

Willian Dihanster Gomes de Oliveira

**Análise de Diferentes Algoritmos  
Semissupervisionados para *Data Augmentation***

São José dos Campos, Brasil

2019

Willian Dihanster Gomes de Oliveira

## **Análise de Diferentes Algoritmos Semissupervisionados para *Data Augmentation***

Relatório apresentando como comprovação das atividades de iniciação científica realizada durante o período de de agosto de 2018 a julho de 2019.

Projeto de Iniciação Científica 2018-2019 financiado pelo  
Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq

Relatório referente ao período de agosto de 2018 a julho de 2019.

---

**Lilian Berton**  
Orientadora

São José dos Campos, Brasil  
2019

# Resumo

A classificação de imagens é uma das grandes áreas de aplicação do Aprendizado de Máquina (AM). No entanto, para um bom desempenho dos algoritmos, faz-se necessária uma grande quantidade de dados, especialmente rotulados, além de um bom descritor de imagens. Em aplicações onde se tem dados rotulados e não rotulados, pode-se aplicar um algoritmo semissupervisionado baseado em grafos para propagar a informação de rótulo para exemplos não rotulados e consequentemente, conseguir mais dados para treinamento do classificador. Portanto, o objetivo deste trabalho é analisar diferentes algoritmos semissupervisionados para *Data Augmentation* (DA) e o uso de redes neurais convolucionais (CNNs) como extratores de características de imagens.

**Palavras-chave:** *data augmentation*. aprendizado de máquina. aprendizado semissupervisionado. classificação de imagens.

# Sumário

|            |                                                                   |           |
|------------|-------------------------------------------------------------------|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                                                 | <b>4</b>  |
| <b>1.1</b> | <b>Contextualização e Motivação</b>                               | <b>4</b>  |
| <b>1.2</b> | <b>Objetivos</b>                                                  | <b>5</b>  |
| 1.2.1      | Objetivos Gerais                                                  | 5         |
| 1.2.2      | Objetivos Específicos                                             | 5         |
| <b>2</b>   | <b>FUNDAMENTAÇÃO TEÓRICA</b>                                      | <b>6</b>  |
| <b>2.1</b> | <b>Aprendizado de Máquina</b>                                     | <b>6</b>  |
| 2.1.1      | Conceitos de Aprendizado de Máquina                               | 6         |
| 2.1.2      | Algoritmos Preditivos                                             | 7         |
| 2.1.2.1    | Classificação de Imagens                                          | 8         |
| 2.1.2.1.1  | Extração de Características                                       | 8         |
| 2.1.2.2    | <i>Support Vector Machine (SVM)</i>                               | 9         |
| 2.1.2.3    | Redes Neurais Artificiais (RNAs)                                  | 10        |
| 2.1.2.4    | <i>Convolutional Neural Networks (CNNs)</i>                       | 12        |
| 2.1.2.5    | Modelos de <i>Convolutional Neural Networks</i>                   | 13        |
| <b>2.2</b> | <b><i>Data Augmentation</i></b>                                   | <b>13</b> |
| 2.2.1      | Transformação de Imagens                                          | 14        |
| 2.2.2      | Algoritmos Semissupervisionados                                   | 15        |
| <b>3</b>   | <b>REVISÃO BIBLIOGRÁFICA</b>                                      | <b>16</b> |
| <b>4</b>   | <b>METODOLOGIA</b>                                                | <b>17</b> |
| <b>4.1</b> | <b>Bases de Dados</b>                                             | <b>17</b> |
| <b>4.2</b> | <b>Softwares e Bibliotecas</b>                                    | <b>18</b> |
| <b>4.3</b> | <b>Experimentação</b>                                             | <b>18</b> |
| <b>5</b>   | <b>RESULTADOS</b>                                                 | <b>19</b> |
| <b>5.1</b> | <b>Experimentos com Transformação de Imagens (TIs)</b>            | <b>19</b> |
| <b>5.2</b> | <b>Experimentos com os Algoritmos Semissupervisionados (GSSL)</b> | <b>19</b> |
| <b>5.3</b> | <b>Comparação</b>                                                 | <b>21</b> |
| <b>5.4</b> | <b>Discussão</b>                                                  | <b>21</b> |
| <b>6</b>   | <b>CONCLUSÃO</b>                                                  | <b>22</b> |
|            | <b>REFERÊNCIAS</b>                                                | <b>23</b> |

# 1 Introdução

## 1.1 Contextualização e Motivação

O Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial, que se baseia em construir e modelar sistemas capazes de adquirir conhecimento e que melhorem automaticamente com a experiência (MITCHELL et al., 1990). No AM o aprendizado indutivo é o mais comum, ou seja, são utilizadas técnicas de indução para obter conclusões genéricas sobre um conjunto particular de exemplos (MONARD; BARANAUSKAS, 2003). Este tipo de aprendizado possui uma hierarquia, podendo ser dividido entre aprendizado supervisionado, não supervisionado e semissupervisionado.

No aprendizado supervisionado, a classificação é uma das tarefas mais conhecidas e com diversas aplicações. Mas, necessita-se uma grande quantidade de dados rotulados para gerar classificadores com boa capacidade de predição. Nem sempre é possível ter uma grande quantidade de dados rotulados, e a obtenção dos mesmos não é uma tarefa fácil e geralmente é custosa. Por exemplo, dados médicos precisam que exames sejam realizados, o que toma tempo, recursos monetários e dependendo da condição médica, pode ser um caso raro. Além disso, nesse caso, é necessário a anotação de um especialista para rotular o dado.

Dessa forma, diversas técnicas para obtenção de mais dados artificiais foram propostas, surgindo o conceito de *data augmentation* (DA). Ou seja, em DA a ideia é utilizar técnicas computacionais com o objetivo de aumentar a quantidade de exemplos rotulados em um conjunto de dados. Em imagens, uma abordagem simples e comum é expandir o conjunto de dados por meio de transformações de imagens. Assim, gera-se novas imagens a partir de rotações, cortes, efeitos *blur* e *etc.* Uma outra abordagem comum, em aprendizado semissupervisionado é tentar rotular essas imagens não rotuladas, a partir da informação de rótulo das imagens rotuladas.

Além disso, em algumas aplicações de classificação de imagens é necessário representar as imagens por um vetor de suas características e portanto utilizar um bom descritor de imagens. Atualmente, as *Convolutional Neural Networks* (CNNs) estão sendo muito utilizadas para essa tarefa. As CNNs são redes neurais baseadas no processo biológico do processamento de dados visuais e possuem uma fase de extração de características e uma de classificação.

Sendo assim, a geração de dados artificiais através de técnicas de *data augmentation* pode ser uma alternativa mais viável. Diversos trabalhos já aplicaram DA e obtiveram melhora em seus resultados. (HUSSAIN et al., 2017) aplicaram DA em imagens de

mamografia e conseguiram aumentar acurácia de validação de 84% para 88%. (WANG et al., 2018) também utilizaram DA em dados médicos, mas dessa vez com imagens do cérebro para detecção de alcoolismo, os autores conseguiram uma acurácia de 97.04% e resultados superior a três outras abordagens do estado da arte.

Portanto, neste trabalho, iremos utilizar e comparar duas abordagens diferentes de obtenção de mais dados rotulados: Transformação de Imagens (TIs) e Algoritmos Semissupervisionado Baseado em Grafos (GSSL) em três base de dados de imagens de aplicações diferentes. Para trabalhar com as imagens, serão utilizadas 4 diferentes Redes Neurais Convolucionais (CNNs) para descrever as imagens em suas características. Por fim, as bases geradas serão validadas por experimentos de classificação com o algoritmo *Support Vector Machine* (SVM). Nos experimentos foram tomadas 5 conjuntos pequenos a partir do conjunto maior, e em seguida, aumentados esses conjuntos pequenos pelas técnicas de DA, até atingir o tamanho do conjunto maior. Com isso, verifica-se se os resultados de classificação utilizando os datasets aumentados por DA estão próximos da base original.

## 1.2 Objetivos

### 1.2.1 Objetivos Gerais

O principal objetivo deste trabalho é estudar e analisar duas diferentes abordagens de DA aplicadas à classificação de imagens.

### 1.2.2 Objetivos Específicos

- Estudar os principais conceitos de AM e os diferentes tipos de redes neurais, como a padrão e as convolucionais.
- Estudar e selecionar técnicas de DA, como a Transformação de Imagens e Algoritmos Semissupervisionados.
- Estudar sobre extração de características de imagens por meio de CNNs.
- Aplicar e avaliar as técnica de Transformação de Imagens nos conjuntos de dados.
- Aplicar e avaliar quatro Algoritmos Semissupervisionados nos conjuntos de dados.
- Analisar e comparar os resultados obtidos pelas duas abordagens.

## 2 Fundamentação Teórica

### 2.1 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial, que se baseia em construir e modelar sistemas capazes de adquirir conhecimento de forma automática e que melhorem automaticamente com a experiência (MICHALSKI; TECUCI, 1994). No AM a forma de adquirir conhecimento e prever eventos futuros é através do aprendizado indutivo. Esse aprendizado indutivo possui uma hierarquia, podendo ser dividido em supervisionado, não supervisionado e semissupervisionado (MONARD; BARANAUSKAS, 2003).

No aprendizado supervisionado, o algoritmo de aprendizado trabalha com um conjunto de treinamento onde o rótulo ou classe de cada exemplo é conhecido. Assim, o objetivo deste algoritmo é construir um estimador que possa determinar corretamente o rótulo de novos exemplos, que não faziam parte do conjunto de treino. Em problemas com rótulos discretos, o problema é conhecido como classificação, para rótulos contínuos, regressão (MONARD; BARANAUSKAS, 2003).

Já no aprendizado não-supervisionado, o algoritmo de aprendizado trabalha com um conjunto de treinamento onde o rótulo ou classe de cada exemplo não é conhecido. E então é comum a tarefa de agrupamento, onde o objetivo é analisar os exemplos fornecidos e determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou *clusters* (MONARD; BARANAUSKAS, 2003).

Ainda, há o aprendizado semissupervisionado definido como um intermédio entre o aprendizado supervisionado e não-supervisionado (CHAPELLE; SCHOLKOPF; ZIEN, 2009). Aqui o algoritmo de aprendizado trabalha com um conjunto de treinamento formado por exemplos rotulados e um conjunto (geralmente grande) de dados não rotulados.

#### 2.1.1 Conceitos de Aprendizado de Máquina

A seguir são apresentados os principais conceitos relacionados com AM, focado em Aprendizado Supervisionado. (MONARD; BARANAUSKAS, 2003)

- Indutor: é o algoritmo ou programa de aprendizado que usa indução a fim de extrair e definir um classificador a partir de um conjunto de exemplos. Esse indutor pode então ser utilizado para classificar novos exemplos.

- Exemplo: é um dado, registro ou modelo, comumente representado por uma tupla de valores, que vão ser usadas e/ou observadas pelos algoritmos.
- Atributo: é descrito como uma característica ou aspecto do exemplo, como cor, tamanho, *etc.* Esses atributos podem ser nominais ou contínuos.
- Classe: é uma espécie de atributo especial, um rótulo do exemplo. Geralmente descreve o fenômeno de interesse, que é a meta que se deseja aprender e fazer previsões a respeito.
- Conjunto de Exemplos: representa um conjunto de dados com seus atributos e respectivos valores. Exemplificado na Tabela 1.

Tabela 1 – Exemplo de um Conjunto de Dados.

|                | $\mathbf{A}_1$ | $\mathbf{A}_2$ | $\cdots$ | $\mathbf{A}_m$ | $\mathbf{Y}$ |
|----------------|----------------|----------------|----------|----------------|--------------|
| $\mathbf{X}_1$ | $x_{11}$       | $x_{12}$       | $\cdots$ | $x_{1m}$       | $y_1$        |
| $\mathbf{X}_2$ | $x_{21}$       | $x_{22}$       | $\cdots$ | $x_{2m}$       | $y_2$        |
| $\vdots$       | $\vdots$       | $\vdots$       | $\ddots$ | $\vdots$       | $\vdots$     |
| $\mathbf{X}_n$ | $x_{n1}$       | $x_{n2}$       | $\cdots$ | $x_{nm}$       | $y_n$        |

Fonte: O autor. Adaptado de (MONARD; BARANAUSKAS, 2003).

- *Bias*: é definido como qualquer preferência de um hipótese sobre outra. Como existe sempre um número grande de hipóteses consistentes, todos os indutores possuem alguma forma de *bias*.
- *Overfitting*: acontece quando um algoritmo não é capaz de generalizar para amostras diferentes, sendo muito específico/ajustado para o conjunto de exemplos apresentado.
- *Underfitting*: quando após a geração do modelo, a taxa de erro ainda é alta. Isto é, o algoritmo não conseguiu aprender.
- Ruído: são inconsistências de dados que podem fazer parte do conjunto de treinamento.
- *Outlier*: é quando um dado possui valores/descrição muito discrepante dos outros dados do conjunto de exemplos.

### 2.1.2 Algoritmos Preditivos

Uma das maiores aplicações do AM, é a predição. Um algoritmo de AM preditivo é uma função que, dados um conjunto de exemplos rotulados, constrói um estimador. Assim, dizemos que temos uma tarefa de classificação quando os rótulos são nominais e então o



estimador gerado é um classificador. Um classificador por sua vez é definido como uma função que, dado um exemplo ainda não rotulado, atribui esse exemplo a uma das possíveis classes. No caso de rótulos contínuos o estimador gerado é um regressor (FACELI et al., 2011).

Com isso, a classificação e a regressão possuem diversas aplicações na vida real. A classificação pode ser utilizada em problemas que podem ser separados entre categorias, por exemplo: classificar e-mails que são *spam* ou não são *spam*. A regressão pode ser aplicada em problemas em que deve-se estimar um valor, como aproximar funções. Neste trabalho será estudado e utilizado um algoritmo para classificação, o *Support Vector Machine*.

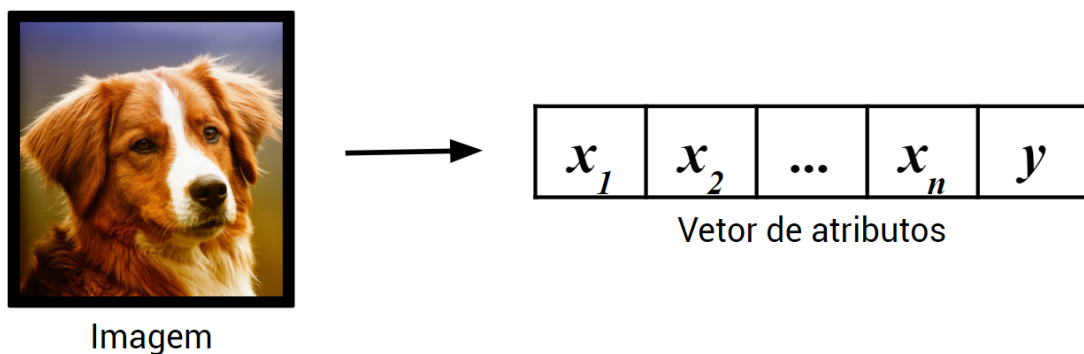
### 2.1.2.1 Classificação de Imagens

A classificação de imagens pode ser vista como uma caso especial da classificação. Neste tipo de classificação, o objetivo é dado uma imagem, predizer corretamente a qual classe a imagem pertence. Assim, classificar imagens pode ser uma tarefa muito útil em algumas aplicações, como por exemplo, detecção de doenças (WANG et al., 2018), reconhecimento facial (VARGAS; PAES; VASCONCELOS, 2016), análise de dados de satélite (PENATTI K. NOGUEIRA, 2015), dentre outras aplicações.

#### 2.1.2.1.1 Extração de Características

Na classificação, os conjuntos de dados são compostos pelas características dos exemplos. No entanto, com imagens a tarefa não é direta. Geralmente, para se trabalhar com imagens é necessário utilizar descritor de imagens para extrair características. Assim, as imagens poderão ser representadas como exemplos munidos de  $n$  atributos e podem formar um conjunto de dados. Esse processo é exemplificado na Figura 1.

Figura 1 – Exemplo prático da extração de características.



Fonte: O autor.

Atualmente, com o uso das *Convolutional Neural Networks*, a extração de características está acoplada e alcançando ótimos resultados (Lecun et al., 1998) devido ao fato da própria rede ter uma fase de extração de características e usar diversos filtros detectores de padrões, seguido pela fase de classificação (opcional).

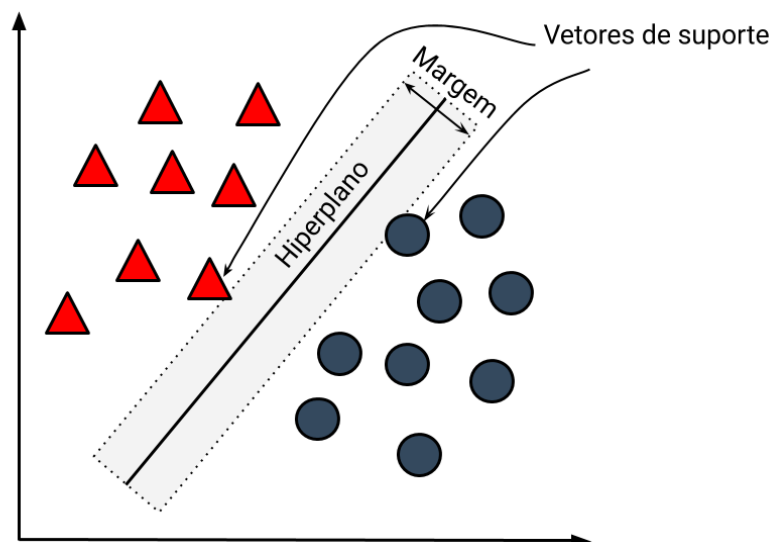
### 2.1.2.2 Support Vector Machine (SVM)

A *Support Vector Machine* é uma técnica proposta por (CORTES; VAPNIK, 1995) para classificação binária. (CORTES; VAPNIK, 1995) sumariza a ideia do método como: vetores de entrada não-lineares são mapeados para o espaço de atributos de alta dimensão, onde é construída uma superfície de decisão com uma alta capacidade de generalização.

Em outras palavras, deve-se encontrar um hiperplano ótimo, onde teremos uma fronteira de decisão que possua menor erro, mas maior margem de separação entre os exemplos das diferentes classes. Para isso, são usados uma pequena quantidade de dados de treinamento chamados *Support Vectors* (SVs), que ajudarão na determinação desta margem (LORENA; CARVALHO, 2007).

Na Figura 2 há um exemplo onde temos um hiperplano separando objetos diferentes, com uma determinada margem. Também é possível visualizar os SVs (Vetores de suporte, em português), que são os exemplos mais próximos da fronteira de decisão.

Figura 2 – Exemplo do cenário de uma SVM.



Fonte: O autor.

Entre as vantagens das SVMs estão sua boa capacidade de generalização, robustez para dados de grandes dimensões, conexidade na função objetivo (na SVM há a otimização de uma função quadrática e portanto, possui apenas um mínimo global) e o tempo de treinamento é geralmente menor que outros algoritmos de AM. Como desvantagem está

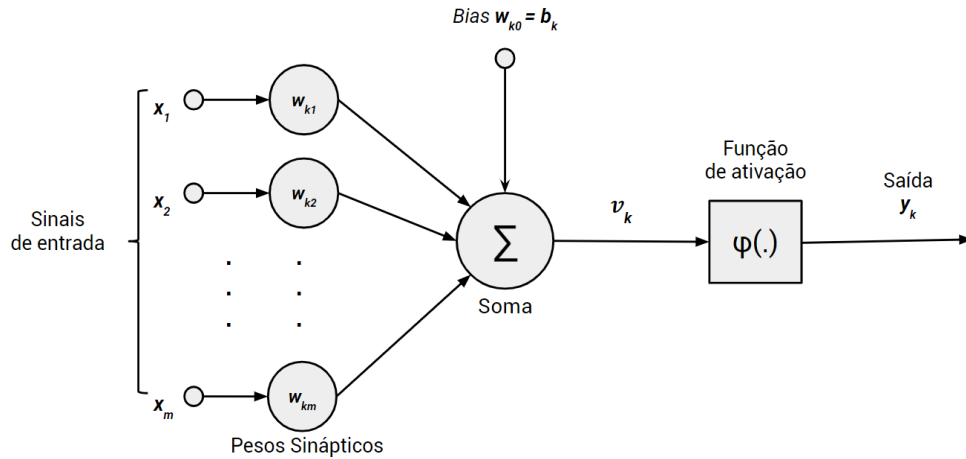
sua complexidade computacional (ordem  $n^2$  a  $n^3$ ), a velocidade de classificação pode ser um pouco demorada e não há como interpretar o modelo gerado (LORENA; CARVALHO et al., 2003).

### 2.1.2.3 Redes Neurais Artificiais (RNAs)

Segundo (HAYKIN, 2011) uma rede neural artificial pode ser vista como um processador constituído de processamento simples que têm a capacidade de armazenar conhecimento adquirido pela experiência e após isso, tornar esse conhecimento disponível para que possa ser usado. Sua semelhança ao cérebro está atrelado ao fato de que o conhecimento da rede é obtido por um processo de aprendizagem e o conhecimento adquirido é armazenado pela conexão dos neurônios ou pesos sinápticos.

A Figura 3 ilustra o modelo de um neurônio de McCulloch–Pitts (MCCULLOCH; PITTS, 1943). Esse neurônio é formado por três elementos: um conjunto de sinapses, um somador e uma função de ativação. Cada elemento do conjunto de sinapses representa um peso. O somador que soma os sinais de entrada ponderado pelas respectivas sinapses do neurônio. A função de ativação para restringir a amplitude da saída de um neurônio. Além disso, esse modelo inclui um *bias*, para aumentar ou diminuir a entrada líquida da função de ativação (HAYKIN, 2011).

Figura 3 – Modelo de um neurônio.



Fonte: O autor.

Assim, a soma ou a entrada do limitador abrupto ou o campo local induzido do neurônio é calculada por:

$$v_k = \sum_{i=0}^m w_{ki} x_i \quad (2.1)$$

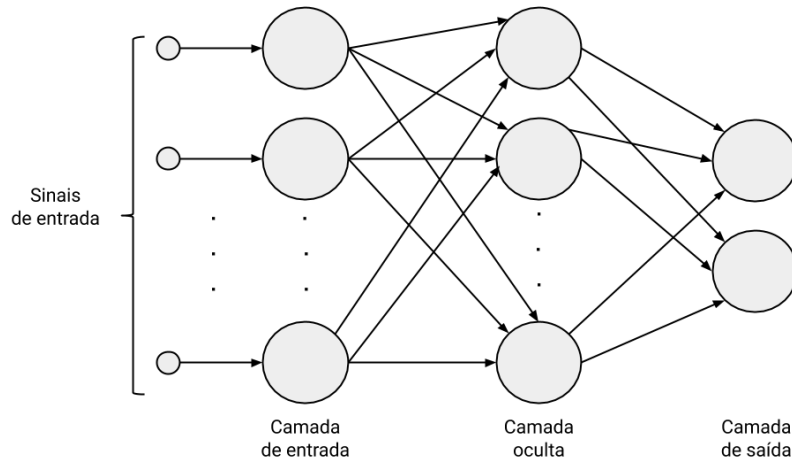
Dado  $v_k$ , um exemplo de função de ativação é a função limiar, dada por:

$$\varphi(v_k) = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (2.2)$$

No Perceptron ([ROSENBLATT, 1958](#)) o objetivo é classificar corretamente o conjunto de entradas em uma das classes  $l_1$  e  $l_2$ . A regra de decisão para a classificação é atribuir o ponto  $x_i$  à classe  $l_1$  se a saída do perceptron  $y_i$  for +1 e à classe  $l_2$  se for -1. Mas isso é idealmente para dados linearmente separáveis. Assim, os pesos sinápticos são adaptados a cada iteração com um algoritmo de correção de erro ([HAYKIN, 2011](#)).

Um exemplo de RNA é o *Multilayer Perceptron* (MLP). Essa rede dispõe de uma camada de entrada e de saída e uma ou mais camadas ocultas. Geralmente, possui alto nível de conectividade com as camadas totalmente conectadas, isto é, cada neurônio de uma camada está conectado a todos os neurônios da camada anterior. Na Figura 4 temos a representação da arquitetura de uma rede MLP ([HAYKIN, 2011](#)).

Figura 4 – Modelo de uma rede MLP.



Fonte: O autor.

Comumente, o treinamento é baseado no algoritmo de retropropagação do erro ([LINNAINMAA, 1970](#)), que é dividido em fase (*forward*) e fase (*backward*). Onde os erros são somados e no final, volta-se propagando e ajustando os pesos sinápticos.

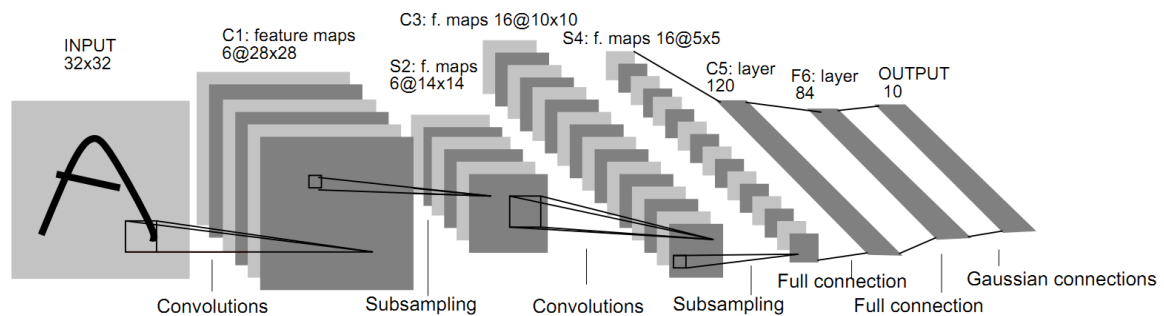
Dentre as vantagens das RNAs estão sua alta capacidade de generalização, auto-aprendizado e sua capacidade de adaptação e em geral. Como desvantagem, tem-se a dificuldade de interpretação do modelo gerado, a escolha dos parâmetros e o treinamento, muitas vezes, pode ser demorado em relação a outros algoritmos ([HAYKIN, 2011](#)).

### 2.1.2.4 Convolutional Neural Networks (CNNs)

As CNNs foram introduzidas por (Lecun et al., 1998) e diferente das redes neurais comuns, se inspiram no processo biológico do processamento de dados visuais. Essas redes aplicam filtros em dados visuais, mas mantêm a relação de vizinhança entre os *pixels* da imagem ao longo do processamento da rede (VARGAS; PAES; VASCONCELOS, 2016).

Além disso, essas redes possuem uma fase de extração de características das imagens e uma de classificação. Em geral, a fase de extração de características é composta por camadas de convolução e de *pooling*, e na de classificação, uma rede MLP (VARGAS; PAES; VASCONCELOS, 2016). Na Figura 5, temos um exemplo de arquitetura da CNN LeNet-5, que possui 3 camadas de convolução, 2 de *pooling* (ou *subsampling*), 1 camada totalmente conectada, seguida por uma camada de saída.

Figura 5 – Exemplo de arquitetura da CNN LeNet-5.



Fonte: (Lecun et al., 1998)

- **Convolução:** nessa camada, são gerados os mapas de *features* da imagem, pelo processo de convolução discreta, que preserva a noção de ordem, é esparsa (algumas unidades de entradas contribuem para uma unidade de saída) e reusa parâmetros (DUMOULIN; VISIN, 2016).
- **Pooling:** o *pooling* é responsável por reduzir o tamanho do mapa de *features*, usando alguma medida para sumarizar regiões das imagens. Algumas medidas comuns são o *AVG Pooling* e o *Max Pooling* (DUMOULIN; VISIN, 2016).
  - *AVG Pooling:* nesse tipo de *pooling*, o valor de saída é a média dos valores de uma vizinhança definido pela grade de *pooling* (DUMOULIN; VISIN, 2016).
  - *Max Pooling:* já neste tipo de *pooling*, o valor de saída é simplesmente o valor máximo dos valores presentes em uma dada vizinhança na grade de *pooling* (DUMOULIN; VISIN, 2016).
- **Camada Totalmente Conectada:** essa camada se encontra ao final do processamento das convoluções e *pooling* em uma CNN e serve para classificação. Essa camada é,

geralmente, formada por uma rede MLP e usa como entrada as *features* de saída da CNN. E por fim, a MLP fará o processo de classificação da imagem passado.

Muitas vezes o treinamento de uma rede CNN pode ser demorado aliado ao fato de demandar uma grande quantidade de dados. Para facilitar o processo de utilizar uma CNN, pode-se carregar os pesos (e outros parâmetros) de uma rede já treinada com uma grande quantidade de dados em uma rede. Esse processo é denominado transferência de aprendizado (TORREY; SHAVLIK, 2010), sendo possível utilizar diretamente a rede carregada com esses parâmetros ou ainda, utilizá-la e ao final, aplicar o *finetuning*. O processo de *finetuning* é onde a CNN realiza um pequeno treinamento para adaptar a rede com o novo conjunto de dados.

#### 2.1.2.5 Modelos de *Convolutional Neural Networks*

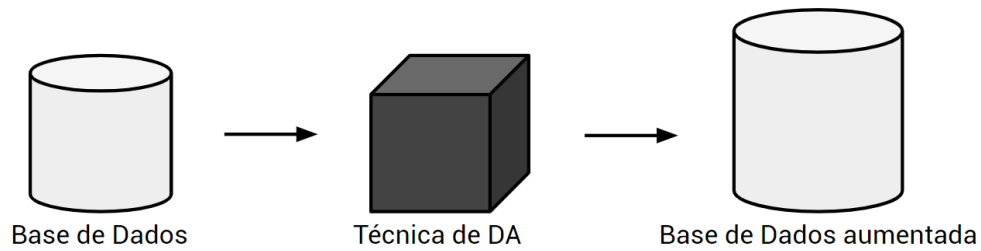
- VGG16 e VGG19: são CNNs com 16 e 19 *weight layers*, respectivamente. São 13 camadas de convolução para a VGG16 e 16 para VGG19, com 3 camadas totalmente conectadas e utilizam *pooling* do tipo *max pooling*. O vetor de características gerado pela rede, dado uma imagem é  $1 \times 4096$ .
- ResNet50: é uma CNN com 50 *weight layers*, sendo 49 de convolução e 1 camada totalmente conectada e o *pooling* é do tipo *global average pooling*. O vetor de características gerado pela rede, dado uma imagem é  $1 \times 2048$ .
- Xception: é uma CNN com 36 de camada convolução, estruturadas em 14 módulos e o *pooling* é do tipo *max pooling* e *global average pooling*. O vetor de características gerado pela rede, dado uma imagem é  $1 \times 4096$ .

## 2.2 *Data Augmentation*

Em *Data Augmentation* (DA), o objetivo é utilizar técnicas computacionais para aumentar o conjunto de treinamento supervisionado afim de se obter algoritmos com maiores capacidade de predição, e consequentemente, melhores resultados.

Dessa forma, existem diversas abordagens para se obter mais dados ao conjunto de treino. Em imagens, algumas técnicas de DA aplicam técnicas de Transformação de Imagens no conjunto de dados para gerar novos exemplos. Em casos onde se tem um conjunto de dados rotulados e não rotulados, pode-se utilizar heurísticas para rotular esses dados não rotulados, gerando um conjunto maior de treinamento de exemplos rotulados (MRKŠIĆ, 2013). Na Figura 6, temos um esquema prático do funcionamento do conceito de DA.

Figura 6 – Exemplo de funcionamento de DA.

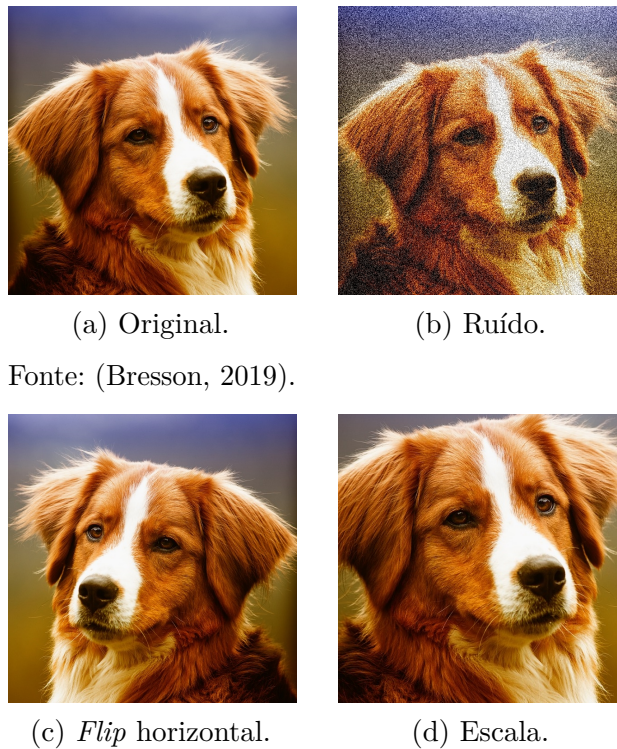


Fonte: O autor.

### 2.2.1 Transformação de Imagens

O método de DA de Transformação de Imagens é uma ideia simples e muito utilizada quando o objetivo é obter mais dados de imagens. Assim, dado um conjunto de imagens, aplica-se efeitos de transformação de imagens nos exemplos e ao final, o conjunto aumentado será composto pelas imagens originais e as imagens com aplicação dos efeitos. Na Figura 7 é apresentado um exemplo prático da aplicação de efeitos em imagens.

Figura 7 – Exemplo de transformações de imagens.



Fonte: (Bresson, 2019).

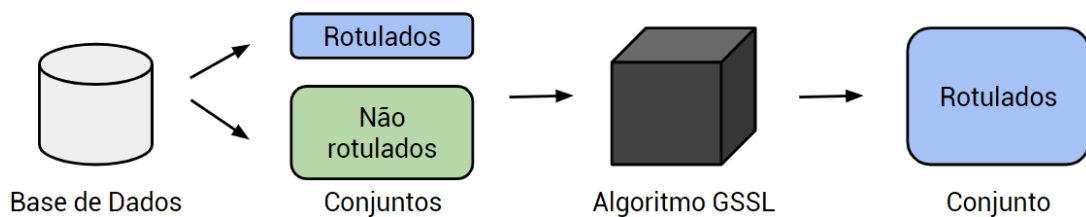
Fonte: O autor.



## 2.2.2 Algoritmos Semissupervisionados

Os algoritmos semissupervisionados se baseiam na ideia de: seja um conjunto de dados com exemplos rotulados e não rotulados, propaga-se a informação de rótulo desses exemplos rotulados para os exemplos não rotulados. Assim, gera-se um conjunto maior de exemplos rotulados para treinamento de um classificador, exemplificado pela Figura 8. Usualmente, utiliza-se um grafo para propagação dos rótulos, onde os vértices são os exemplos, e as arestas podem representar a semelhança entre os exemplos (como os grafos construído neste trabalho).

Figura 8 – Exemplo de funcionamento dos algoritmos semissupervisionados.



Fonte: O autor.

Alguns algoritmos são detalhados a seguir:

- *Label Propagation* (LP): nesse algoritmo, os rótulos são propagados para os vizinhos de acordo com a proximidade e inclusive os exemplos já rotulados podem ser alterados (ZHU; GHAMRANI, 2002).
- *Local and Global Consistency* (LGC): se baseia na ideia de Consistência Global e Local, isto é, pontos de mesma estrutura são prováveis de terem mesmo rótulo (global) e pontos próximos são prováveis de possuírem o mesmo rótulo (local) (ZHOU et al., 2004).
- *Gaussian Fields and Harmonic Functions* (GFHF): já nesse algoritmo, a ideia é utilizar funções harmônicas para rotular os dados e assim um rótulo atribuído a um nó será dado pela média ponderada dos seus vizinhos, gerando rotulações mais suaves (ZHU; GHAMRANI; LAFFERTY, 2003).
- *OMNI-Prop* (OMNI): parte da ideia que se os vizinhos de um nó possuem um rótulo, esse nó também tem esse rótulo. Além disso, não parte do pressuposto da rede ser homofílica (nós semelhantes tendem a ter conexão) ou heterofílica (nós semelhantes tendem a não ter uma conexão). Também, considera se as evidências são suficientes antes de rotular um exemplo (YAMAGUCHI; FALOUTSOS; KITAGAWA, 2015).



### 3 Revisão Bibliográfica

Dado a utilidade da classificação de imagens e sua necessidade de uma grande quantidade de dados, utilizar e obter boas técnicas de DA é importante, e isso é refletido pela quantidade de trabalhos que buscam aplicar DA ou propor novas técnicas. A maioria dos trabalhos que buscam obter mais dados rotulados de imagens, optam por aplicar as TIs, e um número menor, mas crescente, de trabalhos optaram por aplicar algoritmos semissupervisionados. Alguns trabalhos serão detalhados a seguir.

(ZHANG et al., 2019) utilizaram DA com três tipos de métodos: rotação de imagem, correção *gamma* e *noise injection*. Nesse trabalho foi utilizado uma CNN como classificador e o *dataset* utilizado era composto por imagens para classificação de frutas. O melhor resultado encontrado foi 94.94%, sendo aproximadamente 5% a mais que os resultados encontrados na literatura.

(HUSSAIN et al., 2017) estudaram estratégias de transformação de imagens no contexto de imagens médicas, como *flips* horizontais, cortes aleatórios, PCA, *etc.* A base de dados utilizada é composta por imagens de massa e não-massa (normal). O trabalho mostrou que a performance aumentou diante o aumento do conjunto de treino. As abordagens de *flips* e filtros gaussianos levaram a acurácia de validação de 84% para 88%.

Já (PAULIN et al., 2014) propuseram um algoritmo chamado *Image Transformation Pursuit* (ITP), para seleção automática de um menor conjunto de técnicas de TIs e que consiga melhores resultados, representando ganhos reais. O ITP se apoia em uma estratégia gulosa, onde a cada iteração seleciona a técnica com melhor ganho na acurácia. Com a base Imagenet, os autores relataram um ganho de aproximadamente 5% na acurácia top-5.

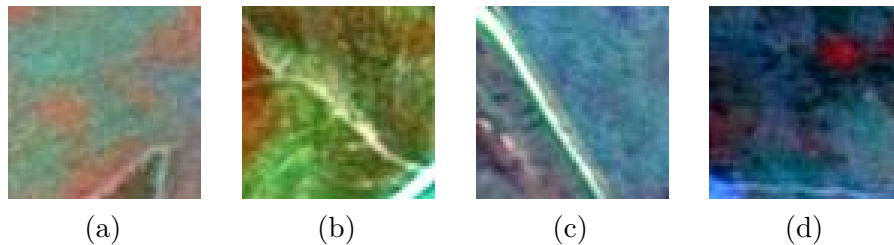
(MRKŠIĆ, 2013) estudou métodos semissupervisionados para DA. O objetivo era investigar até que ponto os métodos baseados em GSSL eram capazes de aumentar os resultados obtidos pelos classificadores e determinar os algoritmos apropriados para cada área. O autor montou um esquema geral para esquemas de DA, aplicou com sucesso em três áreas e usou um esquema para avaliar a qualidade de aplicação de DA para conseguir resultados significativamente melhores.

## 4 Metodologia

### 4.1 Bases de Dados

- *Brazilian Coffee Scenes Dataset*: a base de dados *Brazilian Coffee Scenes Dataset* ([PENATTI K. NOGUEIRA, 2015](#)) é composta por imagens de satélite de plantações de café em cidades de Minas Gerais. O conjunto de dados é composto por 2876 exemplos, sendo 50% exemplos da classe *coffee* e 50% da classe *non coffee*. Na Figura 9 são apresentados alguns exemplos das imagens capturadas pelo sensor.

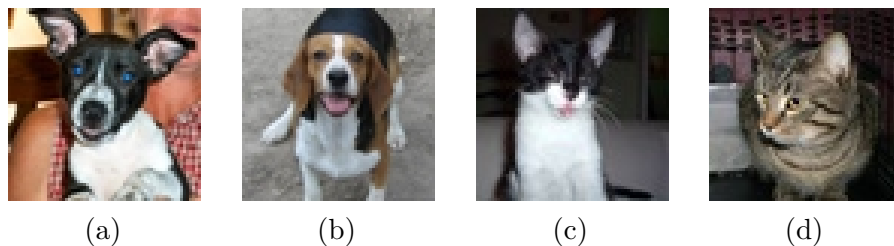
Figura 9 – Exemplo de imagens contidas na base de dados.



Fonte: ([PENATTI K. NOGUEIRA, 2015](#)).

- *Dogs vs Cats*: a base de dados *Dogs vs Cats* é uma base contendo imagens de cachorros e gatos, disponibilizada no *Kaggle* ([KAGGLE, 2015](#)). Nesse caso, usou-se 10.000 exemplos para treino (50% cada classe) e 2.000 exemplos para teste (50% cada classe). Na Figura 10 são apresentado alguns exemplos.

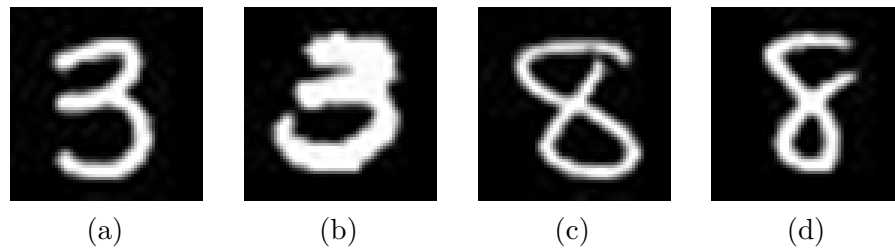
Figura 10 – Exemplo de imagens contida na base de dados.



Fonte: ([KAGGLE, 2015](#)).

- *MNIST DATABASE of handwritten digits*: a MNIST ([Lecun et al., 1998](#)) é composta por imagens de dígitos escritos à mão. Neste caso, usou-se apenas exemplos dos dígitos 3 e 8, sendo 5.600 exemplos pra treino e 1.400 para teste (50% para cada classe). Na Figura 11, temos alguns exemplos.

Figura 11 – Exemplo de imagens contidas na MNIST.



Fonte: (Lecun et al., 1998).

## 4.2 Softwares e Bibliotecas

O Python 3.6 será a principal ferramenta para a realização deste projeto. Sendo utilizado as bibliotecas TensorFlow<sup>1</sup> e Keras<sup>2</sup> para a extrair características, aplicar TIs e CNNs. O scikit-learn<sup>3</sup> para uso do classificador SVM e o NumPy<sup>4</sup> para vetores e matrizes.

## 4.3 Experimentação

- Fase 1: Cada base de dados é separada aleatoriamente em conjunto de treino e teste.
- Fase 2: Nessa fase, as abordagens se diferenciam, conforme as sub-etapas:
  - TIs: o conjunto de treino é dividido entre 5 conjuntos de dados disjuntos, cada um com 10% dos dados originais em relação ao tamanho da base de treino. Aplica-se TIs nesses conjuntos até atingir seu tamanho original e extrai-se as características, gerando 5 bases de treino aumentadas.
  - GSSL: extrai-se a características dos conjuntos de treino, gerando uma base de treino e uma de teste. A partir da base treino, gera-se 5 novas bases de treino diferentes, considerando o rótulo de apenas 10% dos rótulos desses conjuntos. Em seguida, aplica-se um algoritmo GSSL para rotular os outros 90%, gerando as bases de treino aumentadas.
- Fase 3: para cada base/conjunto aumentado, aplica-se no classificador SVM e assim tem-se uma acurácia.
- Fase 4: ao final, será avaliado e comparado cada técnica, calculando os resultados médios.

---

<sup>1</sup> <https://www.tensorflow.org/>

<sup>2</sup> <https://keras.io/>

<sup>3</sup> <https://scikit-learn.org/stable/>

<sup>4</sup> <https://www.numpy.org/>

## 5 Resultados

### 5.1 Experimentos com Transformação de Imagens (TIs)

A Tabela 2 apresenta os resultados obtidos com a técnica de TIs nos conjuntos.

Tabela 2 – Resultados obtidos nas bases de dados com as TIs.

|                     | Conjunto 1 | Conjunto 2 | Conjunto 3 | Conjunto 4 | Conjunto 5 | Média        |
|---------------------|------------|------------|------------|------------|------------|--------------|
| <i>Coffee</i>       |            |            |            |            |            |              |
| ResNet50            | 88,57      | 86,84      | 86,95      | 84,53      | 89,84      | <b>87,35</b> |
| VGG16               | 78,75      | 77,25      | 78,64      | 78,64      | 83,83      | 79,42        |
| VGG19               | 81,29      | 78,52      | 79,91      | 79,45      | 86,14      | 81,06        |
| Xception            | 76,67      | 75,75      | 78,29      | 75,64      | 81,29      | 77,53        |
| <i>Dogs vs Cats</i> |            |            |            |            |            |              |
| ResNet50            | 97,70      | 98,10      | 97,75      | 97,90      | 97,80      | <b>97,85</b> |
| VGG16               | 97,50      | 97,00      | 97,60      | 97,45      | 97,80      | 97,47        |
| VGG19               | 97,80      | 97,40      | 97,75      | 97,60      | 97,45      | 97,60        |
| Xception            | 64,85      | 65,80      | 64,80      | 64,90      | 66,35      | 65,34        |
| <b>MNIST</b>        |            |            |            |            |            |              |
| ResNet50            | 98,40      | 97,91      | 97,79      | 97,54      | 97,30      | <b>97,79</b> |
| VGG16               | 95,58      | 95,45      | 95,58      | 94,59      | 95,33      | 95,31        |
| VGG19               | 96,31      | 96,31      | 95,95      | 95,21      | 96,56      | 96,07        |
| Xception            | 89,56      | 90,54      | 91,28      | 90,91      | 92,26      | 90,91        |

### 5.2 Experimentos com os Algoritmos Semissupervisionados (GSSL)

As Tabelas 3, 4, 5 e 6 apresentam os resultados obtidos com os algoritmos LP, LGC, GFHF e OMNI, respectivamente, nos conjuntos.

Tabela 3 – Resultados obtidos nas bases de dados com o algoritmo LP.

|                     | Conjunto 1 | Conjunto 2 | Conjunto 3 | Conjunto 4 | Conjunto 5 | Média        |
|---------------------|------------|------------|------------|------------|------------|--------------|
| <i>Coffee</i>       |            |            |            |            |            |              |
| ResNet50            | 80,95      | 79,21      | 78,06      | 81,52      | 79,21      | <b>79,79</b> |
| VGG16               | 68,59      | 68,36      | 67,09      | 71,82      | 67,09      | 68,59        |
| VGG19               | 78,41      | 67,79      | 77,14      | 69,52      | 67,21      | 72,01        |
| Xception            | 67,67      | 66,28      | 66,74      | 66,74      | 66,51      | 66,79        |
| <i>Dogs vs Cats</i> |            |            |            |            |            |              |
| ResNet50            | 98,10      | 98,35      | 97,95      | 97,85      | 98,10      | <b>98,07</b> |
| VGG16               | 97,55      | 97,40      | 97,40      | 97,60      | 97,35      | 97,46        |
| VGG19               | 97,60      | 97,40      | 97,60      | 97,50      | 97,65      | 97,55        |
| Xception            | 55,85      | 54,25      | 56,75      | 56,15      | 55,15      | 55,63        |
| <b>MNIST</b>        |            |            |            |            |            |              |
| ResNet50            | 97,30      | 97,42      | 97,17      | 96,93      | 97,17      | <b>97,20</b> |
| VGG16               | 96,93      | 96,68      | 96,44      | 96,44      | 96,44      | 96,59        |
| VGG19               | 96,93      | 97,17      | 96,93      | 96,68      | 97,05      | 96,95        |
| Xception            | 79,48      | 72,23      | 71,50      | 73,59      | 82,56      | 75,87        |

Tabela 4 – Resultados obtidos nas bases de dados com o algoritmo LGC.

|                     | Conjunto 1 | Conjunto 2 | Conjunto 3 | Conjunto 4 | Conjunto 5 | Média        |
|---------------------|------------|------------|------------|------------|------------|--------------|
| <i>Coffee</i>       |            |            |            |            |            |              |
| ResNet50            | 81,18      | 82,33      | 82,33      | 77,23      | 83,60      | <b>81,33</b> |
| VGG16               | 72,51      | 67,55      | 71,48      | 69,98      | 73,09      | 70,92        |
| VGG19               | 69,40      | 67,55      | 76,10      | 67,21      | 60,51      | 68,15        |
| Xception            | 72,17      | 72,29      | 69,75      | 71,48      | 72,40      | 71,62        |
| <i>Dogs vs Cats</i> |            |            |            |            |            |              |
| ResNet50            | 98,40      | 98,50      | 98,45      | 98,55      | 98,40      | <b>98,46</b> |
| VGG16               | 98,15      | 98,10      | 98,05      | 98,20      | 98,45      | 98,19        |
| VGG19               | 98,40      | 97,70      | 98,15      | 97,90      | 98,15      | 98,06        |
| Xception            | 59,50      | 61,60      | 61,10      | 61,50      | 60,40      | 60,82        |
| <b>MNIST</b>        |            |            |            |            |            |              |
| ResNet50            | 98,89      | 99,01      | 98,89      | 98,40      | 99,01      | <b>98,84</b> |
| VGG16               | 98,28      | 97,91      | 97,91      | 98,65      | 97,91      | 98,13        |
| VGG19               | 98,16      | 98,53      | 98,40      | 98,16      | 98,40      | 98,33        |
| Xception            | 93,00      | 92,26      | 89,56      | 89,80      | 92,01      | 91,33        |

Tabela 5 – Resultados obtidos nas bases de dados com o algoritmo GFHF.

|                     | Conjunto 1 | Conjunto 2 | Conjunto 3 | Conjunto 4 | Conjunto 5 | Média        |
|---------------------|------------|------------|------------|------------|------------|--------------|
| <i>Coffee</i>       |            |            |            |            |            |              |
| ResNet50            | 83,14      | 84,30      | 85,57      | 84,87      | 82,33      | <b>84,04</b> |
| VGG16               | 79,79      | 68,13      | 70,32      | 72,86      | 73,67      | 72,95        |
| VGG19               | 77,02      | 75,40      | 79,68      | 77,60      | 72,63      | 76,47        |
| Xception            | 73,21      | 72,86      | 73,44      | 74,48      | 73,09      | 73,42        |
| <i>Dogs vs Cats</i> |            |            |            |            |            |              |
| ResNet50            | 98,35      | 98,45      | 98,35      | 98,30      | 98,40      | <b>98,37</b> |
| VGG16               | 98,05      | 98,15      | 98,35      | 98,00      | 98,35      | 98,18        |
| VGG19               | 97,95      | 97,80      | 97,85      | 97,75      | 98,00      | 97,87        |
| Xception            | 61,00      | 62,70      | 62,80      | 62,25      | 62,90      | 62,33        |
| <b>MNIST</b>        |            |            |            |            |            |              |
| ResNet50            | 98,65      | 98,40      | 98,40      | 98,03      | 98,40      | <b>98,38</b> |
| VGG16               | 97,67      | 97,54      | 97,79      | 97,67      | 97,79      | 97,69        |
| VGG19               | 97,42      | 97,30      | 97,79      | 97,42      | 97,42      | 97,47        |
| Xception            | 91,89      | 91,03      | 90,91      | 90,66      | 91,65      | 91,23        |

Tabela 6 – Resultados obtidos nas bases de dados com o algoritmo OMNI.

|                     | Conjunto 1 | Conjunto 2 | Conjunto 3 | Conjunto 4 | Conjunto 5 | Média        |
|---------------------|------------|------------|------------|------------|------------|--------------|
| <i>Coffee</i>       |            |            |            |            |            |              |
| ResNet50            | 84,18      | 81,18      | 81,99      | 81,99      | 83,37      | <b>82,54</b> |
| VGG16               | 64,78      | 68,94      | 71,02      | 60,28      | 69,63      | 66,93        |
| VGG19               | 72,17      | 67,44      | 70,09      | 65,94      | 68,48      | 68,82        |
| Xception            | 73,21      | 73,67      | 72,06      | 72,06      | 73,21      | 72,84        |
| <i>Dogs vs Cats</i> |            |            |            |            |            |              |
| ResNet50            | 98,40      | 98,30      | 98,25      | 98,45      | 98,40      | <b>98,36</b> |
| VGG16               | 98,40      | 98,30      | 98,05      | 97,95      | 98,25      | 98,19        |
| VGG19               | 97,75      | 97,90      | 97,70      | 97,85      | 97,90      | 97,82        |
| Xception            | 61,50      | 64,45      | 62,45      | 61,90      | 61,00      | 62,26        |
| <b>MNIST</b>        |            |            |            |            |            |              |
| ResNet50            | 98,40      | 98,03      | 98,16      | 98,28      | 98,28      | <b>98,23</b> |
| VGG16               | 98,03      | 98,16      | 97,67      | 97,54      | 97,54      | 97,79        |
| VGG19               | 97,54      | 97,30      | 97,54      | 97,17      | 97,30      | 97,37        |
| Xception            | 90,66      | 88,70      | 90,42      | 90,91      | 87,59      | 89,66        |

### 5.3 Comparação

Para comparar os resultados, calculou-se os resultados médio de cada técnica em relação a média dos 5 conjuntos de cada uma. Também, foram feitos experimentos com a base original (sem aplicação de DA e 5-fold), que podem ser visualizados na Tabela 7.

Tabela 7 – Resultados médio para as técnicas.

|                     | Original | TIs   | LP    | LGC   | GFHF  | OMNI  |
|---------------------|----------|-------|-------|-------|-------|-------|
| <i>Coffee</i>       | 84,59    | 81,34 | 71,80 | 73,01 | 76,72 | 72,78 |
| <i>Dogs vs Cats</i> | 91,41    | 89,57 | 87,18 | 88,88 | 89,19 | 89,18 |
| MNIST               | 98,22    | 95,08 | 91,65 | 96,66 | 96,19 | 95,76 |
| Média               | 91,41    | 88,66 | 83,54 | 86,18 | 87,37 | 85,91 |

### 5.4 Discussão

A analisar os resultados observa-se, em geral, bons resultados para as duas abordagens de DA utilizadas. No entanto, a abordagem de GSSL (especificamente, o algoritmo LGC e GFHF) se saiu um pouco melhor com a base de dados MNIST, enquanto que a abordagem de TIs se saiu melhor com a base *Brazilian Coffee Scenes Dataset*. Na base *Dog vs Cats* as duas abordagens ficaram com resultados bens parecidos.

Em relação aos algoritmos GSSL, o GFHF obteve melhores resultados em geral, com média de 87,37%, enquanto o algoritmo LP obteve piores resultados, com média 83,54%. Já com os descritores de imagens, a ResNet50 obteve resultados superiores em relação as três outras CNNs utilizadas. A Xception obteve o pior resultado, em geral.

Observa-se também que a simulação de aplicação de DA no conjunto de dados pequeno até atingir o tamanho original teve bons efeitos, chegando próximo ao resultado da base original. Na *Dog vs Cats* a abordagem de TIs e o algoritmo GFHF atingiram 89,57% e 89,19% de acurácia respectivamente, próximo da original, que atingiu 91,41%. Para a MNIST a média da abordagem de TIs e o algoritmo LGC foram 95,08% e 96,66%, respectivamente, enquanto a original 98,22%.

Dessa forma, é indicado que esses algoritmos GSSL podem ser utilizados como propagadores de rótulos, em caso de aplicações semissupervisionadas, para melhorar os resultados gerados pelo classificador. Além disso, serviria, por exemplo, para automatização de rotulagem, já que esse é um processo trabalho e custoso, já que exige esforço para anotar cada exemplo manualmente e também exige um especialista para rotular.

## 6 Conclusão

Com a realização deste trabalho percebe-se a grande importância de se obter boas técnicas de DA para aumentar a quantidade de exemplos rotulados e conseguir melhores classificadores, já que a classificação possui inúmeras aplicações.

Assim, o objetivo deste trabalho foi aplicar e comparar algoritmos semissupervisionados que propagam informação de rótulo a exemplos não rotulados. Foram utilizados 4 algoritmos diferentes e pode-se concluir que todos os algoritmos foram eficientes em suas tarefas, gerando bons resultados, porém, o algoritmo *Gaussian Fields and Harmonic Functions* se sobressaiu sobre os outros.

Além disso, buscou-se aplicar e avaliar as *Convolutional Neural Networks* como extratores de características de imagens. As redes ResNet50, VGG16 e VGG19 se saíram bem para quase todos os experimentos, com destaque para a ResNet50 que obteve melhores resultados, em geral. A Xception, por sua vez, obteve os piores resultados.

Os experimentos de classificação foram realizados pelo classificador SVM com as bases de dados aumentadas e a base original. Os resultados demonstraram que a aplicação de DA foi efetiva nas bases simuladas, usando apenas 10% dos dados originais chegou-se a resultados bem próximos da base original.

Dessa forma, pode-se verificar a efetividade desses algoritmos como forma de se obter mais dados rotulados para conseguirmos melhores resultados na classificação. Os quais podem ser aplicados na tarefa de se obter mais dados rotulados de forma automatizada, excluindo o processo manual de rotular cada dado, que denota tempo, esforço e, conseqüentemente, custos.

Como trabalho futuro, pode-se testar novos algoritmos semissupervisionados. Ou ainda, testar novas formas de gerar um grafo para propagar os rótulos, visto que isso é uma característica que pode influenciar no processo de rotular os dados.

# Referências

CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning (chappelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542–542, 2009. Citado na página 6.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 9.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016. Citado na página 12.

FACELI, K. et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. 2011. Citado na página 8.

HAYKIN, S. *Neural Networks and Learning Machines*. Pearson Education, 2011. ISBN 9780133002553. Disponível em: <<https://books.google.com.br/books?id=faouAAAAQBAJ>>. Citado 2 vezes nas páginas 10 e 11.

HUSSAIN, Z. et al. Differential data augmentation techniques for medical imaging classification tasks. In: AMERICAN MEDICAL INFORMATICS ASSOCIATION. *AMIA Annual Symposium Proceedings*. [S.l.], 2017. v. 2017, p. 979. Citado 2 vezes nas páginas 4 e 16.

KAGGLE. *Dogs vs. Cats*. 2015. Disponível em: <<https://www.kaggle.com/c/dogs-vs-cats/data>>. Citado na página 17.

Lecun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219. Citado 4 vezes nas páginas 9, 12, 17 e 18.

LINNAINMAA, S. The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. *Master's Thesis (in Finnish), Univ. Helsinki*, p. 6–7, 1970. Citado na página 11.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado na página 9.

LORENA, A. C.; CARVALHO, A. C. P. d. L. et al. Introdução às máquinas de vetores suporte (support vector machines). 2003. Citado na página 10.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 10.

MICHALSKI, R. S.; TECUCI, G. *Machine learning: A multistrategy approach*. [S.l.]: Morgan Kaufmann, 1994. Citado na página 6.

MITCHELL, T. et al. Machine learning. *Annual review of computer science*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 4, n. 1, p. 417–433, 1990. Citado na página 4.



- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, v. 1, n. 1, p. 32, 2003. Citado 3 vezes nas páginas 4, 6 e 7.
- MRKŠIĆ, N. Semi-supervised learning methods for data augmentation computer science tripos trinity college may 15, 2013. 2013. Citado 2 vezes nas páginas 13 e 16.
- PAULIN, M. et al. Transformation pursuit for image classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 3646–3653. Citado na página 16.
- PENATTI K. NOGUEIRA, J. A. d. S. O. A. B. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? 2015. Citado 2 vezes nas páginas 8 e 17.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 11.
- TORREY, L.; SHAVLIK, J. Transfer learning. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. [S.l.]: IGI Global, 2010. p. 242–264. Citado na página 13.
- VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: *Proceedings of the XXIX Conference on Graphics, Patterns and Images*. [S.l.: s.n.], 2016. p. 1–4. Citado 2 vezes nas páginas 8 e 12.
- WANG, S.-H. et al. Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling. *Journal of medical systems*, Springer, v. 42, n. 1, p. 2, 2018. Citado 2 vezes nas páginas 5 e 8.
- YAMAGUCHI, Y.; FALOUTSOS, C.; KITAGAWA, H. Omni-prop: Seamless node classification on arbitrary label correlation. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. Citado na página 15.
- ZHANG, Y.-D. et al. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, Springer, v. 78, n. 3, p. 3613–3632, 2019. Citado na página 16.
- ZHOU, D. et al. Learning with local and global consistency. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2004. p. 321–328. Citado na página 15.
- ZHU, X.; GHAHRAMANI, Z. *Learning from labeled and unlabeled data with label propagation*. [S.l.], 2002. Citado na página 15.
- ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In: *Proceedings of the 20th International conference on Machine learning (ICML-03)*. [S.l.: s.n.], 2003. p. 912–919. Citado na página 15.