CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO

Matheus Moraes Machado

UM ESTUDO COMPARATIVO DE MÉTODOS PARA AUMENTO DE DADOS PARA CLASSIFICAÇÃO DE LOGOMARCAS EM IMAGENS

Timóteo

2019

Matheus Moraes Machado

UM ESTUDO COMPARATIVO DE MÉTODOS PARA AUMENTO DE DADOS PARA CLASSIFICAÇÃO DE LOGOMARCAS EM IMAGENS

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Me. Aléssio Miranda Júnior

Timóteo

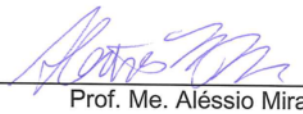
2019

Matheus Moraes Machado

UM ESTUDO COMPARATIVO DE MÉTODOS PARA AUMENTO DE DADOS PARA CLASSIFICAÇÃO DE LOGOMARCAS EM IMAGENS

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Trabalho aprovado. Timóteo, 6 de dezembro de 2019:



Prof. Me. Aléssio Miranda Júnior
Orientador

Prof. Me. Douglas Nunes de Oliveira
Professor Convidado

Prof. Me. Odilon Corrêa da Silva
Professor Convidado

Timóteo
2019

Aos meus pais
que sempre me apoiaram e não mediram esforços para que eu chegasse até aqui.

Agradecimentos

Agradeço primeiramente a Deus, por me conceder saúde e sabedoria para enfrentar as dificuldades e alcançar os meus objetivos.

Aos meus pais Divina e Antônio, agradeço pelo apoio, incentivo e contribuição para a minha formação e crescimento pessoal. Aos meus familiares, agradeço os momentos de alegria e presença em minha vida!

Ao orientador deste trabalho, Me. Aléssio Miranda Júnior, que se dispôs a me orientar, mesmo existindo o obstáculo da distância; Pela atenção a mim dispensada, boa vontade em ensinar e paciência no desenvolvimento deste trabalho.

A todos os professores do CEFET-MG, que contribuíram direta ou indiretamente para o meu aprendizado e para minha formação, em especial, a professora Deisymar pelo empenho e orientações no desenvolvimento deste trabalho.

Aos meus amigos no Brasil e também fora dele, que sempre estiveram presentes, embora fisicamente estavam distantes.

Agradeço ao Dr. Adrian Rosebrock, autor de livros relacionados a visão computacional e *deep learning*, além de ser o criador do site *PyImageSearch.com*, que dissemina conhecimentos a respeito de visão computacional para toda a comunidade.

E por fim, a todos os amigos encontrei no CEFET-MG, pela união e trabalho em equipe. Pela ajuda mútua nos momentos difíceis da nossa jornada. Foram muitos anos de amizade que sei que ainda irão durar ainda por anos...

"Uma imagem vale mais do que mil palavras."
Confúcio

Resumo

As redes sociais se tornaram uma maneira ampla para se compartilhar informações, como imagens. Essas informações podem ser muito úteis para empresas que precisam saber das opiniões dos seus clientes e seus interesses em suas marcas. Para este propósito, um classificador de imagens pode ser de grande utilidade. Atualmente, a estratégia de se usar aprendizagem profunda com redes neurais convolucionais para classificação de imagens é a que mais tem sido empregada. Entretando, ela requer uma grande quantidade de amostras para treinamento, o que nem sempre está acessível. Para resolver este problema, podemos usar *data augmentation*, uma técnica de *regularization* que expande o *dataset* original para aumentar a acurácia da classificação. Sendo assim, este presente estudo busca responder a seguinte questão: É possível alcançar uma boa acurácia na classificação de logomarcas com *data augmentation* nos casos em que grande quantidade de dados para *dataset* não estão disponíveis? A necessidade das empresas de saberem as opiniões de seus clientes e carência de se explorar os métodos de *data augmentation* justificam esta pesquisa. Seu objetivo é comparar o uso de métodos de *data augmentation* no contexto de classificação de logos de marcas e verificar se há melhora na acurácia possuindo um conjunto limitado de dados no *dataset*. Sete métodos (*flip*, *crop*, *rotation*, *gaussian filter*, *gaussian noise*, *scale* e *shear*) foram selecionados com base em trabalhos anteriores. Duas redes neurais convolucionais foram utilizadas para criação do classificador, *AlexNet* e *SmallerVGGNet*, desta forma seria possível verificar se certas combinações obteriam bons resultados em diferentes arquiteturas de redes neurais convolucionais. Oito combinações destes métodos mostraram que *flip*, *crop*, *rotation* e *scale* são uma boa combinação para o contexto de classificação de logomarcas, melhorando a acurácia em até 22.09% usando a *AlexNet* e 12.57% utilizando a *SmallerVGGNet*. Enquanto outros bons resultados foram a combinação de *flip*, *crop* e *rotation*; a combinação de *flip*, *crop*, *rotation* e *gaussian noise*; e o uso de todos os métodos. Como continuação da pesquisa, pretende-se utilizar outras técnicas de *regularization* juntamente com *data augmentation* para melhorar ainda mais a eficácia da classificação.

Palavras-chave: *data augmentation*, classificação de imagens, logomarca, rede neural convolucional, visão computacional.

Abstract

Social networks have become a widely used way to share information, including images. This information can be useful for companies, which need to know their customers' opinions and their interest in their brands. For these purposes, an image classifier has a great utility. Currently, deep learning using convolutional neural networks are heavily employed for image classification. However, they require a large number of training image samples, which are not always accessible. In order to solve this problem, we can use data augmentation, which is a regularization technique that is based on expand the original dataset to increase classification accuracy and avoid overfitting. This present work aims to compare the use of different data augmentation methods for brand logo classification. For tests, seven methods (flip, crop, rotation, gaussian filter, gaussian noise, scale, and shear) were selected based on previous studies. Two convolutional neural networks were used, AlexNet and SmallerVGGNet, this way we could see if some combinations lead to better results in different artificial network architectures. Ten different combinations of those methods show that the combination of flip, crop, rotation, and scale is a more effective combination for a brand logo classifier in the both convolutional neural networks used, improving accuracy by 22.09% using AlexNet and 12.57% using SmallerVGGNet. Other good results are found with the combination of all seven methods; flip, crop, rotation, plus gaussian noise; and the combination of flip, crop, and rotation. Further research is intended to use another regularization technique along with data augmentation to improve even more the accuracy and reduce overfitting.

Keywords: data augmentation, image classification, brand logo, convolutional neural network, computer vision.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CEO	Chief Executive Officer
DL	Deep Learning
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
MPE	Micro e Pequenas Empresas
DA	Data Augmentation
PCA	Principal Component Analysis

Sumário

1 Introdução

Desde a criação do *Twitter* em 2006, as redes sociais se tornaram um dos principais meios para se compartilhar informações (YAN; WU; ZHENG, 2013). Também conhecidas como microblogs, esses meios evoluíram consideravelmente nos últimos anos e junto delas o aumento do número dos seus usuários. Com o uso da *internet*, as pessoas podem utilizar as redes sociais em qualquer lugar e a qualquer momento. Todos esses fatores permitem que as redes sociais gerem uma enorme quantidade de mídia mediante a publicação de textos e imagens dos seus usuários (HOSSAIN; ALHAMID; MUHAMMAD, 2018; WANG et al., 2016).

As empresas divulgam produtos e supervisionam o desenvolvimento de suas marcas em redes sociais. As mídias postadas por usuários transmitindo sentimentos sobre produtos oferecem às empresas uma boa oportunidade para entenderem as opiniões dos seus clientes (WANG et al., 2016). Os comentários positivos e negativos desses consumidores nas redes sociais têm um importante valor para as empresas, que precisam saber a aceitação de suas marcas e dos seus produtos. Mesmo usuários individuais são capazes de dar *feedbacks* que podem ser importantes para as empresas realizarem decisões de compra e selecionarem estratégias futuras para suas marcas e produtos. O rápido aumento da quantidade dessas informações nas redes sociais exige o desenvolvimento de técnicas eficazes de rastreamento de marcas para coleta de dados e análise de conteúdo de mídia (GAO et al., 2014).

Figura 1 – Uma opinião de um cliente

Fonte: Adaptado de Twitter (2016).

Figura 2 – O Primeiro Logo da *Apple*

Fonte: Creative (2018).

Segundo uma pesquisa do *IBM Institute for Business Value*, se aproximar dos clientes é a principal prioridade dos CEOs, concluindo que as companhias podem usar as plataformas sociais para mineração de dados com intuito de monitorar suas marcas, desta forma elas são capazes de entender quais são os pensamentos dos clientes, permitindo melhorar seus produtos, serviços e suas experiências com eles (BAIRD; PARASNIS, 2011).

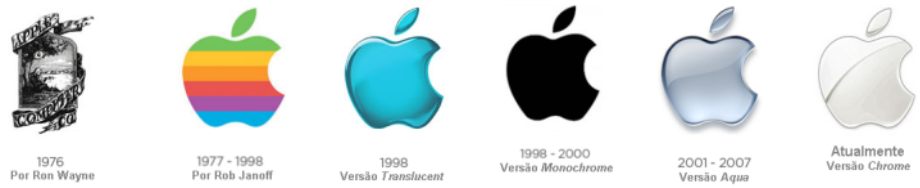
Uma pesquisa feita pela *Brandwatch*, uma empresa de monitoramento de mídia social, indica que atualmente 3,2 bilhões de imagens são compartilhadas por dia. Além disso, 96% das pessoas que falam sobre uma marca online não seguem os perfis destas marcas em redes sociais. Sugerindo que as empresas precisam buscar por novas estratégias para saber as opiniões de seus clientes (SMITH, 2019).

A Figura 1 exemplifica uma opinião de um cliente sobre um produto da empresa americana *Starbucks*. Ele não se refere à marca textualmente ou mesmo utilizando alguma *hashtag*, mas através da imagem postada é possível saber que ele está referenciando à empresa.

Outro contexto que as empresas necessitam para identificar suas marcas em imagens são para protegê-las de usos indevidos e não autorizados (LIU; ZHANG, 2013). A *internet* mudou significativamente o campo de proteção delas em todo o mundo. Todos os dias um novo infrator pode aparecer e desaparecer sem ser percebido, tornando extremamente difícil para os proprietários evitarem a confusão do consumidor e garantir que a reputação de suas marcas permaneçam ilesas (BORUCKA, 2017).

Uma estratégia que as empresas podem abordar é utilizar classificadores de imagens para detectar aquelas que se relacionam com suas marcas. A classificação de imagens tem como objetivo classificar uma imagem com um rótulo (*label*) dado um conjunto de classes

Figura 3 – A Evolução do Logo da *Apple*



Fonte: Adaptado de Creative (2018).

predefinidas (Rizvi et al., 2016).

Nos últimos anos, redes neurais artificiais como *Deep Nets* possibilitaram um enorme avanço nesta área. Entretanto, elas possuem limitações e desafios a serem enfrentados. Um dos problemas do seu uso é que elas dependem de grande quantidade de dados para treinamento, conhecidos como *datasets*, devido a isso, o seu uso fica restrito para situações onde essa grande quantidade de dados esteja disponível, o que nem sempre acontece (Yuille; Liu, 2018). Em muitas aplicações apenas uma pequena quantidade de dados para treino estão disponíveis (Ismail Fawaz et al., 2018).

A marca de uma empresa muda conforme o tempo. Como um exemplo, temos o primeiro logo da empresa americana *Apple*, mostrado na Figura 2, e sua evolução até os dias atuais, representado na Figura 3. Conforme Silva (2017), esta evolução está ligada a alterações históricas que acontecem por questão de aprimoramento da composição da identidade visual da empresa e está conectada ao fato de representar as próprias adaptações pelas quais as instituições enfrentam ao longo do tempo. Sendo assim, novas mudanças impactam na dificuldade de encontrar dados disponíveis para serem utilizados.

Adicionar imagens modificadas ao *dataset* antes do seu treinamento numa rede neural podem gerar melhores resultados nos casos em que grande quantidade de dados para treino não esteja disponível. *Image data augmentation* ou apenas *data augmentation* (DA) é o processo para gerar dados sintéticos a partir de imagens reais usando um método específico e, desta forma, agregar mais dados ao *dataset*, contribuindo para uma melhor acurácia no treinamento em uma rede neural artificial (Ismail Fawaz et al., 2018).

Na literatura, autores utilizam diferentes abordagens para geração de novas amostras para os *datasets* antes de serem treinados em uma rede neural artificial (Rizvi et al., 2016; Hu et al., 2014; Molina et al., 2013; Ismail Fawaz et al., 2018; HAN; LIU; FAN, 2018). Escolher os métodos adequados para *data augmentation* tem sido muito usado em aprendizagem profunda e selecionar as estratégias adequadas para este tipo de abordagem é tão importante quanto selecionar o modelo de rede neural (GOODFELLOW; BENGIO; COURVILLE, 2016).

As técnicas mais comuns utilizadas na literatura aplicam *affine transformations*, que são transformações conhecidas como *translation*, *zoom*, *flip*, *shear*, *mirror* e *color perturbation*. Seus usos na literatura, com outras técnicas, mostraram um aumento na eficiência de

um classificador, melhorando sua acurácia e evitando *overfitting* [1] (HAN; LIU; FAN, 2018). Em 2017, uma pesquisa comparativa de métodos para *data augmentation* aplicados a uma rede neural convolucional (CNN) foi realizada mostrando o impacto de alguns métodos em classificação de imagens (Shijie et al., 2017). Os resultados mostraram que os métodos para aumento de dados resultaram em uma melhor acurácia nos modelos treinados em seu contexto, com exceção da adição de ruídos às imagens. Em (HUSSAIN et al., 2017), outra pesquisa para técnicas de *data augmentation* obteve resultados piores com a adição de ruídos, entretanto, melhorou a acurácia utilizando *flip* e transformações gaussianas.

Conforme visto até então na literatura, a utilização de DA ainda é muito empírica e as técnicas adequadas variam de acordo com o contexto em que são aplicadas.

Portanto, este trabalho busca responder a seguinte questão:

É possível alcançar uma boa acurácia nos resultados na classificação de logomarcas utilizando estratégias de *data augmentation* nos casos em que grande quantidade de dados para *dataset* não estão disponíveis?

1.1 Justificativa

As empresas necessitam de saber opiniões de clientes como uma estratégia de *marketing*, para melhorar seu relacionamento com eles. Além disso, elas precisam proteger suas marcas de usos não autorizados.

Os métodos até então usados para *data augmentation* utilizando aprendizado de máquina podem ser mais explorados visto que são necessários em grandes casos. Esta abordagem pode melhorar com significância os resultados quando não é possível adquirir muitos dados para serem treinados em uma rede neural.

1.2 Problema

Com o uso das tecnologias atuais, uma empresa precisa ter uma base de dados para reconhecer se uma postagem na rede social é útil a ela ou se sua marca está sendo indevidamente usada. Sobre isso, o processo de automatização começa ao receber uma base de dados corrente de postagens e localizar nestas exibições de marcas. Entretanto, não se sabe qual tecnologia ou quais passos seguir.

Um classificador de imagens utilizando redes neurais artificiais resolveria este problema. No entanto, devido aos problemas ainda existentes no seu uso, novas abordagens precisam ser usadas para melhorar a qualidade dos seus resultados. Uma delas, é o uso de *data augmentation*, que pode aumentar a acurácia de um classificador e evitar *overfitting*.

[1] *Overfitting* ocorre quando a rede neural fica limitada aos dados que estão sendo treinados e não se generaliza o suficiente para outros dados (ROSEBROCK, 2019).

1.3 Objetivos

O objetivo geral deste trabalho é empregar métodos de *data augmentation* em classificação de imagens e comparar estas técnicas para verificar se há melhora na acurácia dos resultados no contexto de classificação de marcas que contenha um conjunto limitado de dados em seu *dataset*.

Entre os objetivos específicos estão:

1. Propor a utilização de métodos baseados na literatura para *data augmentation*;
2. Propor a utilização de um modelo de rede neural que atenda a classificação de logomarcas;
3. Verificar estes métodos através da elaboração de um protótipo funcional e verificar se há um aperfeiçoamento na acurácia dos resultados e quais destes métodos obtiveram um melhor resultado.

Para responder às questões propostas e aos objetivos apresentados, pretende-se desenvolver um protótipo funcional utilizando ferramentas disponíveis atualmente e utilizá-lo para um comparativo entre os métodos de DA propostos. Os procedimentos metodológicos seguidos são apresentados no Capítulo 2.

1.4 Estrutura do trabalho

Nas seções anteriores, foi feita uma breve introdução sobre a necessidade de criação de um classificador de marcas em imagens e foi proposto o uso de uma rede neural artificial para criação deste classificador. Porém, um dos problemas que podemos ter para utilização dele seria não obter dados suficientes para treinamento. Logo, foi colocado em pauta o uso de técnicas de DA, que precisam ser mais exploradas e poderiam resolver o problema levantado. Apresentou-se também a justificativa o objetivo para a realização desta pesquisa.

Em seguida, o Capítulo 2 apresentará os procedimentos metodológicos por meio dos quais este trabalho se desenvolve.

As bases teóricas são apresentadas no Capítulo 3, que inclui os principais conceitos para melhor entendimento do trabalho.

O Capítulo 4 apresenta algumas ferramentas que disponibilizam funcionalidades capazes de criar ou utilizar um classificador de logomarcas e também são descritos pesquisas relacionadas à *data augmentation*, justamente com as técnicas mais exploradas.

No Capítulo 5 é apresentado a implementação do protótipo funcional de aplicação de métodos de *data augmentation* para classificação de logomarcas com os métodos de *data augmentation* selecionados.

O Capítulo 6 apresenta os resultados através da aplicação dos métodos selecionados e uma comparação das acurácias geradas pela utilização deles, mostrando sua aplicabilidade no contexto de classificação de logomarcas.

Por fim, no Capítulo 7 são apresentadas as considerações finais do trabalho com o cronograma para a sua continuação.

2 Procedimentos Metodológicos

Conforme Wazlawick (2009), esta presente pesquisa é caracterizada como exploratória por apresentar estudos de casos.

Os procedimentos metodológicos foram organizados nas seguintes etapas:

1. Pesquisar por possíveis soluções que permitem a criação ou utilização de um classificador de logomarcas;
2. Pesquisar por trabalhos científicos que abordam estudos sobre métodos de *data augmentation*;
3. Propor casos de estudo com a utilização de métodos de *data augmentation* para elaboração de um classificador de marcas em imagens;
4. Implementar um protótipo funcional utilizando os casos de estudo propostos;
5. Realizar um comparativo entre os casos de estudo.

As etapas são descritas nas seguintes seções.

2.1 Busca por possíveis soluções

Foram feitas buscas por ferramentas que disponibilizam funcionalidades relevantes que atendem a criação ou utilização de um classificador de logomarcas. As buscas foram feitas pelo *Google* envolvendo as palavras-chave: *brand logo classification* (classificação de logomarcas), *image classification tools* (ferramentas de classificação de imagens), *image classifier tools* (ferramentas de classificador de imagens) e *image classifier build* (contruir classificador de imagens).

Foi encontrada uma ferramenta que já realiza o monitoramento e reconhecimento de logomarcas em conteúdos. Também foram encontrados classificadores de imagens em que é possível realizar um novo treinamento com *datasets* personalizados. Outras ferramentas, como as bibliotecas encontradas, permitem a criação de um classificador.

O capítulo 4 aborda todas as ferramentas encontradas.

2.2 Busca por trabalhos científicos

A busca por trabalhos científicos foi baseada na coleta de informações de estudos que comprovam a importância que as mídias sociais relacionadas as marcas e logomarcas de empresas possuem. Após isso, foram buscados pesquisas que apresentam métodos de *data augmentation* que, embora fossem usados em outros contextos de classificação, poderiam ser empregados para classificação de logomarcas.

Inicialmente foram filtrados por título e leitura do resumo. Foram utilizados no trabalho os artigos que apresentavam conhecimentos relevantes à área computacional. Dentre os coletados era muito forte a presença de assuntos relacionados à visão computacional, que foram utilizados no texto.

Foram escolhidas palavras-chave em inglês, relacionadas ao tema. Não foram encontrados algum termo equivalente sendo usado na literatura em português para algumas definições, como *image data augmentation*. As pesquisas foram realizadas principalmente no Portal de Periódicos da CAPES, IEEE Xplore e no *Google* Acadêmico. Foram pesquisados as seguintes palavras-chave: *customer relationship management* (gestão de relacionamento com o cliente), *brand protection* (proteção de marcas), *brand logo classification* (*classificação de logotipos de marcas*), *image classification* (classificação de imagens) e *image data augmentation* (aumento de dados em imagens).

No geral, as buscas por estes termos tiveram os resultados relacionados à importância da utilização das mídias sociais para o *marketing* das empresas, para proteção de suas marcas e como a opinião e *feedbacks* de clientes são importantes para as companhias e seus CEOs.

Foram encontrados estudos comparativos de diferentes métodos de *data augmentation*, mostrando combinações entre eles e como certas combinações levam a um melhor aprimoramento do classificador.

Livros que abordam visão computacional e aprendizado de máquina também foram empregados, principalmente para definições de termos técnicos.

2.3 Propor a utilização de métodos de *data augmentation*

Revisando a literatura, foram encontrados métodos foram encontrados métodos de *data augmentation* e suas aplicações em classificação de imagens.

Alguns estudos revisados fazem comparações entre alguns métodos e a melhora que eles geram na acurácia da classificação.

A partir desta revisão, foi possível selecionar os métodos e combinações que apresentaram os melhores resultados. Também foram selecionados trabalhos que apresentaram resultados inferiores no contexto em que foram empregados, mas que poderiam gerar um resultado diferente no contexto de classificação de logomarcas.

2.4 Implementação de um protótipo funcional

Foi feita a implementação de um protótipo funcional com objetivo de fazer testes de forma padronizada utilizando métodos de *data augmentation*. O protótipo gera *datasets* expandidos com a utilização destes métodos e é capaz de treinar uma rede neural utilizando as novas bases de dados expandidas para criação de um classificador de imagens.

O classificador foi criado utilizando dois modelos de redes neurais convolucionais escolhidos.

Foi-se estabelecido casos de estudo, onde cada um utiliza uma combinação diferente dos métodos de *data augmentation* selecionados para serem aplicados. Desta forma, cada caso obteve um *dataset* único.

2.5 Comparação entre os casos de estudo

Foi realizado o treinamento de cada um dos *datasets* dos casos de estudo e também do *dataset* original, onde não foi aplicado nenhuma técnica de *data augmentation*.

O método de comparação entre a eficiência de cada treinamento foi realizada através da acurácia, que é quantidade de acertos dividida pelo total de imagens de teste. Através disso, é possível verificar que tipo de abordagens de *data augmentation* possuem uma melhor aplicabilidade para o contexto de classificação de logomarcas.

3 Referencial Teórico

Este capítulo aborda os conceitos fundamentais para melhor compreensão da aplicabilidade e do contexto ao qual se engloba este trabalho.

3.1 Marcas

Uma empresa ou um produto pode ser representado por uma marca. Uma marca pode ser composta pela combinação de nome, letras, números, um símbolo, uma assinatura, uma forma, uma logo, um slogan, uma cor ou uma letra. O nome, entretanto, é o principal elemento e nunca deve ser alterado. Todos os outros elementos podem mudar ao longo do tempo, como as logomarcas (CLIFTON; SIMMONS; AHMAD, 2004).

Logos, também chamados de logotipos ou logomarcas, se consistem de uma mistura de texto e símbolos gráficos. Embora pareça apenas mais uma classe de objeto na classificação de imagens, este tipo de classificação pode ser desafiador em imagens reais. Uma mesma logo quando aparece em diferentes cenários pode ter um aspecto diferente em cada um deles, devido a mudanças no tamanho, rotação, iluminação ou em sua rigidez (HOI et al., 2015).

3.2 Visão Computacional

Como humanos, tendo a capacidade da visão, percebemos o mundo tridimensional a nossa volta de maneira relativamente fácil. Psicólogos perceptivos passaram décadas tentando entender como o sistema visual se comporta e ainda que eles possam imaginar ilusões de ótica para desmembrar alguns de seus princípios, uma solução completa para esse quebra-cabeça permanece vago. Pesquisadores de visão computacional têm desenvolvido, paralelamente, técnicas matemáticas para recuperar a forma tridimensional e a aparência de objetos em imagens. Atualmente podemos até, com sucesso moderado, localizar e nomear as pessoas que se encontram em uma fotografia usando uma combinação de reconhecimento e detecção de rosto, roupas e cabelo. Entretanto, apesar de todos esses avanços, o sonho de ter um computador que interprete uma imagem no mesmo nível que uma criança de dois anos permanece ilusório (SZELISKI, 2010).

A visão computacional busca descrever o mundo que vemos em uma ou mais imagens e reconstruir suas propriedades, como forma, iluminação e distribuição de cores. Ela é utilizada em uma grande variedade de aplicações no mundo real como: (SZELISKI, 2010)

- *Optical Character Recognition* (OCR): reconhecimento de caracteres (como números e letras) em imagens;
- Construção de modelos 3D (fotogrametria): construção automatizada de modelos 3D a partir de fotografias aéreas utilizadas em sistemas como *Bing Maps*;

- Vigilância: monitoramento de intrusos, análise de tráfego rodoviário e monitoramento em piscinas para vítimas de afogamento;
- Reconhecimento de impressões digitais e biometria: para autenticação de acesso automático, bem como aplicações forenses.

Nos últimos oitenta anos, a história da visão computacional pode ser resumida, de acordo com seus principais acontecimentos, da seguinte maneira: (SZELISKI, 2010)

- Na década de 70, o que distinguiu o antigo processamento de imagens com o nascimento da visão computacional foi o desejo de reconstruir estruturas tridimensionais do mundo real utilizando imagens e com isso entender uma cena por completo.
- Nos anos 80, a atenção acadêmica foi voltada para técnicas matemáticas mais sofisticadas para realizar análises quantitativas de imagens e cenas. Pesquisas sobre melhores detectores de bordas e contornos foram bastante realizadas neste período.
- Na década de 90, muitos dos tópicos explorados anteriormente continuaram a ser pesquisados, entretanto, alguns deles se tornaram mais populares. Técnicas de aprendizado utilizando estatística começaram a aparecer, como na análise *eigenface* para reconhecimento facial.
- No começo do novo século, pesquisas começaram a ser realizadas na área de reconhecimento de objetos, baseadas em extrair características de imagens.

3.3 Aprendizagem Profunda

Aprendizagem profunda, aprendizado profundo ou, em Inglês, *deep learning*, é uma subárea de aprendizado de máquina, em Inglês *machine learning*, que por sua vez é uma subárea de inteligência artificial. A Figura 4 demonstra a relação entre estas áreas através de um diagrama de Venn (ROSEBROCK, 2019).

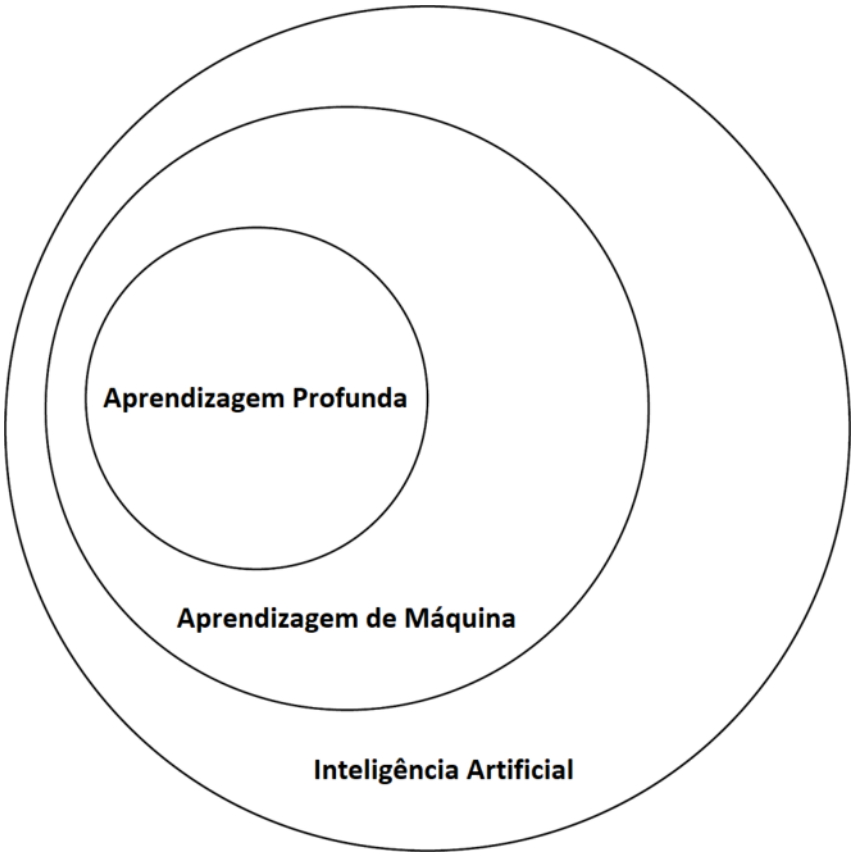
A história das redes neurais artificiais e aprendizagem profunda é longa e confusa. Aprendizagem profunda existe desde 1940 através de vários nomes como *cybernetics*, *connectionism* e o nome mais familiar atualmente *Artificial Neural Networks* (ANNs) (ROSEBROCK, 2019).

Os métodos de *deep learning* (DL) são atualmente o estado-da-arte em muitos problemas possíveis de se resolver via aprendizado de máquina, em particular, problemas de classificação. Nos últimos anos, estes métodos têm revolucionado diversas áreas de aprendizado de máquina, em especial a visão computacional (Vieira et al., 2017).

3.3.1 Rede Neural Artificial

Uma rede neural artifical (ANN) é uma estrutura que simula o funcionamento de um conjunto de neurônios. A ANN mais simples é aquela composta de apenas um neurônio, chamada perceptron. Ele possui diversas entradas com seus respectivos pesos e um valor limite,

Figura 4 – Diagrama de Venn de Aprendizagem Profunda.



Fonte: Adaptado de Rosebrock (2019).

utilizado para decidir se o perceptron será ativado, ou seja, se sua saída será igual a 1. Sendo que o valor de sua saída pode ser 0 ou 1 (TANAKA, 2018).

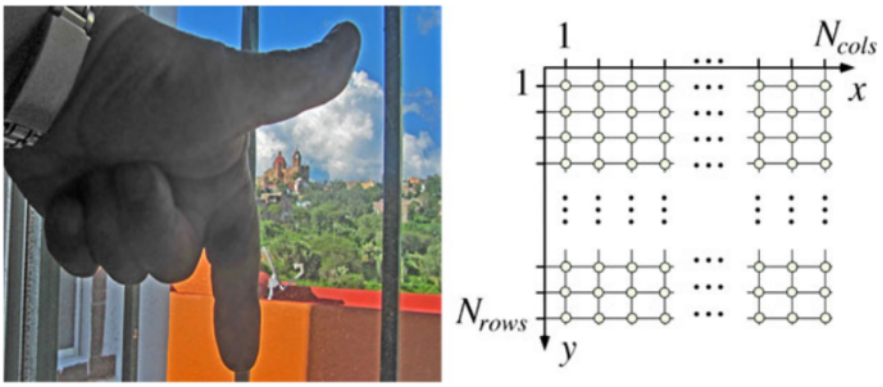
Formamos uma ANN quando juntamos muitos perceptrons. Para treinar um perceptron, comparamos o resultado obtido com o resultado esperado e mudamos os pesos das entradas para minimizar o erro obtido (TANAKA, 2018).

3.3.2 Rede Neural Convolucional

Durante a ascensão da DL, a extração de características e classificador foram integrados a uma estrutura de aprendizado que supera o método tradicional de dificuldades de seleção de recursos. A ideia de DL é descobrir múltiplos níveis de representação, com a esperança de que recursos de alto nível representem uma semântica mais abstrata dos dados. Um dos principais ingredientes da aprendizagem profunda na classificação de imagens é o uso de arquiteturas convolucionais (Guo et al., 2017).

Uma rede neural convolucional, em Inglês convolutional neural network (CNN), é uma classe de rede neural utilizada para processamento e análise de imagens. Ela foi proposta pelo artigo (CUN, 1986) do cientista Yann LeCun, onde foi criado uma arquitetura capaz de reconhecer dígitos manuscritos com precisão de 99,2%. Essa arquitetura foi inspirada em uma pesquisa de 1968, feita por David Hunter Hubel e Torsten Wiesel sobre o funcionamento do cór-

Figura 5 – O sistema de coordenadas *left-hand*



O polegar representa o eixo x e o indicador o eixo y. Fonte: Klette (2014).

tex visual dos mamíferos. A pesquisa sugere que mamíferos percebem visualmente o mundo de forma hierárquica, através de camadas de clusters de neurônios. Quando vemos algo, clusters são ativados hierarquicamente, e cada um detecta um conjunto de atributos sobre o que foi visto. A arquitetura da CNN simula *clusters* de neurônios para detectar atributos daquilo que foi visto, organizados hierarquicamente e de forma abstrata o suficiente para generalizar independentemente de tamanho, posição ou rotação (TANAKA, 2018).

Em muitas aplicações, CNNs são consideradas os mais poderosos classificadores de imagens e são responsáveis por impulsionar o estado-da-arte para frente em subcampos de visão computacional que alavancam a aprendizagem de máquina (ROSEBROCK, 2019).

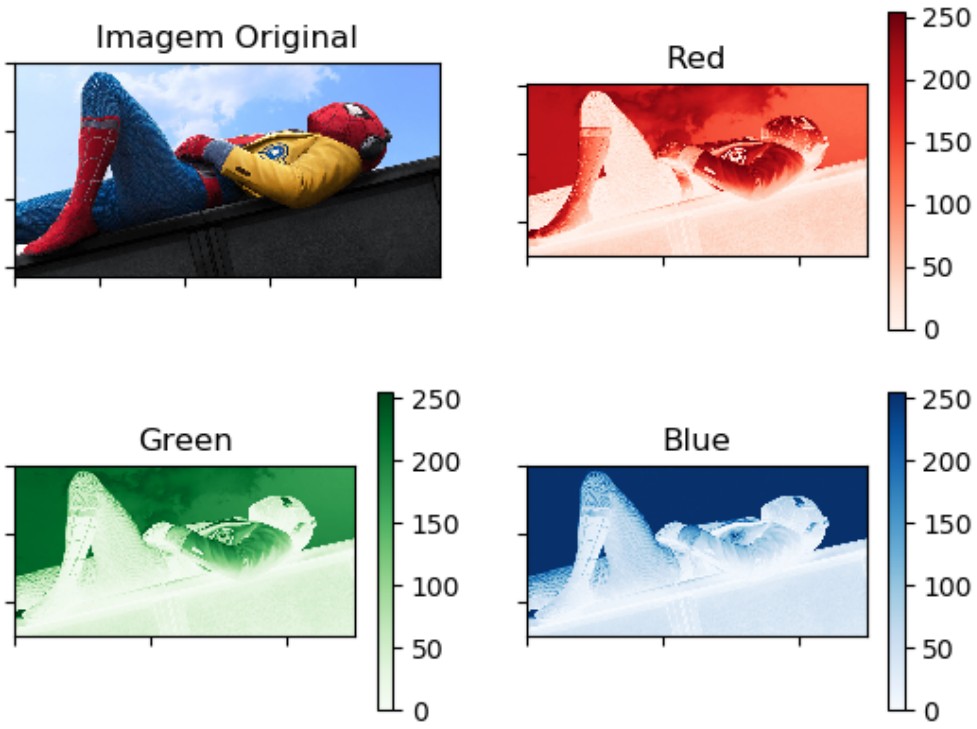
A camada CONV é o bloco de construção principal de uma CNN. Os parâmetros desta camada consistem em um conjunto de filtros K que podem ser aprendidos (*kernels*), onde cada filtro tem uma largura e uma altura e são quase sempre quadrados A função de LOSS (ROSEBROCK, 2019).

No nível mais básico, uma função de LOSS (perda) quantifica quão "bom"ou "ruim"um determinado preditor está na classificação dos pontos de dados de entrada em um conjunto de dados. Quanto menor a perda, melhor o trabalho do classificador em modelar o relacionamento entre os dados de entrada e os rótulos das classes de saída (embora exista um ponto em que podemos superajustar nosso modelo - modelando o treinando os dados com muita atenção, nosso modelo perde a capacidade de generalizar). Por outro lado, quanto maior a nossa perda, mais trabalho precisa ser feito para aumentar a precisão de classificação (ROSEBROCK, 2019).

3.4 Imagens Digitais

Pixels são elementos atômicos de uma imagem. Uma imagem digital no domínio espacial é composta por uma matriz retangular de pixels (x, y, u), onde cada pixel u está em uma localização $(x, y) \in Z$ (KLETTE, 2014). Formalmente, podemos definir uma imagem I como um conjunto retangular da seguinte maneira:

Figura 6 – Uma imagem no modelo RGB.



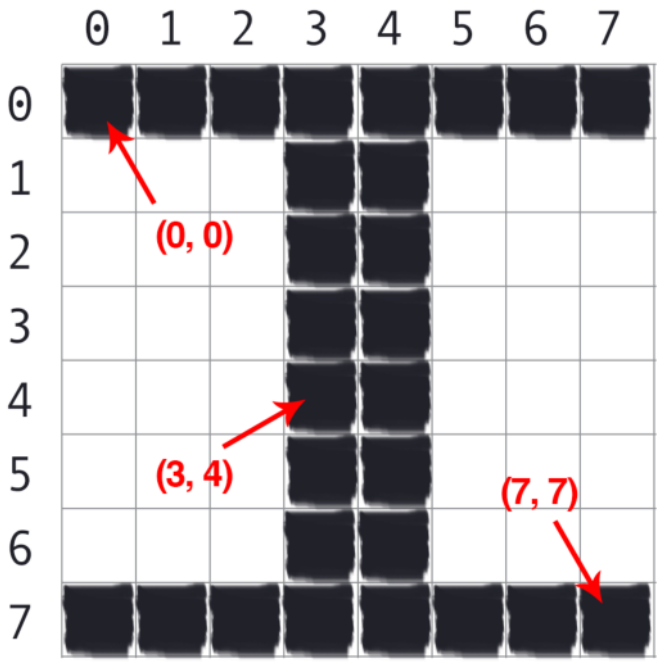
Cada imagem *Red*, *Green* e *Blue* representa um canal e juntas formam a imagem original, sendo esta a junção dos três canais. Fonte: Imagem original obtidada pelo System (2017) e adaptada pelo autor utilizando um algoritmo em *Python*.

$$\Omega = \{(x, y) : 1 \leq x \leq N_{cols} \wedge 1 \leq y \leq N_{rows}\} \tag{3.1}$$

Onde N_{cols} é o número de colunas da imagem e N_{rows} é o número de linhas. É assumido um sistema de coordenadas *left-hand* como mostrado na Figura 5. A Figura 7 exemplifica como este modelo é utilizado.

Uma imagem vetorial possui mais de um canal ou banda. O modelo mais comum utilizado atualmente para imagens coloridas é o modelo RGB, que possui três canais, onde cada letra representa um deles. A Figura 6 mostra um exemplo de como uma imagem RGB é formada por seus canais. Neste modelo, cada pixel é composto por três valores inteiros que variam de 0 a 255, totalizando 256 valores possíveis para cada uma das bandas (KLETTE, 2014).

Figura 7 – Imagem de "I".



Na imagem, cada pixel é representado por um quadrado e está na nas coordenadas em vermelho, onde o primeiro elemento é a coluna e o segundo elemento é a linha onde o pixel se localiza. Fonte: Rosebrock (2019).

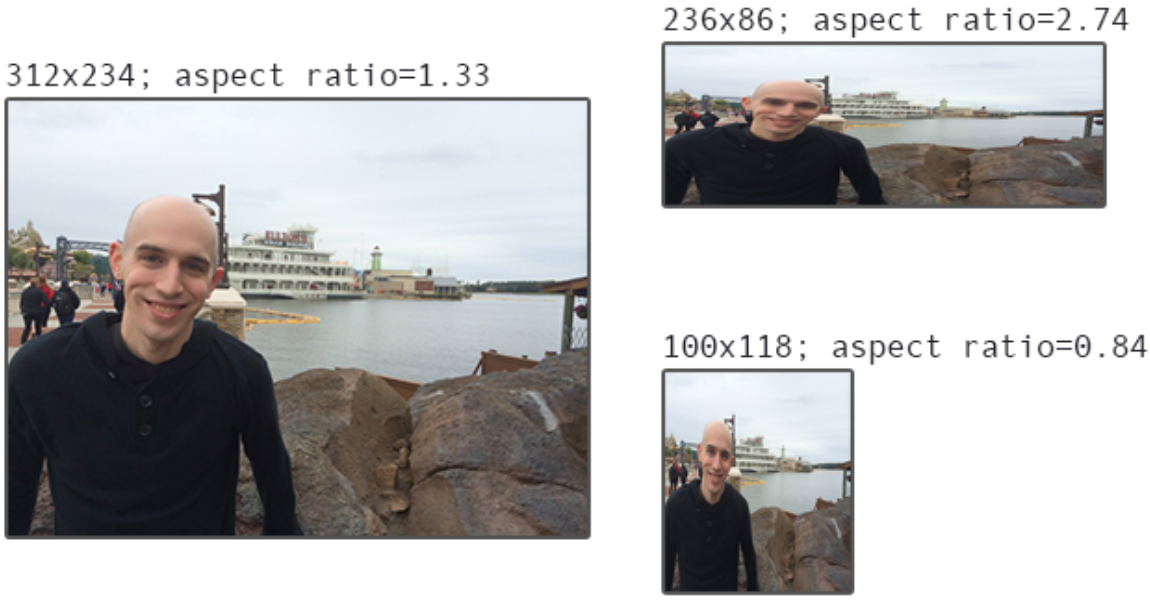
3.5 Redimensionamento e Proporção

Redimensionamento (em Inglês *scaling* ou *resizing*) é o processo de aumentar ou diminuir o tamanho de uma imagem em termos de largura e altura. Ao se redimensionar uma imagem, é importante considerar sua proporção, em Inglês conhecido como *aspect ratio*. (ROSEBROCK, 2019).

A proporção de uma imagem é a relação entre sua largura e altura. Se ela for ignorada, ao se redimensionar uma imagem, pode levar a uma compressão ou distorção dela, como exemplificado na Figura 8. No exemplo, as imagens da direita foram distorcidas e esmagadas por não preservarem o *aspect ratio*. Para evitar este comportamento, basta redimensionar a largura e a altura em quantidades iguais (ROSEBROCK, 2019).

De um ponto de vista estritamente estético, você quase sempre quer garantir que a proporção da imagem seja mantida quando for redimensionada, mas essa diretriz nem sempre é o caso da aprendizagem profunda. A maioria das redes neurais e Redes Neurais Convolucionais aplicadas à tarefa de classificação de imagens assumem uma entrada de tamanho fixo, o que significa que as dimensões de todas as imagens que você passa pela rede neural devem ser as mesmas. Escolhas comuns para tamanhos de imagem de largura e altura inseridas em Redes Neurais Convolucionais incluem 32x32, 64x64, 224x224, 227x227, 256x256 e 299x299 (ROSEBROCK, 2019).

Figura 8 – Proporções de Imagens.



Exemplo de imagens com diferentes proporções (*aspect ratio*). Do lado equerdo, nós temos a imagem original e do lado direito duas imagens que foram distorcidas após um redimensionamento. Fonte: Rosebrock (2019).

3.6 Processamento de Imagens

Processamento de imagens é o processo de mapear uma imagem original I em uma imagem J, tipicamente usado para melhorar a qualidade da imagem original ou por algum propósito complexo de visão computacional (KLETTE, 2014).

Conforme Kotkar e Gharde (2013), o aprimoramento de imagem, conhecido pelo termo *contrast enhancement*, é uma das técnicas mais significativas no processamento digital de imagens. Ela consiste em um conjunto de estratégias que buscam melhorar a qualidade de uma imagem ou convertê-la para uma forma mais adequada que permita ser analisada por um humano ou uma máquina. Os métodos de *contrast enhancement* são divididos em duas principais categorias: métodos no domínio espacial e no domínio da frequência[1].

É possível transformar uma imagem do domínio espacial para o domínio da frequência, aplicar filtro(s) nela neste formato e em seguida retorná-la para o domínio espacial (KLETTE, 2014).

3.7 Classificação de Imagens

Classificação de imagens é a tarefa de classificar uma imagem com um rótulo (*label*) dado um conjunto de classes predefinidas (ROSEBROCK, 2019).

[1] Enquanto o domínio espacial consiste na imagem no plano em si, no domínio da frequência baseia-se em aplicar uma transformada de Fourier na imagem no domínio espacial (Kotkar; Gharde, 2013).

Figura 9 – Imagem de um Cachorro.

Fonte: Rosebrock (2019).

Isso significa que seu objetivo é analisar uma dada imagem de entrada e retornar um rótulo que a categorize. Por exemplo, assumindo um conjunto de classes que incluem gato, cachorro e panda, quando apresentamos a imagem da Figura 9 para o nosso classificador, ele deveria retornar a classe cachorro (ROSEBROCK, 2019).

Caso uma determinada imagem possa ser associada a mais de uma classe ou a nenhuma delas, é possível utilizar classificação *multilabel* (*multilabel classification*) (Sun; Lee; Wang, 2016). No exemplo anterior, utilizando este tipo de classificão, o classificador poderia retornar 95% para a classe cachorro, 4% para gato e 1% para panda (ROSEBROCK, 2019).

3.8 *Data Augmentation*

Regularização é qualquer modificação que fazemos no algoritmo de treinamento que pretende reduzir seu erro de generalização, mas não o seu erro de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016, p. 224, traduzido pelo autor). Em resumo, a regularização ou *regularization* busca reduzir o erro de teste em um treinamento de uma rede neural e, talvez, para atingir este objetivo, pode ocorrer um aumento do erro de treinamento (ROSEBROCK, 2019).

Conforme Rosebrock (2019), dois tipos comuns de regularização são:

1. Modificar a arquitetura da rede neural.
2. Aumentar os dados que estão sendo passados para o treinamento da rede neural.

Em muitos problemas de classificação, os dados disponíveis para compor o *dataset* não são suficientes. Uma abordagem muito utilizada na literatura é a aplicação de *data augmentation* (DA), que consiste na segunda forma de regularização exemplificada (Fawzi et al., 2016).

DA consiste em transformar as amostras disponíveis em novas, cada imagem "aumentada"pode ser considerada uma nova imagem que a rede neural não havia visto antes. Desta forma, a rede neural está constantemente sendo apresentada a novas amostras para treinamento. Embora seja sempre melhor obter amostras de imagens "naturais", *data augmentation* pode ser utilizado para superar as limitações existentes de um pequeno *dataset* (Fawzi et al., 2016; ROSEBROCK, 2019).

Infelizmente, DA é uma arte e envolve muitas escolhas manuais. Escolhas erradas podem gerar amostras que não impactam na acurácia do classificador ou reduzem-a (Fawzi et al., 2016).

O objetivo ao se aplicar DA é aumentar a generalização do modelo treinado da rede neural. O seu uso pode reduzir drasticamente o *overfitting* de uma rede neural (ROSEBROCK, 2019). Na literatura, os métodos mais comuns utilizam *affine transformations*. Alguns exemplos são *translation*, *zoom*, *flip*, *shear*, *mirror* e *color perturbation* (HAN; LIU; FAN, 2018).

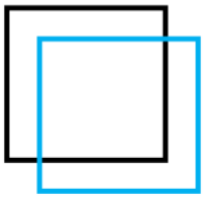
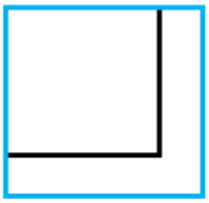
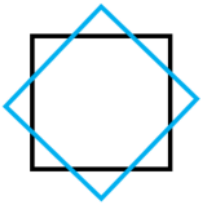
3.8.1 *Affine Transformation* (Transformação Afim)

Transformações afins é um método de mapeamento liner que preserva pontos, linhas retas e planos. Os conjuntos de linhas paralelas permanecem paralelos após esta transformação (HAN; LIU; FAN, 2018).

Geralmente essas transformações são usadas para corrigir distorções ou deformações que ocorrem com câmeras que não possuem um ângulo ideal (HAN; LIU; FAN, 2018). Para DA, este tipo de transformações pode ser muito útil para mostrar outros pontos de vista que uma determinada imagem pode ter.

A Tabela 1 apresenta as matrizes de transformações de alguns métodos que utilizam transformação afim. t_x especifica o deslocamento ao longo do eixo x e t_y ao longo do eixo y. s_x especifica o fator de escala ao longo do eixo x e s_y ao longo do eixo y. sh_x especifica o fator de cisalhamento ao longo do eixo x e sh_y ao longo do eixo y (AFFINE. . . , 2019).

Tabela 1 – Matrizes da Transformação Afim Affine... (2019).

Transformação Afim	Exemplo	Matriz de Transformação
Translation		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$
Scale		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Shear		$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Rotation		$\begin{bmatrix} cos(q) & sin(q) & 0 \\ sin(q) & cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

4 Estado Da Arte

Este capítulo apresenta algumas ferramentas e *softwares* encontrados no mercado que disponibilizam algumas funcionalidades desejáveis que permitem o desenvolvimento de um classificador de imagens que permita proporcionar a fidelização ou engajamento de clientes através das redes sociais.

Algumas destas ferramentas foram utilizadas para criação do protótipo funcional proposto no Capítulo 5.

Além disso, foi descrito as pesquisas atuais na área de *data augmentation* e suas contribuições para a visão computacional.

4.1 Algumas Ferramentas Disponíveis Atualmente

1. *LogoGrab*

   LogoGrab (2019) é uma empresa focada em identificar logo de marcas em imagens e vídeos. Além de identificar que um determinado logo se encontra em uma imagem ou video, ela também mostra sua localização.

   É possível integrar a API *LogoGrab Logo Detection* com alguma plataforma que, por exemplo, no contexto deste estudo seria de extrair imagens de redes sociais. É possível obter uma versão demonstrativa, porém é uma ferramenta paga.

2. *Watson Visual Recognition*

   Watson Visual Recognition é um serviço oferecido pela empresa IBM que pode ser usado pela *IBM Cloud*. Existe uma versão gratuita chamada *Lite* e a paga, *Standard* Watson... (2019).

3. *Vision AI*

   Vision AI é uma ferramenta da Google, também paga, que permite extrair informações de imagens utilizando um modelo já treinado ou treinando seu próprio modelo, porém ainda se encontra em produção e possui apenas uma versão Beta Vision... (2019).

4. *Keras*

   Keras é uma API de rede neural de alto nível escrita em *Python* e capaz de executar em cima do *KerasTensorflow*, CNTK ou *Theano*. Ele foi desenvolvido com foco em possibilitar uma rápida implementação. Partindo da ideia para o resultado com o mínimo possível de atraso faz com que as pesquisas sejam realizadas muito mais facilmente (KERAS, 2019).

5. PyTorch

PyTorch é uma outra biblioteca *open source* capaz de criar modelos de *machine learning*. É possível criar modelos de redes neurais próprios para classificação de imagens utilizando *Python* PyTorch (2019) .

4.1.1 Comparativo Entre as Ferramentas

Não foi encontrado uma ferramenta atual em alto nível flexível o suficiente para o experimento, principalmente integrada com redes sociais e que fosse capaz de extrair postagens delas que contenham uma determinada marca customizada. As ferramentas gratuitas *open source*, que foram encontradas, permitem a implementação de um *software* final capaz de solucionar este problema.

Desta forma, o mais adequado para empresas, ainda mais que sejam MPEs, seria utilizar o conhecimento técnico com as ferramentas gratuitas *open source*.

Uma desvantagem de se usar as ferramentas *open source* é que elas, às vezes, não possuem um produto final adequado. Entretanto, é uma vantagem enorme utilizá-las, pois pode ser uma forma barata adaptada à solução que uma empresa precisa.

No caso de MPEs ou quando uma empresa muda a logo de sua marca, a dificuldade aumenta em se obter dados para usar nessas ferramentas. Há então a necessidade de utilizar *data augmentation* para solucionar o problema da falta de dados necessários.

4.2 Pesquisas Relacionadas a *Data Augmentation*

Em 2017, Jia Shijie, Wang Ping, Jia Peiyi e Hu Siping fizeram uma pesquisa sobre o impacto de técnicas de *data augmentation* com Redes Neurais Convolucionais. A pesquisa buscou responder três questões sobre classificação de imagens:

1. Quais são os impactos da utilização de diferentes métodos de *data augmentation* na sua *performance*?
2. Para diferentes quantidades de imagens para treinamento, quais são as diferenças dos impactos que as técnicas de *data augmentation* causam no seu comportamento?
3. Quais são as diferenças dos impactos na utilização de apenas um método e a combinações deles para uma mesma quantidade de dados no *dataset*?

Para abordar o problema, o trabalho utilizou uma CNN chamada AlexNet pré-treinada. Os *datasets* utilizados foram CIFAR10 e ImageNet (que possui 10 classes). O treinamento abordou diferentes quantidades de imagens (pequena, média e grande).

A quantidade pequena possuía 2000 imagens (200 para cada uma das classes), a média tinha um total de 10000 amostras (1000 para cada categoria) e a grande 50000 (5000 para cada classe).

Foram utilizados sete métodos: *Flip*, *Crop*, *Shift*, *PCA jitter*, *Color jitter*, *Noise* e *Rotate*. Teste foram realizados com estes métodos individualmente e com combinações entre eles.

Tabela 2 – Resultados sobre aplicações de alguns métodos de *data augmentation* em imagens médicas (HUSSAIN et al., 2017).

Método	Acurácia do Treinamento	Acurácia do Teste
Noise	0.625	0.660
Gaussian Filter	0.870	0.881
Jitter	0.832	0.813
Scale	0.887	0.874
Powers	0.661	0.737
Rotate	0.884	0.880
Shear	0.891	0.879
Flip	0.891	0.842

Conforme Shijie et al. (2017), os métodos são definidos da sequinte forma:

- *Flip*: Inverte a imagem horizontalmente.
- *Rotate*: Rotaciona a imagem em uma direção aleatória.
- *Crop*: Corta uma parte da imagem e redimensiona-a para a resolução usada na CNN, caso seja necessário.
- *Shift*: Desloca a imagem para esquerda ou para a direita, a distância destas deslocações podem ser definidas manualmente para mudar a localização da imagem.
- *Noise*: Adiciona ruídos aleatórios nos canais RGB em cada pixel da imagem.
- *Color jitter*: Muda aleatóriamente fatores das imagens como cor, saturação e contraste.
- *PCA jitter*: Aplica PCA[1] na imagem para obter um componente principal que é adicionado à imagem original com uma perturbação gaussiana de (0, 0.1) para geração de uma nova imagem.

A pesquisa mostrou que a utilização separada destes métodos de *data augmentation* aumentaram a acurácia do treinamento, apenas *Noise* não obteve um bom resultado. Além disso, quanto maior o aumento de amostras utilizando estas técnicas, maior foi a acurácia dos resultados.

Outra abordagem realizada no mesmo trabalho foi a junção das técnicas, concluindo que a junção de dois ou três métodos alcançaram melhores resultados que as técnicas utilizadas individualmente.

No trabalho de Hussain et al. (2017) alguns métodos foram aplicados em outro contexto. Desta vez em imagens médicas. Os resultados são mostrados na Tabela 2 . Podemos ver que, da mesma forma, o estudo não obteve bons resultados com a utilização de *Noise*.

[1] Principal Component Analysis (PCA) é um algoritmo matemático que reduz a dimensionalidade dos dados, mantendo a maior parte da variação no conjunto de dados 1 (RINGNÉR, 2008).

No fim de 2017, uma pesquisa para criação de um classificador de imagens para detectar alcoolismo utilizou apenas 100 amostras de dados e através de *data augmentation* conseguiu aumentar o *dataset* para 13100 imagens. O método utilizado juntamente com uma CNN conseguiu obter melhores resultados que três abordagens já utilizadas na literatura. Os métodos de *data augmentation* utilizados neste trabalho foram *Rotate*, *Gamma Correction*, *Noise* e *Translate* (RINGNÉR, 2008).

Outros métodos também são abordados na literatura, como *Slice Window* que foi inspirado no *Crop*. Esta técnica de transformação de dados pode, até certo ponto, garantir que a imagem recortada ainda contenha as mesmas informações que a imagem original. Uma outra técnica baseada em *DTW Barycentric Averaging* (DBA) foi utilizada em séries temporais e demonstrou que o uso de *data augmentation* pode drasticamente melhorar a acurária em modelos de *deep learning* (Ismail Fawaz et al., 2018).

5 Desenvolvimento

Este capítulo apresenta os detalhes do protótipo funcional desenvolvido. Primeiro, é apresentada a arquitetura das redes neurais convolucionais utilizadas. Em seguida, os métodos de *data augmentation*. E, por fim, a arquitetura do protótipo funcional e como ele pode ser utilizado.

O protótipo funcional foi implementado em *Python* e encontra-se disponível no *GitHub* em Brand... (2019).

Para desenvolver a CNN, foi utilizado *Keras*, uma API de rede neural de alto nível escrita em *Python* e capaz de executar em cima do *Tensorflow*, CNTK ou *Theano*. Ela foi desenvolvida com foco em possibilitar uma rápida implementação. Partindo da ideia para o resultado com o mínimo possível de atraso faz com que as pesquisas sejam realizadas muito mais facilmente (KERAS, 2019). Para o protótipo funcional proposto, foi utilizado o *Tensorflow*.

Para implementar os métodos de *data augmentation*, foram utilizadas três bibliotecas principais: *OpenCV*, *Scikit-image* e *NumPy*.

OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de visão computacional e aprendizado de máquina de código aberto (OPENCV, 2019); *Scikit-image* é uma coleção de algoritmos para processamento de imagens (SCIKIT-IMAGE, 2019); enquanto *NumPy* é um pacote fundamental para computação científica (NUMPY, 2019).

As versões da linguagem de programação e das bibliotecas utilizadas são descritas abaixo:

- *Python*: 3.7.3;
- *Keras*: 2.2.4 (Usando TensorFlow em *backend*);
- *Tensorflow*: 2.0.0-alpha0;
- *OpenCV*: 4.0.1;
- *Scikit-image*: 0.14.3.

5.1 Redes Neurais Convolucionais

Para a comparação dos métodos de *data augmentation* foram utilizadas duas arquiteturas de CNNs, *SmallerVGGNet* e *AlexNet*. A seguir são apresentadas as características de cada uma.

5.1.1 *SmallerVGGNet*

A *SmallerVGGNet* foi proposta por Rosebrock (2018). Ela é uma adaptação mais compacta da rede *VGGNet* proposta pelo trabalho de Simonyan e Zisserman (SIMONYAN; ZISSERMAN, 2015).

As arquiteturas *VGGNet* possuem as seguintes características (ROSEBROCK, 2018):

1. Utilizam apenas camadas convolucionais 3×3 empilhadas umas sobre as outras em profundidade crescente;
2. Reduzem o tamanho do volume utilizando *max pooling* (um processo para reduzir as dimensões das imagens);
3. Camadas totalmente conectadas no final da rede antes da utilização de um classificador *softmax*.

Este modelo possui uma entrada com dimensões de 96×96. Desta forma, as imagens para treinamento devem ser redimensionadas para este tamanho. Como cada imagem possui três canais (RGB), a quantidade de neurônios de entrada é 27.648 ($96 \times 96 \times 3$).

5.1.2 *AlexNet*

A *AlexNet* foi proposta por Krizhevsky, Sutskever e Hinton (2012). Esta arquitetura ganhou um torneio de classificação em 2012 e é extremamente responsável pelo atual avanço de aprendizagem profunda aplicada à classificação de imagens.

Segundo documentado no artigo, a imagem de entrada na rede neural deveria ser $224 \times 224 \times 3$, entretanto, conforme explicado por Rosebrock (2019) utilizando estas dimensões de entrada, não é possível obter um número inteiro que satisfaça a equação de uma camada CONV em uma rede neural convolucional:

$$((W - F + 2 \times P)/S) + 1 \tag{5.1}$$

Para esta arquitetura, assumimos S como 2 e P como zero. F é o filtro a ser aplicado, que para esta arquitetura é 11×11. Aplicando esta equação para a entrada 224×224 da arquitetura da AlexNet, obtemos o seguinte resultado:

$$((224 - 11 + 2 \times 0)/4) + 1 = 54.25 \tag{5.2}$$

Para obter um valor inteiro, seria necessário utilizar uma entrada de 227×227:

$$((227 - 11 + 2 \times 0)/4) + 1 = 55 \tag{5.3}$$

5.2 Métodos de *Data Augmentation*

Para verificar quais combinações de métodos de *data augmentation* possuem melhores resultados para classificação de logomarcas, foi realizada uma comparação da acurácia e falsos positivos e negativos de cada uma das classes para 10 casos. Baseado no trabalho de Shijie et al. (2017) e Hussain et al. (2017), foram selecionados os sequintes métodos: *Flip*, *crop*, *rotation*, *gaussian filter*, *gaussian noise*, *scale* e *shear*. As aplicações destes métodos podem ser observadas na Figura 10.

Os casos de estudo propostos foram:

- Caso 1: Dataset original;
- Caso 2: *Flip* e *gaussian filter*;
- Caso 3: *Flip*, *crop* e *rotation*;
- Caso 4: *Flip*, *crop* e *scale*;
- Caso 5: *Flip*, *rotation* e *shear*;
- Caso 6: *Flip*, *crop*, *rotation* e *gaussian filter*;
- Caso 7: *Flip*, *crop*, *rotation* e *gaussian noise*;
- Caso 8: *Flip*, *crop*, *rotation* e *scale*;
- Caso 9: *Flip*, *crop*, *rotation*, *gaussian filter*, *gaussian noise*, *scale* e *shear*;
- Caso 10: Dataset original, com a mesma quantidade de imagens que o dataset do caso 8.

Os métodos e combinações foram escolhidos por terem gerado bons resultados em ambos os estudos, de Shijie et al. (2017) e Hussain et al. (2017). O *gaussian filter* foi escolhido por ter sido o pior resultado nas duas pesquisas, desta forma, é possível verificar se para o contexto deste presente estudo ele também levará a um pior resultado ou ao oposto, melhorando a acurácia.

As implementações de cada método são descritas abaixo:

- *Flip*: É executada uma inversão horizontal e vertical para cada imagem original. Para isso, são utilizadas as funções do NumPy *fliplr* (inversão da esquerda para a direita) e *flipud* (inversão de cima para baixo);
- *Crop*: Uma parte aleatória da imagem original é cortada. Este corte pode ser no lado esquerdo, direito, no topo ou embaixo da imagem e não pode exceder um terço do tamanho da imagem;
- *Rotation*: É executada uma rotação aleatória na imagem original utilizando a função *ImageDataGenerator* do *Keras*;

Figura 10 – Métodos de *Data Augmentation* utilizados

(a) Imagem Original (b) *Flip* (c) *Crop*

(d) *Rotation* (e) *Gaussian Filter* (f) *Gaussian Noise*

(g) *Scale* (h) *Shear*

Fonte: Elaborado pelo autor.

- *Gaussian filter*: Um filtro gaussiano é aplicado na imagem original utilizando a função *GaussianBlur* do OpenCV. O núcleo gaussiano utilizado foi de dimensão 11x11 e o desvio-padrão dele variando de 1 até 10 na direção X;
- *Gaussian noise*: Um ruído gaussiano aleatório é aplicado na imagem original usando a função *random_noise* do *Scikit-image*;
- *Scale*: Uma transformação afim (*affine transformation*) é aplicada na imagem original com um vator de escala variando entre 0.7 e 1.3 em X e Y. Para isso, foi utilizada a função *AffineTransform* do *Scikit-image*.

- *Shear*: Uma transformação afim (*affine transformation*) é aplicada na imagem original utilizando o fator *shear* entre 0.1 e 0.4. Para isso, foi utilizada a função *AffineTransform* do *Scikit-image*.

Para todos esses métodos, com excessão do *flip*, são geradas cinco novas imagens para cada imagem original. Para aumentar a quantidade de imagens do dataset original, de forma que obtenha a mesma quantidade de imagens que a de outro caso, foi realizada a duplicação das imagens em cada classe até atingir a quantidade de imagens selecionada.

5.3 Arquitetura do protótipo funcional

A arquitetura do protótipo funcional pode ser observada na Figura 11. A implementação das CNNs podem ser encontradas dentro do diretório *cnn*.

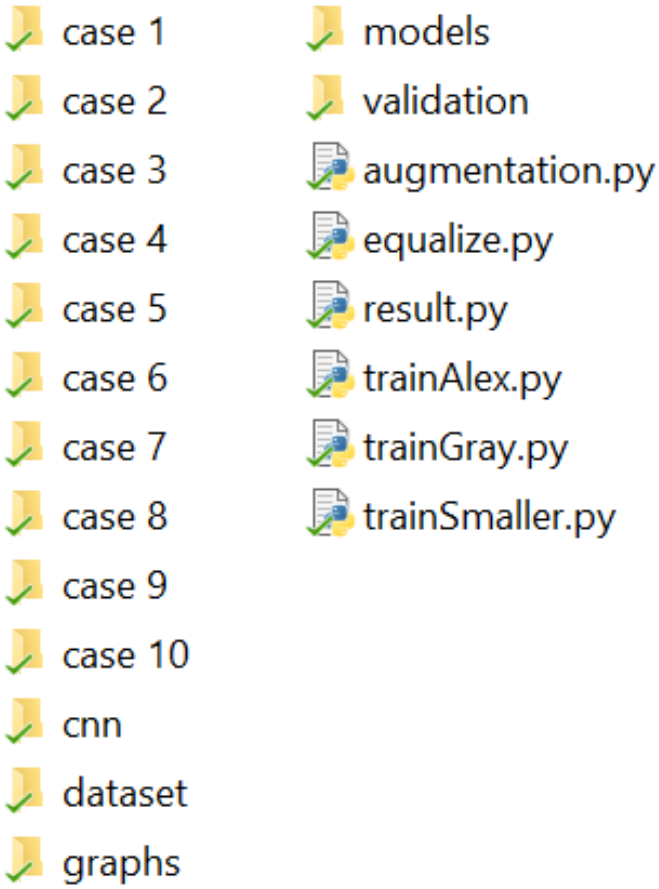
Em *dataset* temos o *dataset* original. É preciso cloná-lo e renomear para *"case 1"* para ser o nosso primeiro caso de estudo. Ao executar o arquivo *equalize.py*, indicamos qual pasta de *dataset* queremos expandir para uma nova pasta. Neste estudo foi realizada uma expansão na pasta *case 10*. As imagens são replicadas de forma que o *dataset* obtenha uma nova quantidade. Isso foi realizado para uma comparação de treinamento com *datasets* de mesmo tamanho.

O arquivo *augmentation.py* aplica *data augmentation* para cada um dos casos descritos neste capítulo e as novas imagens são salvas em suas respectivas pastas (*case 1*, *case 2*, [...] , *case 10*).

Os arquivos *trainSmaller.py*, *trainAlex.py* e *trainGray.py* realizam o treinamento da rede neural para cada um dos *datasets* (o *dataset* original e os novos *datasets* gerados por meio de *data augmentation*) e salva os modelos com seus rótulos no diretório *models*. Os gráficos mostrando as perdas e as acurácias durante o treinamento são salvos na pasta *graphs*.

Por fim, o arquivo *result.py* carrega cada um dos modelos treinados da pasta *models* e realiza o teste usando as imagens existentes no diretório *test*.

Figura 11 – Arquitetura do protótipo funcional



Fonte: Elaborado pelo autor.

6 Experimentos

Este capítulo apresenta os experimentos realizados utilizando o protótipo funcional desenvolvido com os casos de estudo propostos.

6.1 Ambiente dos Experimentos

Os experimentos foram realizados utilizando as seguintes características de *hardware*:

- Processador AMD Ryzen 5 2500U com *Radeon Vega Mobile Gfx*, 2000 Mhz, 4 *Cores*, 8 *Logical Processors*;
- Memória Física (RAM) de 8 GB (7.64 GB usáveis).

Para o sistema operacional, foi utilizado o *Microsoft Windows* 10 *Pro* de 64 bits.

6.2 Base de dados

Foram escolhidas três marcas para realização dos testes: Coca-Cola, Lacoste e Starbucks. Duas bases de dados originais (não sintéticas) foram utilizadas, uma para treinamento e outra para validação.

A quantidade de imagens da base de treinamento para cada um dos casos de estudo, juntamente com o tamanho da matriz das imagens para cada rede neural estão descritos na Tabela 3. Para a base de validação foram selecionadas 1.050 imagens.

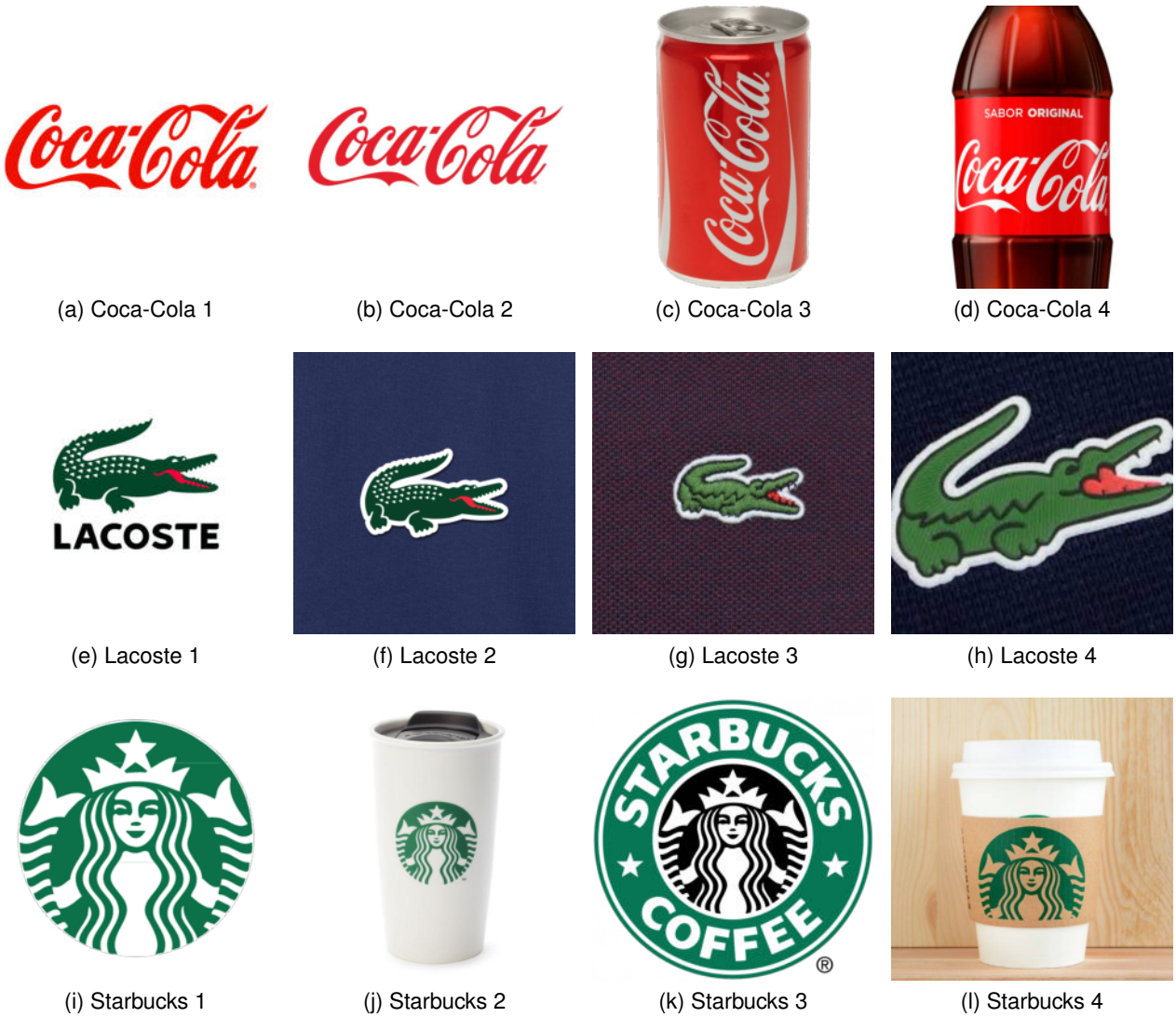
6.2.1 Base de dados para treinamento

As imagens para treinamento foram coletadas a partir do *Google* Imagens. Elas foram escolhidas de forma que representassem bem as logomarcas de cada marca e não possuíssem um tamanho muito grande. A Figura 14 mostra algumas das imagens utilizadas. Ao total, foram coletadas 96 imagens, sendo 32 para cada uma das três classes.

6.2.2 Base de dados para validação

As imagens para a validação também foram coletadas principalmente a partir do *Google* Imagens. Outras foram obtidas a partir de um *dataset* chamado WebLogo-2M. Wong et al. (2016) possui cerca de 2 milhões de imagens de 194 classes de logomarcas que foram coletadas a partir do *Twitter*. Estas imagens foram coletadas a partir de *hashtags*, logo muitas delas não contêm a logomarca na imagem. Devido a isso, foram utilizadas apenas algumas imagens da marca. Outras imagens, porém, foram fotografadas pelo autor. A Figura 13 mostra alguns exemplos de fotografias. A quantidade total de imagens obtidas foi de 1.050.

Figura 12 – Base de dados para treinamento.



(a) Coca-Cola 1 (b) Coca-Cola 2 (c) Coca-Cola 3 (d) Coca-Cola 4

(e) Lacoste 1 (f) Lacoste 2 (g) Lacoste 3 (h) Lacoste 4

(i) Starbucks 1 (j) Starbucks 2 (k) Starbucks 3 (l) Starbucks 4

Imagens coletadas para treinamento. Fonte: Imagens coletadas do *Google* Imagens.

6.3 Testes realizados na *SmallerVGGNet*

Os casos de estudo foram testados utilizando a rede neural *SmallerVGGNet* através de 500 épocas. As amostras de entrada da rede neural foram redimensionadas para o tamanho de 96×96.

Os resultados são apresentados na Tabela 4. As Figuras 15, 16, 17, 18 e 19 mostram os gráficos com as perdas e as acurácias durante o treinamento da rede neural.

6.4 Testes realizados na *AlexNet*

Os casos de estudo foram testados utilizando a rede neural *AlexNet* através de 400 épocas. As amostras de entrada da rede neural foram redimensionadas para o tamanho de 227×227.

Figura 13 – Fotografias de logomarcas

(a) Foto 1

(b) Foto 2

(c) Foto 3

(d) Foto 4

Fotografias tiradas pelo autor para serem utilizadas na validação dos modelos de rede neural treinados. Fonte: Elaborado pelo autor.

Os resultados são apresentados na Tabela 5. As Figuras 20, 26, 22, 23 e 24 mostram os gráficos com as perdas e as acurácias durante o treinamento da rede neural.

6.5 Testes realizados na *AlexNet* com imagens em escala de cinza

Para uma verificação de como ocorre uma redução de perda de validação ao se treinar imagens com apenas um canal ou banda (sem utilizar o modelo RGB), foram realizados testes

Tabela 3 – Amostras de treinamento de cada caso de estudo

Caso de Estudo	Quantidade de Imagens	Tamanho na *SmallerVGGNet*	Tamanho na *AlexNet*
1	96	20.74MB	115.94MB
2	768	165.89MB	927.52MB
3	1248	269.57MB	1507.22MB
4	1248	269.57MB	1507.22MB
5	1248	269.57MB	1507.22MB
6	1728	373.25MB	2086,92MB
7	1728	373.25MB	2086.92MB
8	1728	373.25MB	2086.92MB
9	3168	684.29MB	3826.03MB
10	1728	373.25MB	2086.92MB

Tabela 4 – Resultados dos Casos de Estudo da *SmallerVGGNet*.

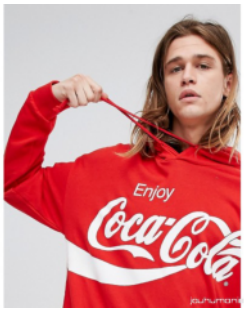
Caso	Acurácia do Treinamento	Perda	Acurácia do Teste	Perda	Tempo (dd:hh:mm:ss)
1	1	0.0018	0.7305	2.6099	00:01:31:23
2	1	6.50×10^{-4}	0.68	5.8117	00:05:56:11
3	1	3.51×10^{-6}	0.8181	2.1946	00:08:10:52
4	0.9984	0.0059	0.8076	2.4972	00:08:10:53
5	1	6.44×10^{-5}	0.8029	3.2131	00:08:07:33
6	1	7.06×10^{-4}	0.7495	5.0537	00:11:08:14
7	0.9994	0.0121	0.8095	2.6849	00:10:53:50
8	1	2.57×10^{-6}	0.8371	2.512	00:11:37:50
9	0.9991	0.007	0.781	4.3662	00:20:30:27
10	1	2.66×10^{-7}	0.7114	5.147	00:10:57:44

Tabela 5 – Resultados dos casos de estudo da *AlexNet*

Caso	Acurácia do Treinamento	Perda	Acurácia do Teste	Perda	Tempo (dd:hh:mm:ss)
1	0.9688	1.2553	0.6362	9.4334	00:05:34:28
2	0.9948	0.5416	0.7381	2.1429	00:18:21:08
3	0.9968	1.0512	0.799	2.8605	01:01:43:04
4	0.9952	0.6865	0.7771	2.4486	01:01:53:29
5	1	0.2655	0.7238	1.4029	01:03:33:16
6	0.9931	0.4242	0.721	2.8487	01:18:26:58
7	1	0.3668	0.8371	1.0632	01:17:57:32
8	0.9954	0.3532	0.8571	1.1775	01:11:23:01
9	0.9965	0.1634	0.8124	1.0159	02:13:21:08
10	0.9959	0.7742	0.7229	5.6727	01:11:48:50

Figura 14 – Base de dados para validação

(a) Coca-Cola 1



(b) Coca-Cola 2

(c) Coca-Cola 3

(d) Coca-Cola 4

(e) Lacoste 1

(f) Lacoste 2

(g) Lacoste 3

(h) Lacoste 4

(i) Starbucks 1

(j) Starbucks 2

(k) Starbucks 3

(l) Starbucks 4

Imagens coletadas para validação. Fonte: Imagens coletadas do *Google* Imagens.

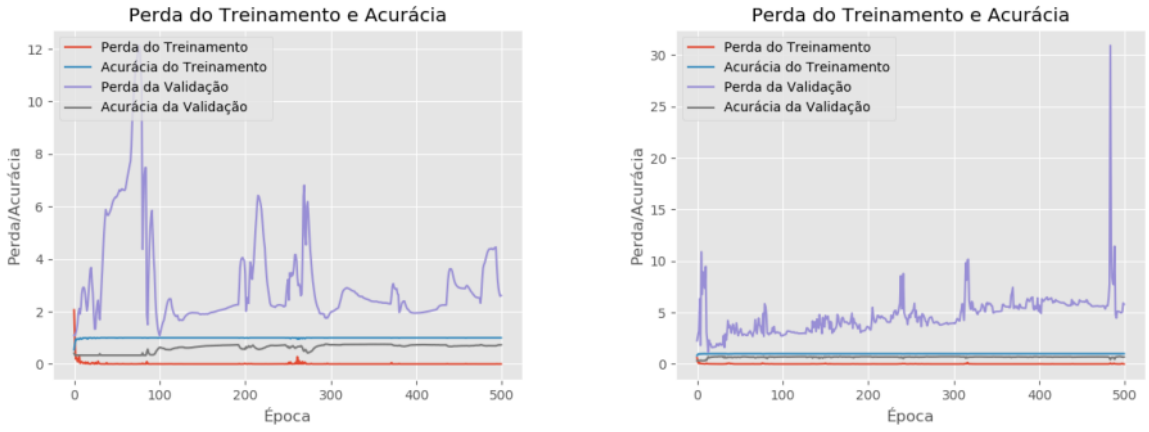
Tabela 6 – Resultados de experimentos com imagens em escala de cinza na *AlexNet*

Caso	Acurácia do Treinamento	Perda	Acurácia do Teste	Perda	Tamanho da matriz
8	0.9959	0.3458	0.781	1.3256	695.64MB
10	1	0.1283	0.6	2.8188	695.64MB

com imagens em escala de cinza utilizando o caso de estudo 10 (com o *dataset* sem o uso de *data augmentation* e o caso de estudo 8 que apresentou melhores resultados.

Os resultados são apresentados na Tabela 6. A Figura 25 mostra os gráficos com as perdas e as acurácias durante o treinamento da rede neural.

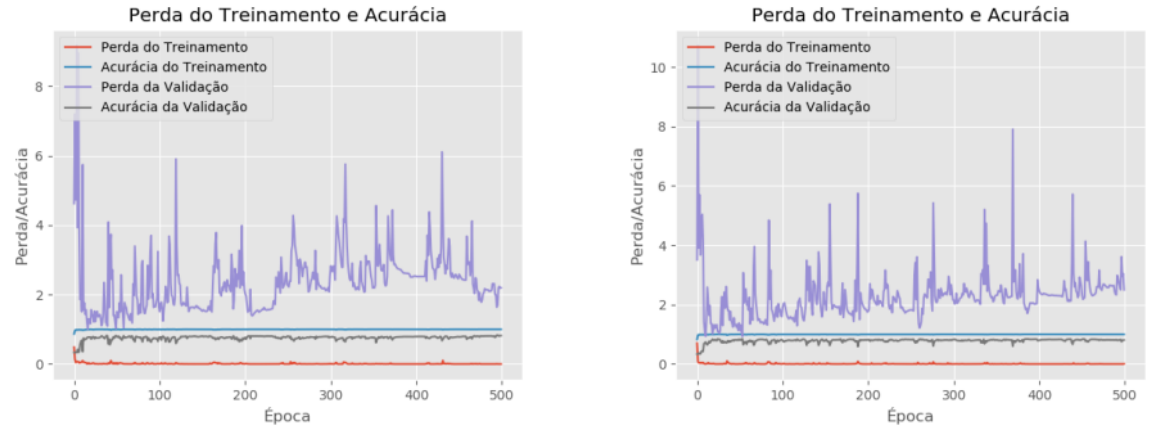
Figura 15 – Perdas e acurácias da *SmallerVGGNet* (Casos 1 e 2)



(a) Caso 1 (b) Caso 2

Fonte: Elaborado pelo autor.

Figura 16 – Perdas e acurácias da *SmallerVGGNet* (Casos 3 a 4)



(a) Caso 3 (b) Caso 4

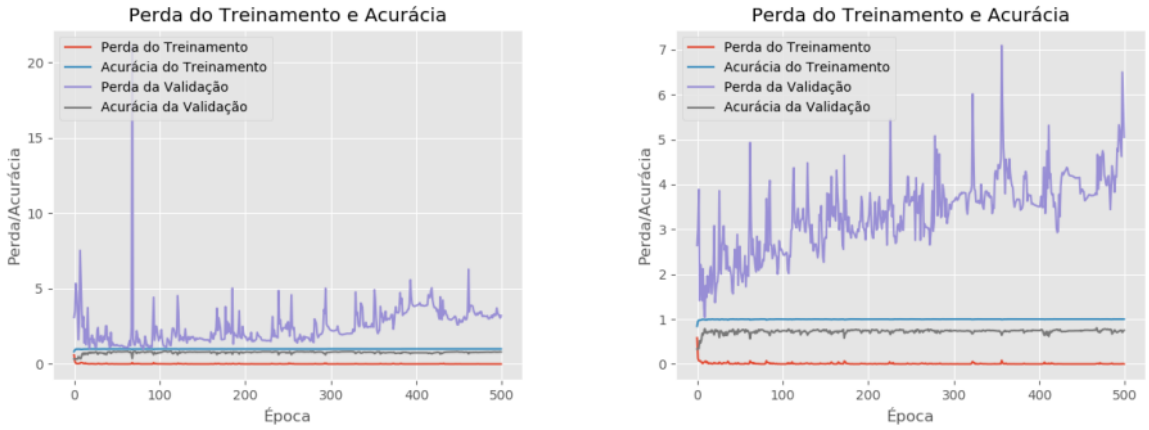
Fonte: Elaborado pelo autor.

6.6 Discussão dos resultados

Conforme os resultados obtidos, verificados na Tabela 4 e 5, a melhor combinação de métodos de DA foi o caso 8, com *flip*, *crop*, *rotation* e *scale*. A combinação obteve melhores resultados nas duas redes neurais utilizadas, obtendo uma melhoria da acurácia de até 22.09% na *AlexNet* e 12.57% na *SmallerVGGNet* em relação ao uso dos *datasets* sem o uso de métodos de DA.

Outros bons resultados, foram os casos de estudo 3, 7 e 9. A combinação de métodos de DA utilizada no caso de estudo 3 (*flip*, *crop* e *rotation*) obteve um bom resultado na *SmallerVGGNet*, enquanto o caso de estudo 9 (*flip*, *crop*, *rotation*, *gaussian filter*, *gaussian noise*, *scale* e *shear*) obteve melhor resultado na *AlexNet*. Já o caso de estudo 7 (*flip*, *crop*, *rotation*

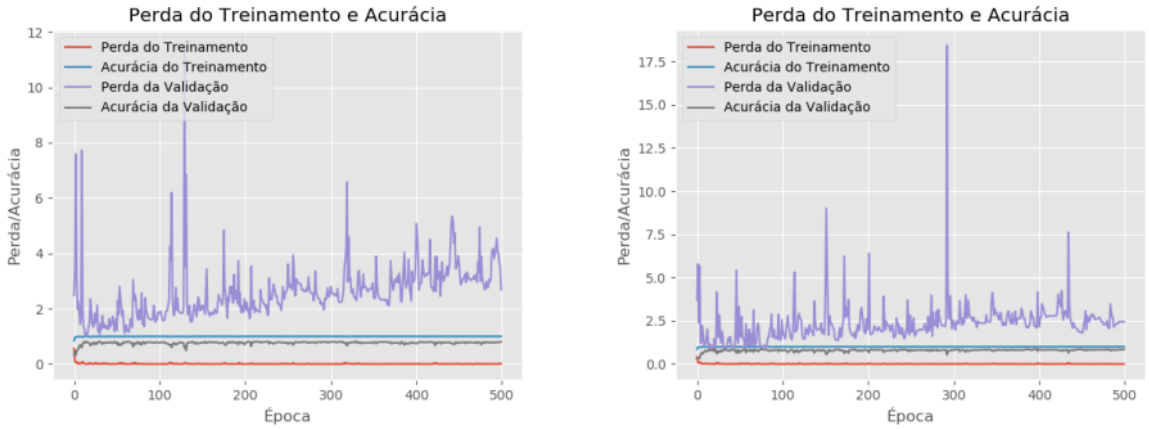
Figura 17 – Perdas e acurácias da *SmallerVGGNet* (Casos 5 e 6)



(a) Caso 5 (b) Caso 6

Fonte: Elaborado pelo autor.

Figura 18 – Perdas e acurácias da *SmallerVGGNet* (Casos 7 e 8)



(a) Caso 7 (b) Caso 8

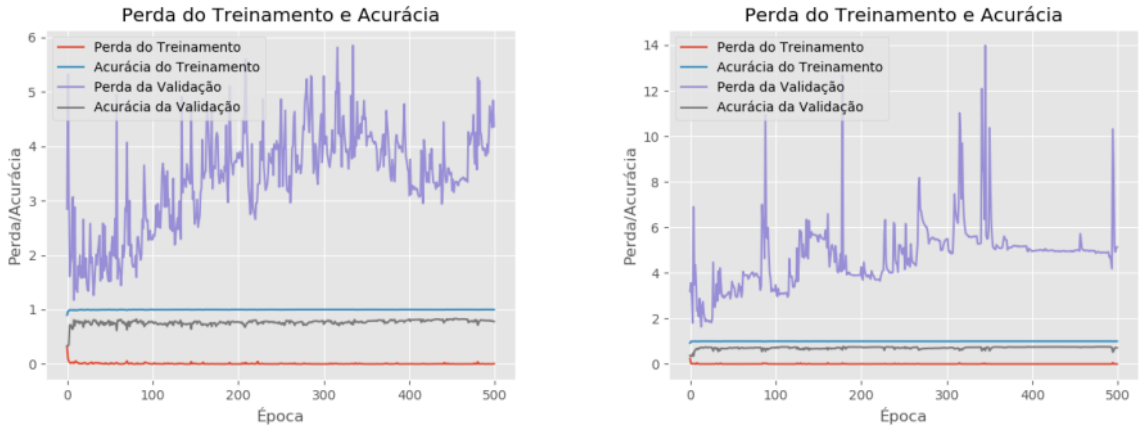
Fonte: Elaborado pelo autor.

e *gaussian noise*), apresentou melhores resultados em ambas redes neurais.

O uso de *gaussian noise* levou a piores resultados nas pesquisas de Shijie et al. (2017) e Hussain et al. (2017). Entretanto, seu uso nos casos de estudo 7 e 9 obteve bons resultados. Isso demonstra como os métodos e combinações adequadas variam de acordo com o contexto em que uma classificação está sendo realizada.

Além do aumento da acurácia, os métodos de DA de fato reduziram o *overfitting*, conforme visto nos resultados e nos gráficos (Figuras 15, 16, 20 e 26).

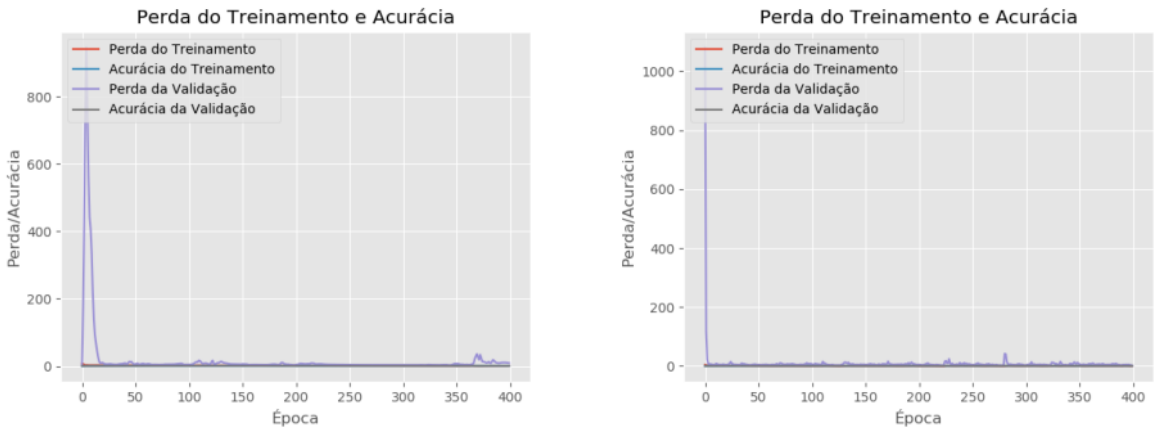
Figura 19 – Perdas e acurácias da *SmallerVGGNet* (Casos 9 e 10)



(a) Caso 9 (b) Caso 10

Fonte: Elaborado pelo autor.

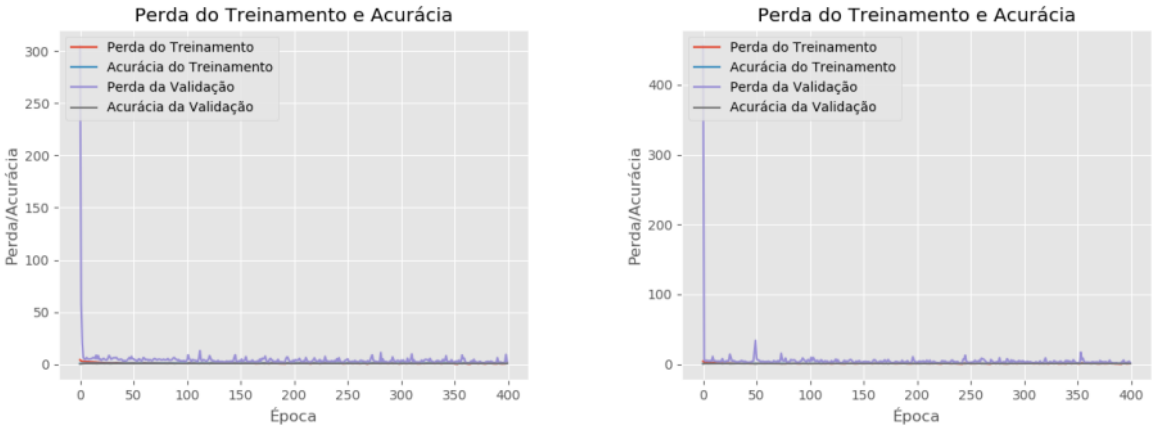
Figura 20 – Perdas e Acurácias da *AlexNet* (Casos 1 e 2)



(a) Caso 1 (b) Caso 2

Fonte: Elaborado pelo autor.

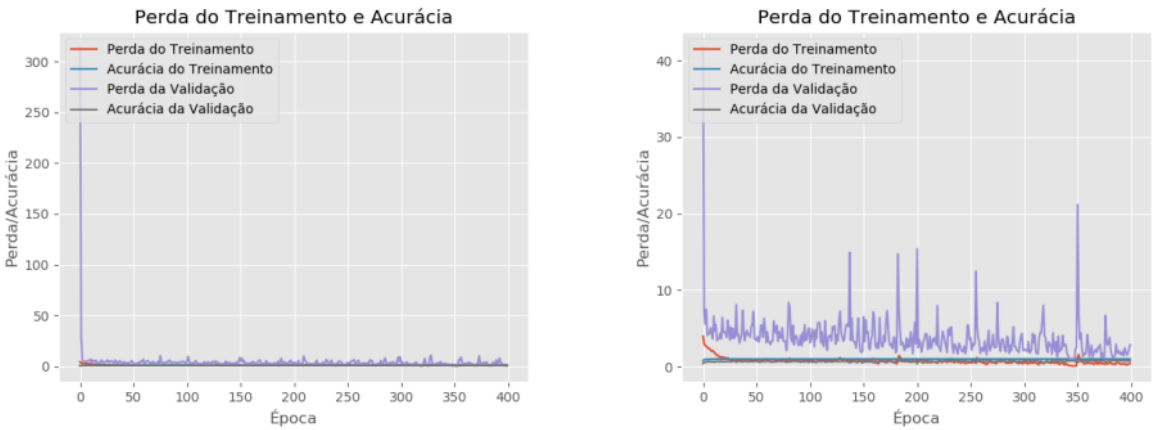
Figura 21 – Perdas e Acurácias da *AlexNet* (Casos 3 e 4)



(a) Caso 3 (b) Caso 4

Fonte: Elaborado pelo autor.

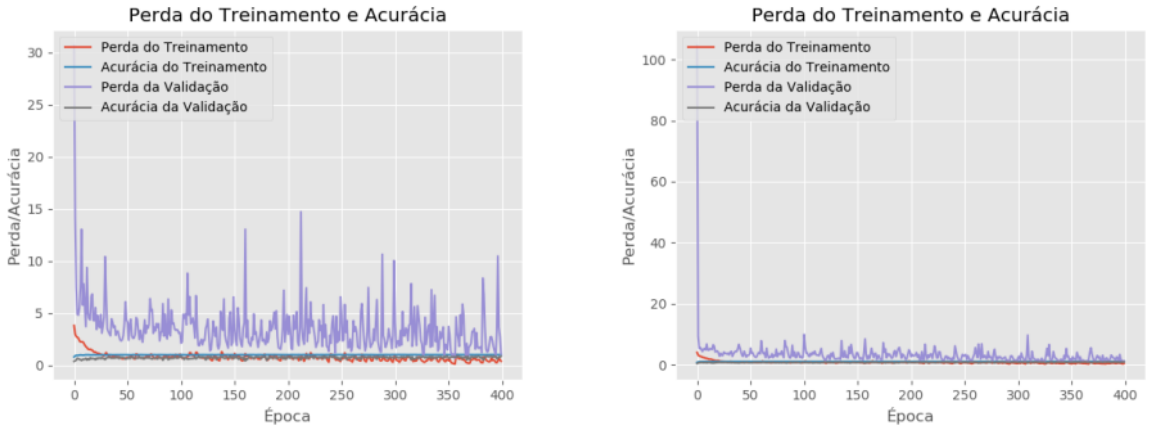
Figura 22 – Perdas e Acurácias da *AlexNet* (Casos 5 e 6)



(a) Caso 5 (b) Caso 6

Fonte: Elaborado pelo autor.

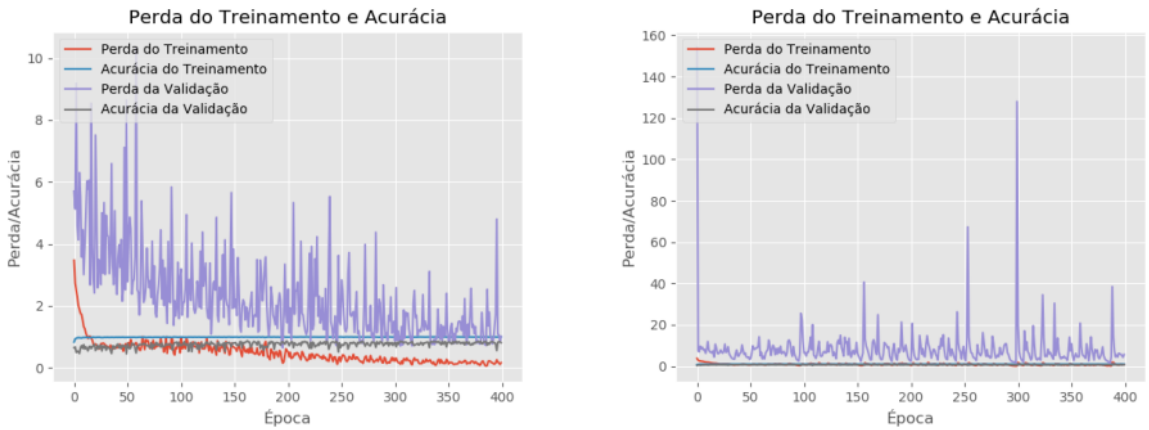
Figura 23 – Perdas e acurácias da *AlexNet* (Casos 7 e 8)



(a) Caso 7 (b) Caso 8

Fonte: Elaborado pelo autor.

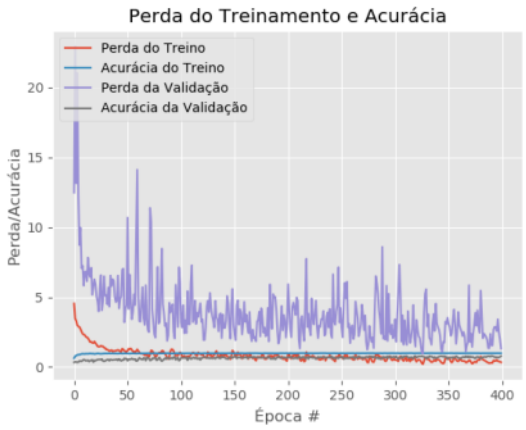
Figura 24 – Perdas e acurácias da *AlexNet* (Casos 9 e 10)



(a) Caso 9 (b) Caso 10

Fonte: Elaborado pelo autor.

Figura 25 – Perdas e Acurácias da *AlexNet* (Casos 8 e 10) para imagens em escala de cinza

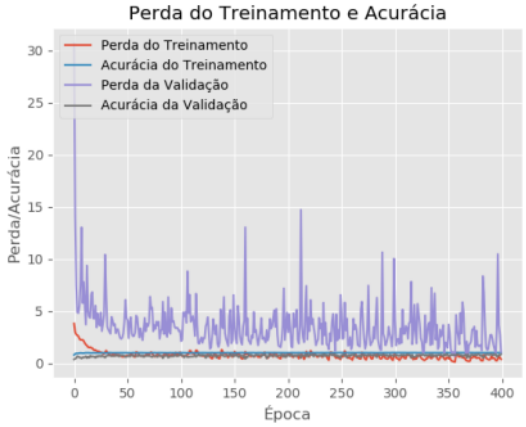


(a) Caso 8

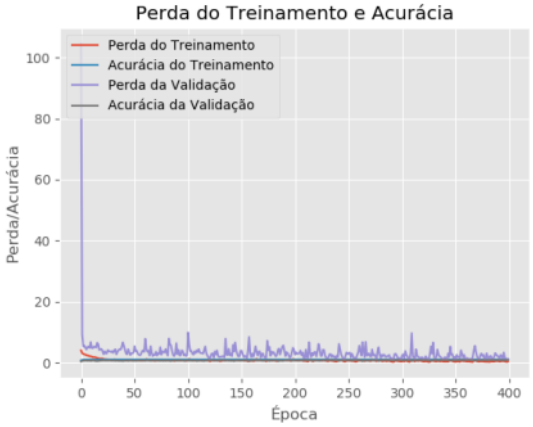
(b) Caso 10

Fonte: Elaborado pelo autor.

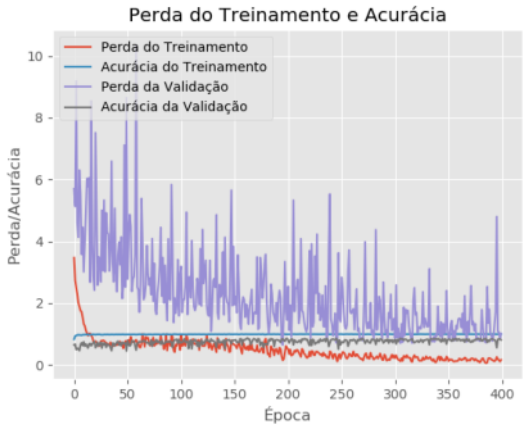
Figura 26 – Perdas e Acurácias da *AlexNet* (Casos 7 a 10)



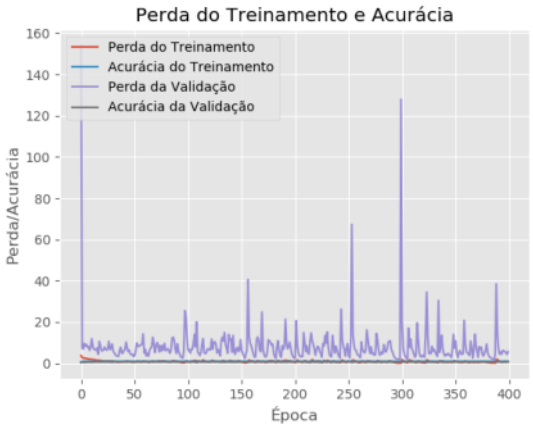
(a) Caso 7



(b) Caso 8



(c) Caso 9



(d) Caso 10

Fonte: Elaborado pelo autor.

7 Considerações Finais

Os resultados comprovaram a melhora que os métodos de *data augmentation* podem ocasionar no treinamento de redes neurais convolucionais para classificação de imagens e a importância de selecionar a combinação de métodos adequada, pois certas combinações podem reduzir a acurácia, em vez de aumentá-la.

Para o contexto de classificação de logomarcas em imagens, a combinação dos métodos *flip*, *crop*, *rotation* e *scale* obteve melhores resultados com os testes realizados em dois modelos de redes neurais convolucionais, *SmallerVGGNet* e *AlexNet*.

A melhor combinação se destacou em ambas as redes neurais utilizadas, obtendo uma melhoria na acurácia de até 22.09% na *AlexNet* e 12.57% na *SmallerVGGNet* em relação ao uso dos *datasets* sem o uso de métodos de *data augmentation*.

Outros bons resultados foram a combinação de *flip*, *crop* e *rotation*, que obteve um bom resultado na *SmallerVGGNet*, e *flip*, *crop*, *rotation*, *gaussian filter*, *gaussian noise*, *scale* e *shear* (a utilização de todos os métodos propostos), que obteve melhor resultado na *AlexNet*. *Flip*, *crop*, *rotation* e *gaussian noise* também apresentou bons resultados em ambos os modelos de redes neurais convolucionais.

7.1 Considerações e limitações

Embora o uso da técnica de *data augmentation* tenha ocasionado uma melhoria na acurácia da classificação, outras abordagens de *regularization* poderiam ser utilizadas em conjunto para aperfeiçoar os resultados, de forma que o classificador se generalize ainda mais.

Para selecionar a melhor combinação de métodos de *data augmentation* para um determinado contexto, também deve ser levado em consideração o modelo de rede neural utilizado, pois os resultados mostraram que determinadas combinações foram melhores para um dos modelos, enquanto para o outro não foram e vice-versa.

7.2 Trabalhos futuros

A seguir são apresentadas algumas propostas de trabalhos futuros. Estas sugestões visam resolver o problema proposto por esta pesquisa de outras maneiras. Portanto, recomenda-se:

- Utilizar *data augmentation* com outras classes de redes neurais. Neste estudo foram utilizadas redes neurais convolucionais, comprovando a melhora que DA ocasiona nelas, o que pode não ocorrer com outras arquiteturas.
- Abordar métodos de *transfer learning*, uma outra técnica de *regularization* também capaz de melhorar a acurácia e evitar o *overfitting* no caso em que grande quantidade de dados

para o *dataset* não estão disponíveis;

- Utilizar as melhores combinações dos métodos de *data augmentation* abordados nesta pesquisa juntamente com outras técnicas de *regularization* para aprimorar a acurácia da classificação;
- Abordar casos de imagens que não possuem rótulos (não possuem nenhuma logomarca) ou imagens que possuem uma ou mais logormarcas;
- Implementar uma ferramenta capaz de extrair imagens de redes sociais utilizando o melhor caso de modelo de rede neural treinado deste estudo.

Referências

AFFINE Transformation MATLAB Simulink. 2019. Disponível em: <https://www.mathworks.com/discovery/affine-transformation.html>. Citado nas páginas 9, 30 e 31.

BAIRD, C. H.; PARASNIS, G. From social media to social customer relationship management. *Strategy Leadership*, v. 39, p. 30–37, 09 2011. Citado na página 14.

BORUCKA, M. An excellent desk reference on international brand protection online. *Journal of Intellectual Property Law Practice*, v. 13, n. 1, p. 88–89, 10 2017. ISSN 1747-1532. Disponível em: <https://doi.org/10.1093/jiplp/jpx174>. Citado na página 14.

BRAND Logo Classifier. 2019. Disponível em: <https://github.com/matheus7mm/Final-Project---Computer-Vision>. Citado na página 36.

CLIFTON, R.; SIMMONS, J.; AHMAD, S. *Brands and Branding*. Wiley, 2004. (Economist (Hardcover)). ISBN 9781576601471. Disponível em: <https://books.google.com.br/books?id=ykUGuV3ncaEC>. Citado na página 22.

CREATIVE, T. L. *Apple Logo Evolution – It all Started With a Fruit*. 2018. Disponível em: <https://www.thelogocreative.co.uk/apple-logo-evolution-it-all-started-with-a-fruit/>. Citado nas páginas 14 e 15.

CUN, Y. L. Learning process in an asymmetric threshold network. In: ____. [S.l.: s.n.], 1986. p. 233–240. ISBN 978-3-642-82659-7. Citado na página 24.

Fawzi, A. et al. Adaptive data augmentation for image classification. In: *2016 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2016. p. 3688–3692. ISSN 2381-8549. Citado nas páginas 29 e 30.

GAO, Y. et al. Brand data gathering from live social media streams. In: *Proceedings of International Conference on Multimedia Retrieval*. New York, NY, USA: ACM, 2014. (ICMR '14), p. 169:169–169:176. ISBN 978-1-4503-2782-4. Disponível em: <http://doi.acm.org/10.1145/2578726.2578748>. Citado na página 13.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado nas páginas 15 e 29.

Guo, T. et al. Simple convolutional neural network on image classification. In: *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*. [S.l.: s.n.], 2017. p. 721–724. Citado na página 24.

HAN, D.; LIU, Q.; FAN, W. A new image classification method using cnn transfer learning and web data augmentation. *Expert Systems with Applications*, v. 95, p. 43 – 56, 2018. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417417307844>. Citado nas páginas 15, 16 e 30.

HOI, S. C. H. et al. Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. *CoRR*, abs/1511.02462, 2015. Disponível em: <http://arxiv.org/abs/1511.02462>. Citado na página 22.

HOSSAIN, M. S.; ALHAMID, M. F.; MUHAMMAD, G. Collaborative analysis model for trending images on social networks. *Future Generation Computer Systems*, v. 86, p. 855 – 862, 2018. ISSN 0167-739X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X17301383>. Citado na página 13.

Hu, W. et al. Image classification using multiscale information fusion based on saliency driven nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, v. 23, n. 4, p. 1513–1526, April 2014. ISSN 1057-7149. Citado na página 15.

HUSSAIN, Z. et al. Differential data augmentation techniques for medical imaging classification tasks. American Medical Informatics Association, 2017. ISSN 1559-4076. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977656/>. Citado nas páginas 9, 16, 34, 38 e 48.

Ismail Fawaz, H. et al. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv e-prints*, p. arXiv:1808.02455, Aug 2018. Citado nas páginas 15 e 35.

KERAS. 2019. Disponível em: <https://keras.io/>. Citado nas páginas 32 e 36.

KLETTE, R. *Concise Computer Vision: An Introduction into Theory and Algorithms*. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 1447163192, 9781447163190. Citado nas páginas 25, 26 e 28.

Kotkar, V. A.; Gharde, S. S. Image contrast enhancement by preserving brightness using global and local features. In: *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*. [S.l.: s.n.], 2013. p. 262–271. Citado na página 28.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>. Citado na página 37.

LIU, X.; ZHANG, B. Automatic collecting representative logo images from the internet. *Tsinghua Science and Technology*, v. 18, p. 606–617, 12 2013. Citado na página 14.

LOGOGRAB. 2019. Disponível em: <https://www.logograb.com/>. Citado na página 32.

Molina, J. F. et al. Color and size image dataset normalization protocol for natural image classification: A case study in tomato crop pathologies. In: *Symposium of Signals, Images and Artificial Vision - 2013: STSIVA - 2013*. [S.l.: s.n.], 2013. p. 1–5. ISSN 2329-6232. Citado na página 15.

NUMPY. 2019. Disponível em: <https://www.numpy.org/>. Citado na página 36.

OPENCV. 2019. Disponível em: <https://opencv.org/about/>. Citado na página 36.

PYTORCH. 2019. Disponível em: <https://pytorch.org/>. Citado na página 33.

RINGNÉR, M. What is principal component analysis? *Nature biotechnology*, Nature Publishing Group, v. 26, n. 3, p. 303, 2008. Citado nas páginas 34 e 35.

Rizvi, S. T. H. et al. Gabor filter based image representation for object classification. In: *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*. [S.l.: s.n.], 2016. p. 628–632. Citado na página 15.

ROSEBROCK, A. *Keras and Convolutional Neural Networks (CNNs)*. 2018. Disponível em: <https://www.pyimagesearch.com/2018/04/16/keras-and-convolutional-neural-networks-cnns/>. Citado na página 37.

ROSEBROCK, A. *Deep Learning for Computer Vision with Python*. 2.1.0. ed. [S.l.]: PyImageSearch.com, 2019. Citado nas páginas 16, 23, 24, 25, 27, 28, 29, 30 e 37.

SCIKIT-IMAGE. 2019. Disponível em: <https://scikit-image.org/>. Citado na página 36.

Shijie, J. et al. Research on data augmentation for image classification based on convolution neural networks. In: *2017 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2017. p. 4165–4170. Citado nas páginas 16, 34, 38 e 48.

SILVA, J. de Oliveira da. O estudo das identidades visuais flexíveis com aplicação na identidade visual da uneb. Universidade do Estado da Bahia - UNEB, v. 1, n. 1, p. 90, jul 2017. Citado na página 15.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2015. Citado na página 37.

SMITH, K. *123 Amazing Social Media Statistics and Facts*. 2019. Disponível em: <https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/>. Citado na página 14.

Sun, K. W.; Lee, C. H.; Wang, J. Multilabel classification via co-evolutionary multilabel hypernetwork. *IEEE Transactions on Knowledge and Data Engineering*, v. 28, n. 9, p. 2438–2451, Sep. 2016. ISSN 1041-4347. Citado na página 29.

SYSTEM, C. A. L. *CALS Con Screening: Spider-Man: Homecoming (2017, PG-13)*. 2017. Disponível em: <https://cals.org/event/spiderman-homecoming/>. Citado na página 26.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. [S.l.]: Springer, 2010. ISBN 1848829345, 9781848829343. Citado nas páginas 22 e 23.

TANAKA, M. *Classificação de imagens com deep learning e TensorFlow*. 2018. Disponível em: <https://imasters.com.br/back-end/classificacao-de-imagens-com-deep-learning-e-tensorflow>. Citado nas páginas 24 e 25.

TWITTER. *Postagem do Twitter*. 2016. Disponível em: <https://twitter.com/elonmusk/status/700398127044362240>. Citado na página 13.

Vieira, V. et al. *Tópicos em gerenciamento de dados e informações 2017*. [S.l.]: Sociedade Brasileira de Computaçãoo, 2017. Citado na página 23.

VISION AI. 2019. Disponível em: <https://cloud.google.com/vision/>. Citado na página 32.

WANG, F. et al. Logo information recognition in large-scale social media data. *Multimedia Systems*, v. 22, n. 1, p. 63–73, Feb 2016. ISSN 1432-1882. Disponível em: <https://doi.org/10.1007/s00530-014-0393-x>. Citado na página 13.

WATSON Visual Recognition. 2019. Disponível em: <https://www.ibm.com/watson/services/visual-recognition/>. Citado na página 32.

WAZLAWICK, R. *Metodologia de Pesquisa em Ciência da Computação*. [S.l.]: Elsevier, 2009. ISBN 8535235221, 9788535235227. Citado na página 19.

Wong, S. C. et al. Understanding data augmentation for classification: When to warp? In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. [S.l.: s.n.], 2016. p. 1–6. Citado na página 42.

YAN, Q.; WU, L.; ZHENG, L. Social network based microblog user behavior analysis. *Physica A: Statistical Mechanics and its Applications*, v. 392, n. 7, p. 1712 – 1723, 2013. ISSN 0378-4371. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0378437112010540>. Citado na página 13.

Yuille, A. L.; Liu, C. Deep Nets: What have they ever done for Vision? *arXiv e-prints*, p. arXiv:1805.04025, May 2018. Citado na página 15.