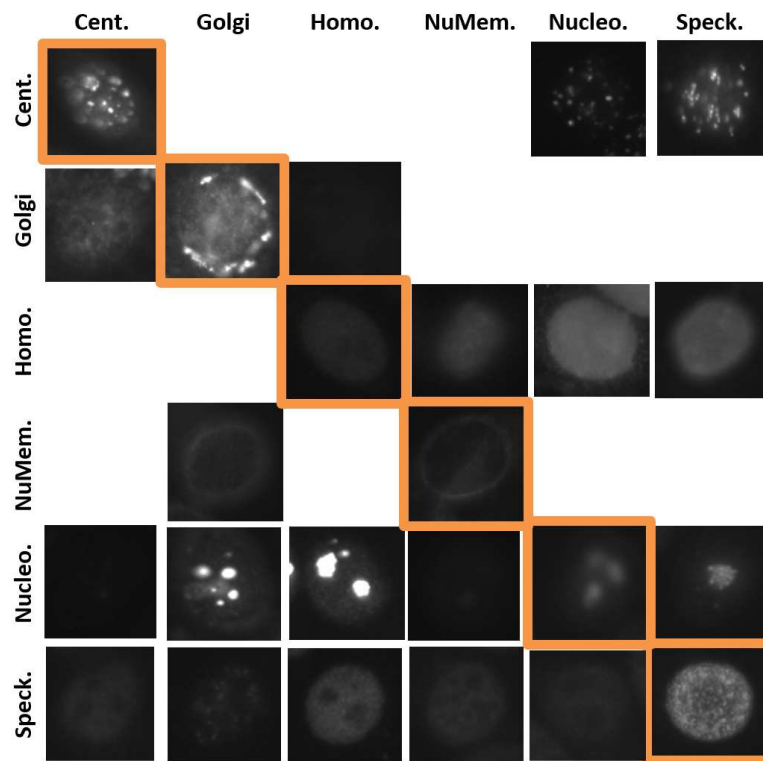
(a) LeNet-5



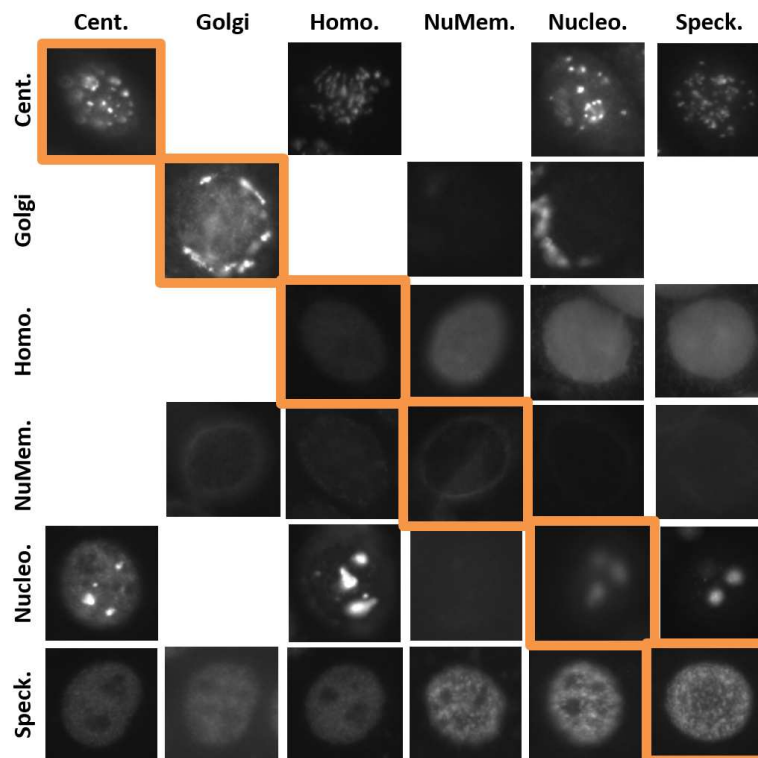
(b) AlexNet



(c) Inception-V3



(d) VGG-16



(e) ResNet-50

Figura 4.4: Exemplos de imagens classificadas de forma correta e incorreta de acordo com as matrizes de confusão.

## 4.4 Treinamento Completo Vs *Fine-Tuning*

Neste trabalho, realizou-se a comparação do desempenho de três modelos *CNN* mais profundos, Inception-V3, VGG-16 e ResNet-50, para diferentes estratégias de treinamento (conforme descrito na Seção 2.4.5): treinamento completo e *fine-tuning*. Para o treinamento aplicando a estratégia de *fine-tuning*, os parâmetros dos modelos *CNN* foram inicializados com valores adquiridos do treinamento do modelo com o grande conjunto de dados ImageNet [Deng et al., 2009]. Realizou-se a substituição da última camada e da camada de classificação *softmax*, por uma camada *softmax* com seis neurônios, a fim de adaptá-la ao número de classes presentes no problema de classificação estudado neste trabalho.

A Tabela 4.5 mostra a acurácia média de validação cruzada com 5-*fold*, considerando o treinamento em todas as estratégias de pré-processamento. Quanto aos experimentos com treinamento completo, o modelo VGG-16 treinado com imagens pré-processadas com alargamento de contraste e subtração da média alcançou o melhor resultado 96,07%. Interessantemente, para todos os modelos *CNNs*, as estratégias de pré-processamento com ou sem subtração da média apresentam resultados muito similares. Ao comparar com o treinamento completo (Tabela 4.1) é possível observar que o *fine-tuning* proporcionou melhorias para todos os modelos *CNNs*, exceto para o ResNet-50.

Tabela 4.5: Acurácia da validação e tempo médio de treinamento, com 5-*fold*, para cada modelo *CNN* considerando treinamento com *fine-tuning* sem aumento do conjunto de dados.

<i>CNN</i>	Original	Original + sub. média	Alarg. de Contraste	Alarg. de + Sub. Média	Contraste + Sub. Média	Eq. Hist.	Eq. Hist. + Sub. Média
Inception-V3	90,50	<b>90,55</b>	90,01	90,00		85,78	85,87
	67,08 min.	66,87 min.	67,33 min.	66,84 min.		67,02 min.	66,93 min.
VGG-16	94,32	94,25	95,97	<b>96,07</b>		93,64	93,57
	76,29 min.	76,44 min.	76,72 min.	76,75 min.		77,20 min.	77,23 min.
ResNet-50	93,31	<b>93,37</b>	92,89	92,95		90,21	90,11
	61,77 min.	61,67 min.	61,87 min.	61,83 min.		61,78 min.	61,72 min.

A Tabela 4.6 mostra os resultados com *fine-tuning* e aumento de dados. O aumento de dados melhorou os resultados para todos os modelos. Ao analisar qual é a melhor estratégia de pré-processamento, os melhores resultados foram obtidos ao usar imagens originais para Inception-V3 e ResNet-50 e imagens com alargamento de contraste para VGG-16.

Foi observado que ao considerar as melhores estratégias de pré-processamento (originais e alargamento de contraste) com e sem subtração da média de imagem, para os três modelos *CNN*, os valores de acurácia da validação são muito próximos. No entanto, quando treinados com aumento de dados e com subtração de imagem média, a acurácia dos modelos Inception-V3 e ResNet-50 aumentou em 2,46 e 2,33

pontos percentuais, respectivamente. Com o aumento de dados de treinamento, alargamento de contraste e subtração de imagem média, a acurácia do modelo VGG-16 aumentou em 1,24 pontos percentuais. Além disso, o *fine-tuning* trouxe melhorias significativas para o modelo ResNet-50 em todas as estratégias de pré-processamento em comparação com o treinamento completo e aumento de dados (Tabela 4.2).

Tabela 4.6: Acurácia da validação e tempo médio de treinamento, com 5-*fold*, para cada modelo *CNN* considerando treinamento com *fine-tuning* e aumento de dados.

<i>CNN</i>	Original	Original + sub. média	Alarg. de Contraste	Alarg. de + Sub. Média	Contraste Eq. Hist.	Eq. Hist. + Sub. Média
<b>Inception-V3</b>	92,97	<b>93,01</b>	90,48	90,48	87,15	87,32
	149,18 min.	149,71 min.	149,47 min.	149,79 min.	149,27 min.	149,81 min.
<b>VGG-16</b>	93,32	93,39	97,24	<b>97,31</b>	95,85	96,15
	150,11 min.	150,73 min.	150,39 min.	150,94 min.	150,23 min.	150,91 min.
<b>ResNet-50</b>	<b>95,91</b>	95,70	94,06	94,08	92,63	92,69
	150,39 min.	150,97 min.	150,53 min.	151,18 min.	150,61 min.	151,09 min.

Para cada modelo *CNN* treinado com *fine-tuning*, foram selecionadas as melhores estratégias de pré-processamento para treinar os modelos considerando todo o conjunto de treinamento e as previsões sobre o conjunto de testes. Todos os melhores resultados foram obtidos com aumento do conjunto de dados, conforme pode ser visto na Tabela 4.6. Ao considerar as estratégias de pré-processamento, os melhores resultados foram obtidos usando imagens originais e com subtração da média para Inception-V3, imagens com alargamento de contraste e subtração da média para VGG-16 e imagens originais para ResNet-50. Apesar de todos os modelos terem o melhor resultado com aumento de dados, as diferenças são muito pequenas, a melhoria máxima foi de 2,46 pontos percentuais para o Inception-V3. Os valores na Tabela 4.7 foram obtidos a partir do treinamento dos modelos considerando todo o conjunto de treinamento e previsão com o conjunto de testes.

Tabela 4.7: Acurácia da validação e teste, com 5-*fold*, para cada modelo *CNN* considerando a melhor estratégia de pré-processamento

<i>CNN</i>	Acurácia da Validação (%)	Acurácia do Teste (%)
Inception-V3 (original + sub. média + aum. dos dados)	93,01	92,71
VGG-16 (alarg. de contraste + sub. média + aum. dos dados)	97,31	96,69
ResNet-50 (original + aum. dos dados)	95,91	95,50

A sumarização completa dos experimentos discutidos neste capítulo é vista na Tabela 4.8.

Tabela 4.8: Sumarização de todos os experimentos apresentados neste trabalho, considerando validação cruzada com *5-fold* para acurácia da validação, tempo de treinamento e acurácia do teste (entre parênteses). O Aumento dos Dados é representado por A.D.

		Original		Original + sub. média		Alargamento de Contraste		Alargamento de Contraste + sub. média		Equalização de Histograma		Equalização de Histograma + sub. média	
			A.D.		A.D.		A.D.		A.D.		A.D.		A.D.
Treinamento Completo	LeNet-5	89,57 21,45 min.	<b>92,94</b> <b>145,23 min.</b> (Teste: 94,35)	90,27 22,37 min.	90,09 145,62 min.	84,01 21,44 min.	90,07 145,29 min.	83,93 22,37 min.	90,66 145,80 min.	69,60 21,54 min.	84,22 145,22 min.	70,84 22,48 min.	84,88 145,79 min.
	AlexNet	91,51 39,00 min.	<b>98,00</b> <b>145,38 min.</b> (Teste: 94,35)	94,33 39,00 min.	96,92 146,00 min.	90,95 39,00 min.	95,16 145,51 min.	91,17 39,00 min.	96,16 145,86 min.	79,39 39,00 min.	80,94 145,51 min.	76,02 39,00 min.	78,55 146,01 min.
	Inception-V3	96,69 117,00 min.	<b>97,89</b> <b>151,08 min.</b> (Teste: 98,28)	96,40 117,00 min.	97,88 151,67 min.	92,49 117,00 min.	96,19 151,32 min.	92,15 117,00 min.	95,56 151,86 min.	91,43 117,00 min.	88,92 151,42 min.	87,62 117,00 min.	89,83 151,82 min.
	VGG-16	92,63 140,00 min.	97,17 152,48 min.	92,54 140,00 min.	97,12 152,66 min.	88,70 140,00 min.	94,93 152,43 min.	88,79 140,00 min.	95,57 152,88 min.	78,39 140,00 min.	89,35 152,47 min.	78,43 140,00 min.	89,68 152,75 min.
	ResNet-50	85,90 132,00 min.	88,00 153,00 min.	90,73 132,00 min.	88,13 153,51 min.	86,50 132,00 min.	<b>96,03</b> <b>153,56 min.</b> (Teste: 96,38)	80,81 132,00 min.	92,09 153,56 min.	62,47 132,00 min.	59,94 153,16 min.	77,05 132,00 min.	49,67 153,62 min.
Fine-Tuning	Inception-V3	90,50 67,08 min.	<b>92,97</b> <b>149,18 min.</b> (Teste: 92,71)	90,55 66,87 min.	93,01 149,71 min.	90,01 67,33 min.	90,48 149,47 min.	90,00 66,84 min.	90,48 149,79 min.	85,78 67,02 min.	87,15 149,27 min.	85,87 66,93 min.	87,32 149,81 min.
	VGG-16	94,32 76,29 min.	93,32 150,11 min.	94,25 76,44 min.	93,39 150,73 min.	95,97 76,72 min.	<b>97,24</b> <b>150,39 min.</b> (Teste: 96,69)	96,07 76,75 min.	97,31 150,94 min.	93,64 77,20 min.	95,85 150,23 min.	93,57 77,23 min.	96,15 150,91 min.
	ResNet-50	93,31 61,77 min.	<b>95,91</b> <b>150,39 min.</b> (Teste: 95,50)	93,37 61,67 min.	95,70 150,97 min.	92,89 61,87 min.	94,06 150,53 min.	92,95 61,83 min.	94,08 151,18 min.	90,21 61,78 min.	92,63 150,61 min.	90,11 61,72 min.	92,69 151,09 min.

## 4.5 Comparação com Trabalhos Relacionados

Realizou-se a comparação entre o melhor resultado, em termos da acurácia do teste, obtido pelo modelo Inception-V3 sem pré-processamento e com aumento de dados, com outros trabalhos de última geração presentes literatura. O melhor resultado geral do presente trabalho, considerando acurácia do conjunto de testes, foi do Inception-V3, treinado a partir do zero com imagens originais e com aumento de dados (98,28% apresentado na Tabela 4.3).

Todos os valores apresentados na Tabela 4.9 correspondem aos melhores resultados relatados nos trabalhos originais. É importante ressaltar que todos os trabalhos consideraram o mesmo conjunto de dados de células *HEp-2* do ICPR 2014. Os resultados dos experimentos realizados nesta dissertação são superiores (em termos de acurácia) aos resultados publicados pelos trabalhos comparados.

Tabela 4.9: Maior acurácia dos métodos de classificação sobre o conjunto de dados *HEp-2* do ICPR 2014 para cada artigo comparado.

Método	Acurácia (%)
Gagnaniello et al. [2016]	82,60
Manivannan et al. [2016]	87,10
Kastaniotis et al. [2017]	82,30
Gao et al. [2017]	88,58
Gao et al. [2017] (com aumento dos dados)	96,76
Lei et al. [2018]	91,38
<b>Presente trabalho (2018)</b>	<b>98,28</b>

# Capítulo 5

## Conclusões

Este trabalho apresentou métodos para classificar automaticamente células *Hep-2* em imagens de imunofluorescência, sendo o primeiro a considerar os modelos AlexNet, Inception-V3 e VGG-16 para essa tarefa, ao lado dos modelos LeNet-5 e ResNet-50. Adicionalmente, foi estudada a estimação automática de hiperparâmetros utilizando *Tree of Parzen Estimators (TPE)* e uso de diferentes técnicas de pré-processamento de imagem baseadas em combinações de realce de contraste e normalização de imagem.

Após um extenso estudo a fim de avaliar o impacto do pré-processamento de imagens com base em: melhorias de contraste, centralização dos dados e aumento do conjunto de dados; os resultados apresentados permitem concluir que, para a classificação de imagens de células *Hep-2*, as *CNNs* estudadas obtiveram melhores resultados quando o conjunto de dados aumentado é analisado, ou seja, nenhuma estratégia de pré-processamento é aplicada além do aumento de dados.

Ao realizar uma análise comparativa entre os 5 modelos *CNNs*, verificou-se que os modelos LeNet-5, AlexNet, VGG-16 e ResNet-50 alcançaram resultados muito próximos em termos de acurácia, tanto para a validação cruzada com *5-fold* quanto para o conjunto de testes. Ao considerar o conjunto de testes, verifica-se que a acurácia variou de 94,35% (LeNet-5 e AlexNet) para 96,82% (ResNet-50). Esta diferença considerada é de apenas 2.47 pontos percentuais.

Dois estratégias de treinamento de *CNNs* foram exploradas, permitindo verificar que embora a técnica de *fine-tuning* pareça promissora, ela não obteve resultados significativamente melhores do que as *CNNs* treinadas a partir do zero. Talvez, deva ser considerada em cenários em que o tempo de treinamento é um problema. Curiosamente, os modelos VGG-16 e ResNet-50 foram os únicos que tiraram melhor proveito da técnica de *fine-tuning*, sendo a acurácia média da validação de *5-folds* superior a 92% em todas as 6 estratégias de pré-processamento. Ao comparar *fine-tuning* e treinamento completo, observa-se que a melhora mais significativa obtida pelo modelo VGG-16 ocorreu ao considerar equalização de histograma como pré-processamento sem aumento de dados, sendo uma melhoria de 15,25 pontos percentuais. Já o modelo ResNet-50 obteve a surpreendente melhoria de 43,02 pontos percentuais, considerando equalização de histograma com subtração da média e aumento de dados.

A combinação dos métodos propostos nesta pesquisa podem ser usados por especialistas em medicina para escolher as melhores estratégias de classificação de células *HEp-2* para identificar e acompanhar as condições autoimunes e as doenças que as causam. Além disso, este trabalho representa um importante passo na investigação de diferentes técnicas de pré-processamento, comparação de múltiplas arquiteturas *CNNs* e alcance de resultados, em termos de acurácia, superiores aos obtidos pelos principais trabalhos relacionados (98,28%, treinando o modelo Inception-V3 a partir do zero, sem pré-processamento e com aumento de dados).

Como trabalhos futuros, sugere-se: a) considerar a otimização de mais hiperparâmetros e outros algoritmos de otimização; b) testar outras estratégias de aumento de dados, como *flips* verticais e horizontais, rotações em ângulos aleatórios, *random crops*, variação nas intensidades de imagem; c) aplicar outras estratégias de normalização, como a normalização do desvio padrão e *ZCA Whitening*; d) usar as *CNNs* como extratores de características; e) aplicar as estratégias propostas em outro conjunto de dados de imagens de células *HEp-2* e em outros tipos de células, a fim de avaliar se as descobertas feitas neste trabalho podem ser estendidas a outros conjuntos de dados semelhantes.

# Referências Bibliográficas

- Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y. & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Agaian, S.; Madhukar, M. & Chronopoulos, A. T. (2014). Automated screening system for acute myelogenous leukemia detection in blood microscopic images. *IEEE Systems Journal*, 8(3):995–1004.
- Bach, J.-F. (2002). The effect of infections on susceptibility to autoimmune and allergic diseases. *New England Journal of Medicine*, 347(12):911–920.
- Bayramoglu, N.; Kannala, J. & Heikkilä, J. (2015). Human epithelial type 2 cell classification with convolutional neural networks. In *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 1–6.
- Bengio, Y. (2009). Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127.
- Bengio, Y. (2012). *Practical Recommendations for Gradient-Based Training of Deep Architectures*, pp. 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Benoist, C. & Mathis, D. (2001). Autoimmunity provoked by infection: how good is the case for t cell epitope mimicry? *Nature immunology*, 2(9):797–801.
- Bergstra, J.; Yamins, D. & Cox, D. D. (2013a). Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*, pp. 13–20. Citeseer.
- Bergstra, J.; Yamins, D. & Cox, D. D. (2013b). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13*, pp. I–115–I–123. JMLR.org.

- Bergstra, J. S.; Bardenet, R.; Bengio, Y. & Kégl, B. (2011). Algorithms for hyperparameter optimization. In Shawe-Taylor, J.; Zemel, R. S.; Bartlett, P. L.; Pereira, F. & Weinberger, K. Q., editores, *Advances in Neural Information Processing Systems 24*, pp. 2546--2554. Curran Associates, Inc.
- Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536--538.
- Chatenoud, L. (2016). Precision medicine for autoimmune disease. *Nat Biotech*, 34(9):930–932. News and Views.
- Chatfield, K.; Simonyan, K.; Vedaldi, A. & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531.
- Cheng, H.; Shi, X.; Min, R.; Hu, L.; Cai, X. & Du, H. (2006). Approaches for automated detection and classification of masses in mammograms. *Pattern Recognition*, 39(4):646 – 668. Graph-based Representations.
- Chervonsky, A. V. (2010). Influence of microbial environment on autoimmunity. *Nature immunology*, 11(1):28--35.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- Cong, Y.; Wang, S.; Liu, J.; Cao, J.; Yang, Y. & Luo, J. (2015). Deep sparse feature selection for computer aided endoscopy diagnosis. *Pattern Recognition*, 48(3):907 – 917.
- Costa, L. d. F. & Cesar-Jr., R. M. (2009). *Shape Classification and Analysis: Theory and Practice*. CRC Press, Inc., Boca Raton, FL, USA, 2nd edição.
- Dellavance, A.; G. Junior, A.; Cintra, A. F.; Ximenes, A. C.; Nuccitelli, B.; Mühlen, C. A. v.; Bichara, C. D.; Yano, C.; Carvalho, D. G.; Bonfa, E. S.; Guimarães, F. N.; Mundim, H. M.; Pfrimer, I. A.; Rego, J.; Andrade, L. E.; Mesquita, M. M.; Santiago, M. B.; Silva, N. A.; Miranda, P. J.; Leser, P.; Francescantonio, P. L. C.; Jarach, R.; Levy, R. A.; Neves, S. P.; Cruvinel, W. M. & Santos, W. S. (2002). I consenso nacional para padronização dos laudos de FAN HEp-2. *Jornal Brasileiro de Patologia e Medicina Laboratorial*, 38(3):207–216.
- Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K. & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.
- Devijver, P. A. & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*. Prentice-Hall.
- Díaz, G.; González, F. A. & Romero, E. (2009). A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. *Journal of Biomedical Informatics*, 42(2):296 – 307.

- Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, 31(4–5):198 – 211. Computer-aided Diagnosis (CAD) and Image-guided Decision Support.
- Dougherty, G. (2009). *Digital image processing for medical applications*.
- Duchi, J.; Hazan, E. & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.
- Duda, R. O.; Hart, P. E. & Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA.
- Egerer, K.; Roggenbuck, D.; Hiemann, R.; Weyer, M.-G.; Büttner, T.; Radau, B.; Krause, R.; Lehmann, B.; Feist, E. & Burmester, G.-R. (2010). Automated evaluation of autoantibodies on human epithelial-2 cells as an approach to standardize cell-based immunofluorescence tests. *Arthritis Research & Therapy*, 12(2):1–9.
- Ensafi, S.; Lu, S.; Kassim, A. A. & Tan, C. L. (2017). Accurate hep-2 cell classification based on sparse bag of words coding. *Computerized Medical Imaging and Graphics*, 57:40 – 49. Recent Developments in Machine Learning for Medical Imaging Applications.
- Faria, L. C.; Rodrigues, L. F. & Mari, J. F. (2018). Cell classification using handcrafted features and bag of visual words. In *XIV Workshop de Visão Computacional*, Ilhéus, BA, Brazil. WVC.
- Foggia, P.; Percannella, G.; Saggese, A. & Vento, M. (2014). Pattern recognition in stained hep-2 cells: Where are we now? *Pattern Recognition*, 47(7):2305 – 2314.
- Fritzler, M. J. (2011). The antinuclear antibody test: last or lasting gasp? *Arthritis & Rheumatism*, 63(1):19–22.
- Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119 – 130.
- Gao, Z.; Wang, L.; Zhou, L. & Zhang, J. (2017). Hep-2 cell image classification with deep convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 21(2):416–428.
- Gonzalez, R. C. & Woods, R. E. (2007). *Digital image processing 3rd edition*.
- González-Hidalgo, M.; Guerrero-Peña, F. A.; Herold-García, S.; i Capó, A. J. & Marrero-Fernández, P. D. (2015). Red blood cell cluster separation from digital images for use in sickle cell disease. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1514–1525.
- Goodfellow, I.; Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.

- Graganiello, D.; Sansone, C. & Verdoliva, L. (2016). Cell image classification by a scale and rotation invariant dense local descriptor. *Pattern Recognition Letters*, 82, Part 1:72 – 78. Pattern recognition Techniques for Indirect Immunofluorescence Images Analysis.
- Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J. & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354 – 377.
- Gual-Arnau, X.; Herold-García, S. & Simó, A. (2015). Erythrocyte shape classification using integral-geometry-based methods. *Medical & Biological Engineering & Computing*, 53(7):623--633.
- Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S. & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48. Recent Developments on Deep Big Vision.
- Hawkins, J. & Blakeslee, S. (2007). *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan, New York, NY, USA.
- Haykin, S. (2007). *Neural Networks: A Comprehensive Foundation (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- He, K.; Zhang, X.; Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.
- Hobson, P.; Lovell, B. C.; Percannella, G.; Vento, M. & Wiliem, A. (2014). Classifying anti-nuclear antibodies hep-2 images: A benchmarking platform. In *2014 22nd International Conference on Pattern Recognition*, pp. 3233–3238.
- Hutter, F.; Hoos, H. H. & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In Coello, C. A. C., editor, *Learning and Intelligent Optimization*, pp. 507--523, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Ikehara, S. (2008). A novel method of bone marrow transplantation (bmt) for intractable autoimmune diseases. *Journal of Autoimmunity*, 30(3):108 – 115. The Use of Bone Marrow Transplantation to Treat Autoimmune Disease.
- Jalalian, A.; Mashohor, S. B. T.; Mahmud, H. R.; Saripan, M. I. B.; Ramli, A. R. B. & Karasfi, B. (2013). Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review. *Clinical Imaging*, 37(3):420–426.
- Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345--383.

- Kandel, E. R.; Schwartz, J. H.; Jessell, T. M.; Siegelbaum, S. A. & Hudspeth, A. (2000). *Principles of neural science*, volume 4. McGraw-hill New York, New York, NY, USA.
- Kastaniotis, D.; Fotopoulou, F.; Theodorakopoulos, I.; Economou, G. & Fotopoulos, S. (2017). Hep-2 cell classification with vector of hierarchically aggregated residuals. *Pattern Recognition*, 65:47 – 57.
- Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*.
- Krizhevsky, A. & Hinton, G. (2009). Learning multiple layers of features from tiny images.
- Krizhevsky, A.; Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L. & Weinberger, K. Q., editores, *Advances in Neural Information Processing Systems 25*, pp. 1097--1105. Curran Associates, Inc.
- LeCun, Y.; Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436-444.
- Lecun, Y.; Bottou, L.; Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Lei, H.; Han, T.; Zhou, F.; Yu, Z.; Qin, J.; Elazab, A. & Lei, B. (2018). A deeply supervised residual network for hep-2 cell classification via cross-modal transfer learning. *Pattern Recognition*, 79:290 – 302.
- Lippmann, R. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(2):4–22.
- Loddo, A.; Di Ruberto, C. & Kocher, M. (2018). Recent advances of malaria parasites detection systems based on mathematical morphology. *Sensors*, 18(2):513.
- Lovell, B. C.; Percannella, G.; Vento, M. & Wiliem, A. (2014). Performance evaluation of indirect immunofluorescence image analysis systems. In *International Conference on Pattern Recognition (ICPR)*.
- Mackay, I. & Rose, N. (2013). *The Autoimmune Diseases*. Elsevier Science.
- Manivannan, S.; Li, W.; Akbar, S.; Wang, R.; Zhang, J. & McKenna, S. J. (2016). An automated pattern recognition system for classifying indirect immunofluorescence images of hep-2 cells and specimens. *Pattern Recognition*, 51:12 – 26.
- McClelland, J. L.; Rumelhart, D. E. & PDP Research Group, C., editores (1986). *Parallel Distributed Processing: Explorations in the Microstructure, Vol. 2: Psychological and Biological Models*. MIT Press, Cambridge, MA, USA.

- Meroni, P. L. & Schur, P. H. (2010). Ana screening: an old test with new recommendations. *Annals of the Rheumatic Diseases*, 69(8):1420--1422.
- Moccia, S.; Vanone, G. O.; Momi, E. D.; Laborai, A.; Guastini, L.; Peretti, G. & Mattos, L. S. (2018). Learning-based classification of informative laryngoscopic frames. *Computer Methods and Programs in Biomedicine*, 158:21 – 30.
- Moore, A. E.; Sabachewsky, L. & Toolan, H. W. (1955). Culture characteristics of four permanent lines of human cancer cells. *Cancer Research*, 15(9):598--602.
- Nazaré, T. S.; da Costa, G. B. P.; Contato, W. A. & Ponti, M. (2018). Deep convolutional neural networks and noisy images. In Mendoza, M. & Velastín, S., editores, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 416--424, Cham. Springer International Publishing.
- Negnevitsky, M. (2011). *Artificial Intelligence*. Addison-Wesley Publishing Company, USA, 3rd edição.
- Neoh, S. C.; Srisukkham, W.; Zhang, L.; Todryk, S.; Greystoke, B.; Lim, C. P.; Hossain, M. A. & Aslam, N. (2015). An intelligent decision support system for leukaemia diagnosis using microscopic blood images. *Scientific reports*, 5:14938.
- Nosaka, R. & Fukui, K. (2014). Hep-2 cell classification using rotation invariant co-occurrence among local binary patterns. *Pattern Recognition*, 47(7):2428 – 2436.
- Perner, P.; Perner, H. & Müller, B. (2002). Mining knowledge for hep-2 cell image classification. *Artificial Intelligence in Medicine*, 26(1–2):161 – 173. Medical Data Mining and Knowledge Discovery.
- Phan, H. T. H.; Kumar, A.; Kim, J. & Feng, D. (2016). Transfer learning of a convolutional neural network for hep-2 cell image classification. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pp. 1208–1211.
- Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1 – 17.
- Ponti, M. A.; Ribeiro, L. S. F.; Nazare, T. S.; Bui, T. & Collomosse, J. (2017). Everything you wanted to know about deep learning for computer vision but were afraid to ask. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pp. 17–41.
- Rioux, J. D. & Abbas, A. K. (2005). Paths to understanding the genetic basis of autoimmune disease. *Nature*, 435(7042):584--589.
- Rodrigues, L. F.; Naldi, M. C. & Mari, J. F. (2016a). Morphological analysis and classification of erythrocytes in microscopy images. In *XII Workshop de Visão Computacional*, Campo Grande, MS, Brazil. WVC.

- Rodrigues, L. F.; Naldi, M. C. & Mari, J. F. (2017). Exploiting convolutional neural networks and preprocessing techniques for hep-2 cell classification in immunofluorescence images. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 170–177.
- Rodrigues, L. F.; Silva, J. H.; Gondim, P. H. C. C. & Mari, J. F. (2016b). Leukocytes classification in microscopy images for acute lymphoblastic leukemia identification. In *XII Workshop de Visão Computacional*, Campo Grande, MS, Brazil. WVC.
- Rosenblatt, F. (1962). Principles of neurodynamics.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C. & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211--252.
- Russell, S. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edição.
- Sack, U.; Knoechner, S.; Warschkau, H.; Pigla, U.; Emmrich, F. & Kamprad, M. (2003). Computer-assisted classification of hep-2 immunofluorescence patterns in autoimmune diagnostics. *Autoimmunity Reviews*, 2(5):298 – 304.
- Scherer, D.; Müller, A. & Behnke, S. (2010). *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*, pp. 92--101. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Scotti, F. (2005). Automatic morphological analysis for acute leukemia identification in peripheral blood microscope images. In *2005 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, volume 2005, pp. 96--101, Messian, Italy. IEEE.
- Sharif Razavian, A.; Azizpour, H.; Sullivan, J. & Carlsson, S. (2014). Cnn features off-the-shelf: An astounding baseline for recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Shen, L.; Lin, J.; Wu, S. & Yu, S. (2014). Hep-2 image classification using intensity order pooling based features and bag of words. *Pattern Recognition*, 47(7):2419 – 2427.
- Shin, H.-c.; Roth, H. R.; Gao, M.; Lu, L.; Xu, Z.; Nogue, I.; Yao, J.; Mollura, D. & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE transactions on medical imaging*, 35(5):1285--98.
- Simonyan, K. & Zisserman, A. (2014a). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

- Simonyan, K. & Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. Technical report, abs/1409.1556.
- Solomon, D. H.; Kavanaugh, A. J. & Schur, P. H. (2002). Evidence-based guidelines for the use of immunologic tests: antinuclear antibody testing. *Arthritis Care & Research*, 47(4):434–444.
- Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I. & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.
- Stoklasa, R.; Majtner, T. & Svoboda, D. (2014). Efficient k-nn based hep-2 cells classifier. *Pattern Recognition*, 47(7):2409 – 2418.
- Sutskever, I.; Martens, J.; Dahl, G. & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1139–III–1147. JMLR.org.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J. & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.
- Tan, P.-N.; Steinbach, M. & Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Telea, A. C. (2014). *Data Visualization: Principles and Practice, Second Edition*. A. K. Peters, Ltd., Natick, MA, USA, 2nd edição.
- Tieleman, T. & Hinton, G. (2012). Lecture 6.5- rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.
- Tozzoli, R.; Bonaguri, C.; Melegari, A.; Antico, A.; Bassetti, D. & Bizzaro, N. (2013). Current state of diagnostic technologies in the autoimmunology laboratory. *Clinical Chemistry and Laboratory Medicine*, 51(1):129–138.
- Wiik, A. S.; Høier-Madsen, M.; Forslid, J.; Charles, P. & Meyrowitsch, J. (2010). Antinuclear antibodies: A contemporary nomenclature using hep-2 cells. *Journal of Autoimmunity*, 35(3):276 – 290. Special Issue - In Honour of Professor Haralampos M. Moutsopoulos.
- Wu, Q.; Merchant, F. & Castleman, K. (2008). *Microscope Image Processing*. Academic Press, Inc., Orlando, FL, USA, 1st edição.

# Apêndice A

## Lista de Publicações

A seguir são listados os artigos publicados e submetidos decorrentes deste trabalho.

**RODRIGUES, L. F.;** NALDI, M. C.; MARI, J. F. *Morphological analysis and classification of erythrocytes in microscopy images*. In: Workshop de Visão Computacional, Campo Grande, 2016. p. 69-74.

**RODRIGUES, L. F.;** SILVA, J.H.; GONDIM, P.H.C.C; MARI, J. F. *Leukocytes classification in microscopy images for acute lymphoblastic leukemia identification*. In: Workshop de Visão Computacional, Campo Grande, 2016. p. 334-338.

Esses artigos, apesar de não serem de classificação de células *HEp-2*, foram publicados no início do curso de Mestrado e representam o primeiro contato com processamento e classificação de imagens biomédicas. A ideia deste projeto de Mestrado, registrado no segundo semestre de 2017, foi fortemente influenciada pelos resultados alcançados por esses dois trabalhos anteriores.

**RODRIGUES, L. F.;** NALDI, M.C.; MARI, J. F. *Classificação de células HEp-2 utilizando redes neurais convolucionais*. In: Simpósio de Integração Acadêmica - SIA, 2017, Rio Paranaíba. Do Lógico ao Abstrato: A Ciência no Cotidiano, 2017.

Nesse resumo publicado nos anais do Simpósio de Integração Acadêmica (SIA) promovido pela Universidade Federal de Viçosa, foi realizada a divulgação desse projeto de mestrado e dos resultados iniciais para a comunidade científica local. Além disso, o trabalho recebeu prêmio de menção honrosa sendo considerado o melhor trabalho de pós-graduação apresentado na modalidade oral do SIA/2017.

**RODRIGUES, L. F.;** NALDI, M. C.; MARI, J. F. *Exploiting Convolutional Neural Networks and preprocessing techniques for HEP-2 cell classification in immunofluorescence images*. In: Graphics, Patterns and Images (SIBGRAPI), 2017 30th SIBGRAPI Conference on. IEEE, 2017. p. 170-177.

Nesse artigo são apresentados os primeiros resultados obtidos ao avaliar o impacto de seis diferentes técnicas de pré-processamento de imagens e aumento do conjunto de dados baseado em rotações. Três arquiteturas de *CNNs* foram usadas: LeNet-5, AlexNet e GoogLeNet. Foi o primeiro trabalho a utilizar as arquiteturas AlexNet e GoogLeNet para classificar células *HEP-2*. Os resultados apontados nesse artigo, foram melhorados e são descritos no Capítulo 3.

**RODRIGUES, L. F.;** NALDI, M. C.; MARI, J. F. *HEP-2 Cell Image Classification Based on Convolutional Neural Networks*. In: Workshop of Computer Vision (WVC), Natal, 2017, p. 13-18.

Uma extensão ao trabalho submetido no SIBGRAPI (2017) foi realizada nesse artigo. Além de avaliar três arquiteturas de *CNNs* e usar seis diferentes estratégias de pré-processamento, esse artigo adotou o uso de validação cruzada com *k-fold* e alguns hiperparâmetros das *CNNs* foram empiricamente ajustados.

**RODRIGUES, L. F.;** NALDI, M. C.; MARI, J. F. *Classification of HEP-2 cells in immunofluorescence images comparing multiple Convolutional Neural Networks and preprocessing techniques*.

O artigo publicado no SIBGRAPI (2017) foi convidado para uma versão estendida na *Special Issue on Feature representations for medical images and activity understanding* do “*Journal of Visual Communication and Image Representation*”. Essa extensão foi submetida em maio de 2018 e compreende a comparação entre cinco arquiteturas de *CNNs*: LeNet-5, AlexNet, Inception-V3 (GoogLeNet), VGG-16 e ResNet-50 usando seis diferentes estratégias de pré-processamento, e validação cruzada com *k-fold*. Além disso, analisou-se o impacto da otimização de hiperparâmetros usando o algoritmo *Tree of Parzen Estimators* e avaliação de estratégias de treinamento completo e *fine-tuning*, a fim de estudar a influência das seis diferentes abordagens de pré-processamento em cada uma das cinco arquiteturas de *CNNs*.

FARIA, L.C.; **RODRIGUES, L. F.**; MARI, J. F. *Cell classification using handcrafted features and bag of visual words*. In: Workshop de Visão Computacional, Ilhéus, 2018.

No final desse curso de Mestrado realizou-se contribuições neste artigo que, apesar de não considerar técnicas de aprendizado profundo, utiliza o mesmo conjunto de dados avaliado nessa dissertação. Neste artigo, imagens de células *Hep-2*, eritrócitos e leucócitos são classificadas utilizando extração manual de características e *Bag of Visual Words*, tendo como principal contribuição a combinação simples e eficiente dos descritores SIFT e SURF, empilhados em uma única matriz. Este trabalho recebeu o *Best Paper Award* do Workshop de Visão Computacional 2018.