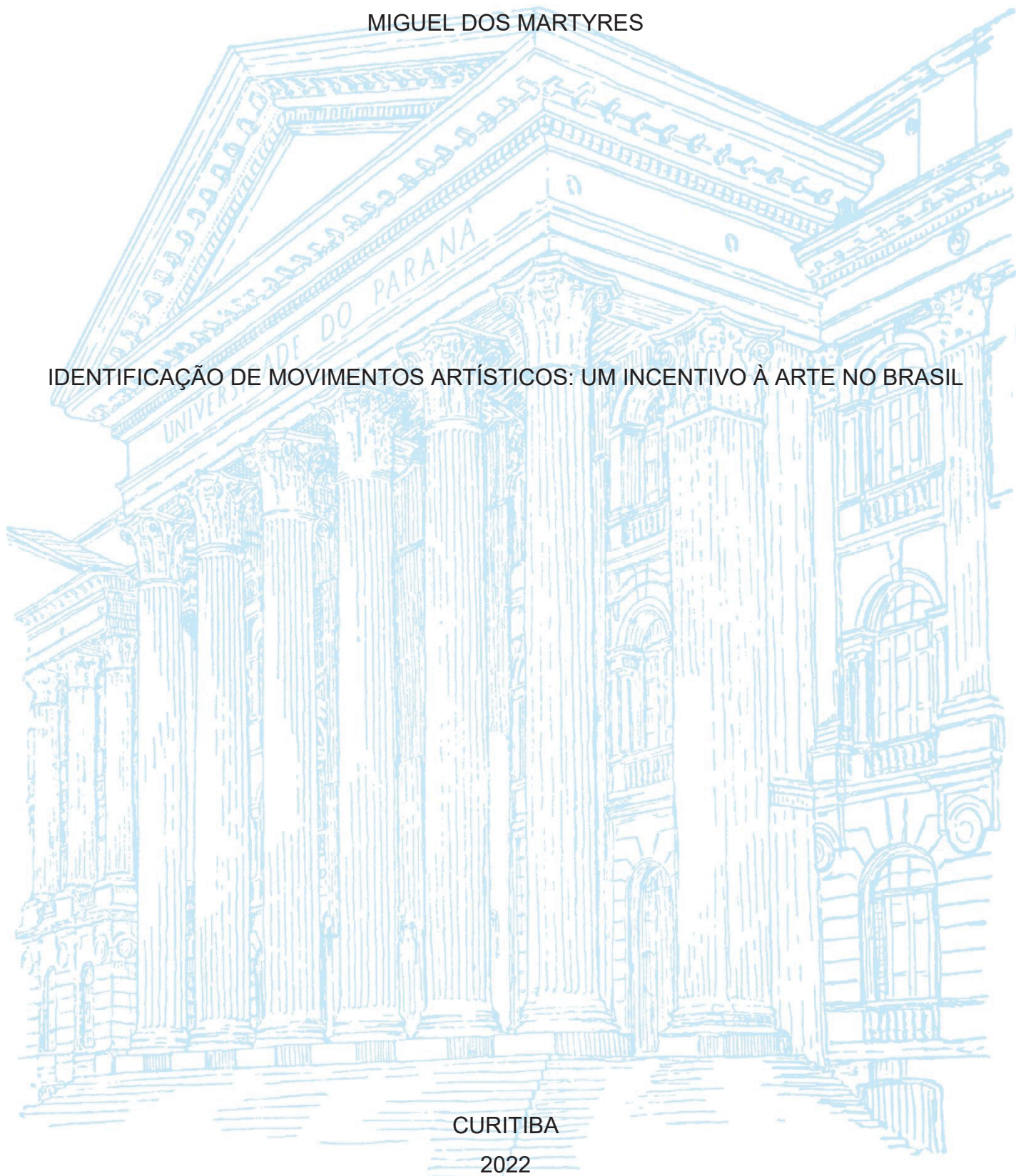


UNIVERSIDADE FEDERAL DO PARANÁ

MIGUEL DOS MARTYRES

IDENTIFICAÇÃO DE MOVIMENTOS ARTÍSTICOS: UM INCENTIVO À ARTE NO BRASIL



CURITIBA

2022

MIGUEL DOS MARTYRES

IDENTIFICAÇÃO DE MOVIMENTOS ARTÍSTICOS: UM INCENTIVO À ARTE NO BRASIL

Trabalho de Conclusão de Curso apresentada ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador(a): Prof(a). Dr(a). Razer Anthom Nizer Rojas Montano.

CURITIBA

2022

## TERMO DE APROVAÇÃO

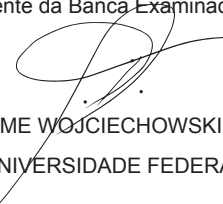
Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **MIGUEL DOS MARTYRES** intitulada: **Identificação de movimentos artísticos: um incentivo à arte no Brasil**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 19 de Maio de 2022.



RAZER ANTHON NIZER ROJAS MONTAÑO  
Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

# Identificação de movimentos artísticos: um incentivo à arte no Brasil

Miguel dos Martyres  
*Especialização em Inteligência Artificial Aplicada*  
*Universidade Federal do Paraná (UFPR)*  
Curitiba, Brasil  
miguelmartyres@gmail.com

Razer Anthon Nizer Rojas Montano  
*Especialização em Inteligência Artificial Aplicada*  
*Universidade Federal do Paraná (UFPR)*  
Curitiba, Brasil  
razer@ufpr.br

**Resumo**— O acesso à arte e a cultura são direitos de todo ser humano. A arte é o entendimento de como fazemos parte do mundo. É a manifestação do pensamento humano, por meio de estética, comunicação ou emoções. Este artigo tem por objetivo identificar o potencial de se utilizar *Convolutional Neural Networks (CNN)*, para categorizar o movimento artístico ao qual as pinturas e desenhos tem a maior probabilidade de pertencer.

**Palavras chave**— *cnn, arte, movimento artístico*

**Abstract**— Access to art and the culture are the rights of every human being. Art is the understanding of how we are part of the world. It is the manifestation of human thoughts, through aesthetics, communication or emotions. This work aims to identify the potential of *Convolutional Neural Networks (CNN)*, to categorize the artistic movement to which paintings and drawings are most likely to belong.

**Keywords**— *cnn, art, artistic movement*

## I. INTRODUÇÃO

O filósofo americano e pesquisador de estética da arte, Morris Weitz, acredita que é impossível definir o conceito do que é a arte [1]. Isso se dá, pois, a produção artística e o conceito da arte se transformam ao longo da história da humanidade.

Ao levar isso em consideração, afirma-se que o conceito de arte é subjetivo, leva a discussões e incita a imaginação. Esses debates têm que ser incentivados pelas escolas e plataformas tecnológicas que têm-se hoje.

De acordo Andrade e Adriana Rocha Vilela Arantes [2], “A Arte no Brasil ao longo do tempo foi crescendo consideravelmente de acordo com momentos históricos e correntes pedagógicas, principalmente no que tange às mudanças da arte na tendência tradicional contemporânea.”.

Com base nessa pesquisa, conclui-se que o esforço pedagógico para incluir a arte no ensino surtiu resultados, uma vez que o ensino da arte nas escolas desde a educação básica ao ensino médio é obrigatório [3].

Nesse sentido, mesmo com a obrigatoriedade do ensino da arte em escolas, não existem dados que avaliem a qualidade desse ensino. Mas têm-se outros dados que indicam um déficit no incentivo à arte. Segundo Faria [4], em 2009 o MAM (Museu de Arte Moderna de São Paulo) recebeu 168.466 visitantes. Em contraste, o museu do Prado, em Madrid, Espanha, recebeu 2.598.016 visitantes no ano de 2008. Da mesma forma, de acordo com o mesmo artigo, somente 4,3% dos visitantes dos museus no Brasil estão na faixa etária de 14 a 19 anos.

Além disso uma pesquisa do IPEA (Instituto de Pesquisa Econômica Aplicada) realizada em 2010 [5] revelou que somente 30% dos brasileiros já visitaram algum museu.

Sendo assim, este artigo aborda a interpretação de pinturas e desenhos de diferentes movimentos artísticos com o objetivo de incentivar a arte. Para atingir isso, a proposta é criar um

modelo classificatório que consiga interpretar pinturas e desenhos, e categorizá-los no movimento artístico ao qual a obra teria mais chance de pertencer.

O resultado é um modelo de aprendizado de máquina com o objetivo de incentivar o interesse no estudo da arte, incluir os estudantes como parte desse processo e também facilitar a assimilação de alguns conteúdos dentro do ensino da história da arte.

Para isso as imagens utilizadas no treinamento do modelo foram delimitadas entre obras dos movimentos artísticos a partir do Renascimento (século XIV – século XVI) até as vanguardas modernistas (século XIX – século XX).

Isso porque com o desenvolvimento da produção artística contemporânea na metade do século XX, a classificação em movimentos artísticos começa a ficar cada vez mais complexa, até que nos dias de hoje não temos mais como enquadrar as obras em movimentos artísticos distintos.

Da mesma forma, foi realizada a delimitação das obras que foi utilizada para o treinamento do modelo na região da Europa – com algumas exceções, já que movimentos artísticos se desenvolveram de maneiras diferentes e até independentes em outros continentes.

### A. Motivações

A cultura e a arte têm que ser preservadas e incentivadas. Hoje no Brasil, mesmo com a obrigatoriedade da inclusão das artes nas escolas, percebe-se que a busca pela educação artística é baixa. De acordo com os dados fornecidos em 2010 pelo IPEA (Instituto de Pesquisa Econômica Aplicada) [5] 70% dos brasileiros nunca visitaram um museu.

Com o avanço da tecnologia e o advento da internet, naturalmente a divulgação artística ficou mais fácil. Uma obra, que outrora precisaria de uma exposição e um público para ser divulgada, atualmente, ao postá-la em alguma rede social como o Instagram, pode ter uma propagação maior, sem precisar sair de casa.

Da mesma forma, a arte e a criatividade podem ser incentivadas por meio da tecnologia. Desde filtros de obras renascentistas no Instagram, a jogos online onde o usuário pode desenhar em tempo real.

O modelo classificatório de pinturas e desenhos que este artigo propõe, pode incentivar desde pessoas que não tem o hábito de pintar, a ilustradores e pintores formados. Já que, ao inserir uma imagem de sua obra no modelo classificatório, o usuário terá uma perspectiva do movimento artístico que seu feito poderia ter pertencido. Com isso, pode-se descobrir sobre todo o contexto histórico e técnico do porquê sua obra foi categorizada como tal.

## B. Objetivos

O presente artigo tem o objetivo de desenvolver um modelo de classificação que receba como entrada pinturas e desenhos. Após processar essas imagens e extrair suas características, o modelo classificatório irá retornar a qual movimento artístico a obra informada teria maior probabilidade de pertencer.

Para criar o modelo classificatório são primariamente utilizados modelos de *Convolutional Neural Networks (CNN)* com configurações distintas. Também é utilizado o conceito de *Transfer Learning*, para usar as vantagens de redes neural já treinadas para aumentar a acurácia do modelo proposto.

## C. Organização do artigo

Este artigo é composto por 5 capítulos. No capítulo II de Fundamentação Teórica serão apresentados dois assuntos principais. Primeiro, será realizada uma breve introdução sobre os movimentos artísticos, e a escolha de utilizar somente dos principais movimentos artísticos. Em seguida, no capítulo III serão apresentados os principais conceitos das técnicas usadas neste trabalho, aprofundando-se nos tópicos mais relevantes para o resultado final que este trabalho propõe.

O capítulo III detalha as bases de dados utilizados nos experimentos e inicia uma discussão acerca da escolha dos dados utilizados. Em seguida serão detalhados os modelos e arquitetura das CNNs utilizadas neste artigo, assim como o uso da transferência de aprendizado. Após isso, será explicado as métricas de como os resultados serão avaliados.

Com isso, os resultados dos modelos classificatórios com diferentes configurações e arquiteturas serão comparados e analisados no capítulo IV. Por fim, as considerações finais, juntamente com a conclusão do melhor modelo treinado, se foi obtido um resultado aceitável pelo modelo e os trabalhos futuros serão discutidos no capítulo V.

## II. FUNDAMENTAÇÃO TEÓRICA

Existem poucos estudos voltados a reconhecimento de obras arte. Porém há uma grande quantidade de análises voltadas a reconhecimento de padrões em imagens. Como por exemplo o artigo de Baheti *et al.* [6], que, com base em um banco de dados de ruas e rodovias não estruturadas da Índia, os autores criaram um modelo de aprendizado de máquina, com a função de segmentar as diversas informações das vias e estradas da Índia de forma coesa. Do mesmo modo, Badrinarayanan *et al.* [7] e Bappy *et al.* [8] contam com a arquitetura *encoder-decoder* para extrair características, segmentar e categorizar as imagens.

Apesar de não existirem muitos estudos sobre categorização de obras de arte, este trabalho propõe-se a utilizar conceitos de reconhecimento de imagens e padrões para classificar pinturas e desenhos de forma eficaz.

No decorrer deste capítulo, serão apresentadas as definições de conceitos empregados na elaboração do modelo classificatório que este artigo propõe. Desde uma abordagem geral, descrevendo o que são os movimentos artísticos a estudos aprofundados sobre conceitos de Aprendizado de Máquina e Deep Learning.

### A. Movimentos Artísticos

Movimentos artísticos podem ser definidos como um grupo de obras em diferentes linguagens (pintura, escultura, gravura, literatura, teatro e afins) que compartilham os mesmos ideais artísticos, técnica e período na história.

As regras para classificar uma obra em um determinado movimento artístico são subjetivas, já que é preciso analisar

tudo o contexto histórico de onde, quando e por quem a obra foi criada, e ainda qual técnica foi utilizada.



Fig. 1. À esquerda a obra “Vieux guitariste aveugle” (1903) e à direita a pintura “Portrait de Dora Maar” (1937), ambas por Pablo Picasso. Fonte: adaptado de WIKIART [24].

Levando isto em consideração, é importante salientar que o mesmo artista não necessariamente faz parte de um único movimento artístico. Um bom exemplo disto é o artista espanhol Pablo Picasso, conhecido mundialmente por ser o fundador do cubismo, contudo, no início de sua carreira, suas obras tinham um estilo consideravelmente diferente do que ele desenvolveu ao longo dos anos seguintes, e que o fizeram famoso. Pode-se visualizar isto na figura 1.

Em suma, movimentos artísticos não podem ser classificados utilizando como referência somente um critério, pois isso pode levar a conclusões equivocadas. Assim como muitos dos conceitos empregados nas obras são subjetivas. Esses são alguns dos motivos pelos quais é complexo treinar um modelo de aprendizado de máquina capaz de classificar obras em seus respectivos movimentos artísticos.

Os movimentos artísticos utilizados para este artigo são:

#### 1) Renascimento (séc. XIV-XVI)

O Renascimento foi um movimento artístico que surgiu na Europa no século XIV, e que caracterizou a transição da Idade Média para a Idade Moderna com suas mudanças culturais e científicas.

A partir do Renascimento veio também a individualidade na arte, tanto de autoria, quanto das imagens retratadas. No sentido de que, pela primeira vez passou-se a ser comum os próprios artistas assinarem suas obras e as pessoas retratadas nas imagens terem uma identidade.

Além disso, com o advento do Renascimento houve uma busca mais aprofundada pelo estudo científico em outras áreas, como física, medicina, navegação, cartografia e astronomia. Naturalmente, os avanços desses estudos também refletiram na produção artística da época, principalmente em relação à medicina e aos estudos do corpo humano.

Apesar disso, as obras deste período ainda estavam muito ligadas à Igreja Católica. Por isso, a grande maioria das obras buscavam representar mitos, passagens bíblicas e a ilustração do que a Igreja Católica queria passar aos seus fiéis, já que a alfabetização era restrita à membros da Igreja e alguns nobres na Europa.



Fig. 2. “A criação de Adão” por Michelangelo (1510). Fonte: adaptado de WIKIART [24].

Dentre as características visuais das obras deste movimento artístico, existem o humanismo nas representações de figuras humanas, com uma fidelidade visual grande (naturalismo) e os retratos de situações cotidianas. Além disso, o Renascimento contava com esculturas em mármore e uma grande quantidade de afrescos e pouquíssimas pinturas de paisagens isoladas.

### 2) Barroco (séc. XVI-XVIII)

O Barroco foi um movimento artístico que surgiu na Europa em meados do século XVI. Assim como o Renascimento, que tinha fortes ligações com a Igreja Católica, o Barroco tinha ligações ainda mais estreitas, pois este movimento surgiu no contexto em que começou a acontecer a Reforma Protestante e, com isso, a Igreja Católica precisava reafirmar seu poder para com seus fiéis.



Fig. 3. “Judite decapitando Holofernes” por Artemisia Gentileschi (1620). Fonte: adaptado de WIKIART [24].

As obras do Barroco são representadas por dualidades, tanto em no conceito quanto no estético. Como por exemplo o sagrado e profano, fé e razão e, luz e sombra. A última sendo a origem para uma das técnicas mais características do

Barroco italiano: o *chiaroscuro*. Essa técnica se dá ao contraste forte entre luz e sombra, o que resultava em um fundo da pintura muito escuro (muitas vezes preto) e elementos específicos na obra muito claros (com muita luz), onde o autor gostaria que o olho do observador focasse.

### 3) Romantismo (séc. XVIII-XIX)

O Romantismo foi um movimento artístico que surgiu na Europa no final do século XVIII. Em contraponto às escolas artísticas do Renascimento e Barroco, que buscavam a objetividade e razão, os artistas romantistas eram preocupados com a retratação dos aspectos subjetivos da experiência e, da existência humana. Para isso, as obras romantistas quase sempre retratavam a experiência humana como temática principal.



Fig. 4. “A Liberdade Guiando o Povo” por Eugène Delacroix (1830). Fonte: adaptado de WIKIART [24].

Algumas características visuais das obras romantistas se assemelham bastante com o Barroco: a valorização das cores por meio do contraste entre luz e sombra, composições que valorizam o dinamismo das obras e a busca pelas emoções humanas e representação de acontecimentos históricos.

### 4) Simbolismo (séc. XIX)

O Simbolismo surgiu na França no final do século XIX. Veio como oposição ao naturalismo e racionalidade da época. Seus temas eram místicos, espirituais e subjetivos, uma vez que, para o Simbolismo, tudo podia assumir um significado simbólico, desde a elementos da natureza, até as construções dos homens.

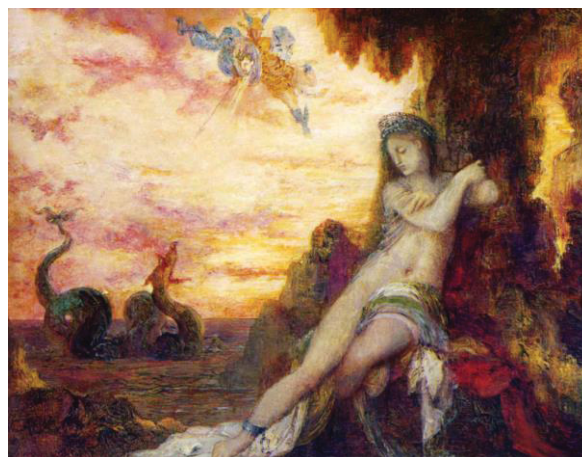


Fig. 5. “Perseus e Andromeda” por Gustave Moreau (1870). Fonte: adaptado de WIKIART [24].

### 5) Impressionismo (séc. XIX)

O Impressionismo foi um movimento artístico que surgiu no final do século XIX na França, e que rompia com as convenções da arte academia da época. Isso porque, o Impressionismo buscava captar as impressões de luminosidade, cor e sombra de paisagens, principalmente.

Dentre as características mais marcantes deste movimento artístico, as que mais se destacavam era o raro uso de tinta preta, e o uso de outras cores para fazer o efeito de luz e sombra, como o azul e marrom.



Fig. 6. “Casas do Parlamento” por Claude Monet (1904). Fonte: adaptado de WIKIART [24].

O Impressionismo ainda não se encaixava como Arte Moderna, mas é só com seu rompimento com a arte acadêmica que foram surgir experimentações artísticas que mais tarde dariam origem às vanguardas modernistas.

### 6) Pós-impressionismo (séc XIX-XX)

O Pós-impressionismo, diferente dos outros movimentos artísticos citados até então, não tinha uma unidade estética entre os diferentes artistas. Assim, o que agrupou os esses artistas pós-impressionistas foram as suas experimentações artísticas mais ousadas em cor, forma e temáticas, sempre ampliando as ideias impressionistas.



Fig. 7. À esquerda “Autorretrato com a Orelha Enfaixada” por Vincent Van Gogh (1889). À direita “Cesta de Maças” por Paul Cézanne (1885). Ambos foram artistas pós-impressionistas. Fonte: adaptado de WIKIART [24].

Os artistas pós-impressionistas também começam a fazer experimentações com distorções espaciais e anatômicas em

suas obras, e isso os diferencia ainda mais do Impressionismo, uma vez que no último, os impressionistas ainda tinham uma preocupação grande em manter essas proporções.

### 7) Cubismo (séc. XX)

O Cubismo foi um movimento artístico que surgiu na Europa no século XX. Diferente dos movimentos artísticos até então, os artistas cubistas representavam em suas obras os objetos em três dimensões, em superfície plana, ao simplificá-las com formas geométricas e linhas retas.

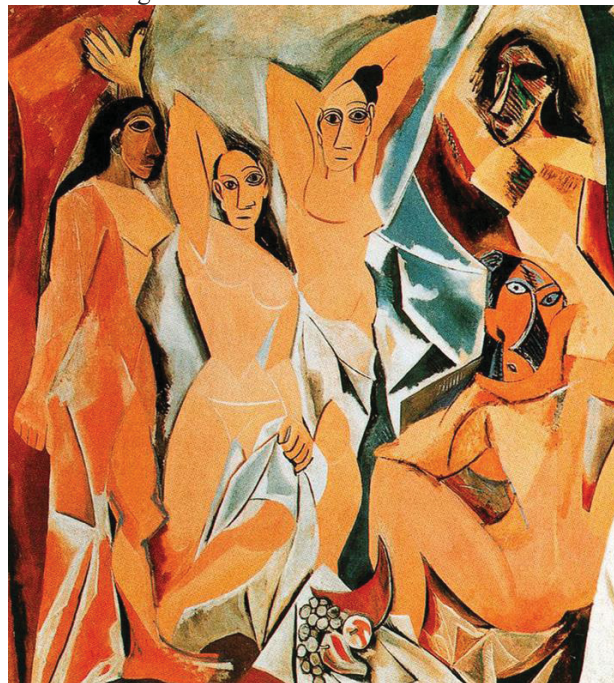


Fig. 8. “Les demoiselles d'Avignon” por Pablo Picasso (1907). Fonte: adaptado de WIKIART [24].

O Cubismo não tenta representar formas e objetos, mas sugeri-los ao espectador, além disso, muitas obras cubistas tentam retratar o dinamismo do movimento com formas geométricas estáticas.

### 8) Expressionismo (séc. XX)

Surgido na Alemanha no século XX, o Expressionismo foi um movimento artístico da vanguarda modernista. Os artistas expressionistas tentavam retratar em suas obras a visão interior ou, melhor dizendo, as emoções que o artista gostaria de expor aos espectadores.

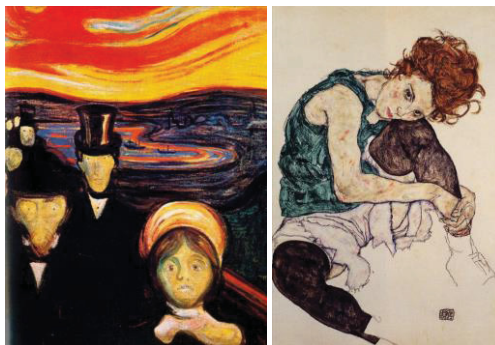


Fig. 9. À esquerda “Ansiedade” por Edvard Munch (1894). À direita “Seated Woman with Bent Knee” por Egon Schiele (1917). Ambos foram artistas expressionistas. Fonte: adaptado de WIKIART [24].

O Expressionismo foi um reflexo de uma sociedade que estava muito próxima da iminente Primeira Guerra Mundial, até chegar na Segunda Guerra Mundial, gerando uma instabilidade emocional muito grande entre as pessoas da época. As obras deste movimento artístico tentam refletir essa angústia e emoção.

### 9) Surrealismo (séc. XX)

Surgida na mesma época que o Expressionismo, o Surrealismo está ligado aos estudos da psicologia de Sigmund Freud (1856-1939), ao enfatizar o subconsciente e as temáticas oníricas em suas obras.

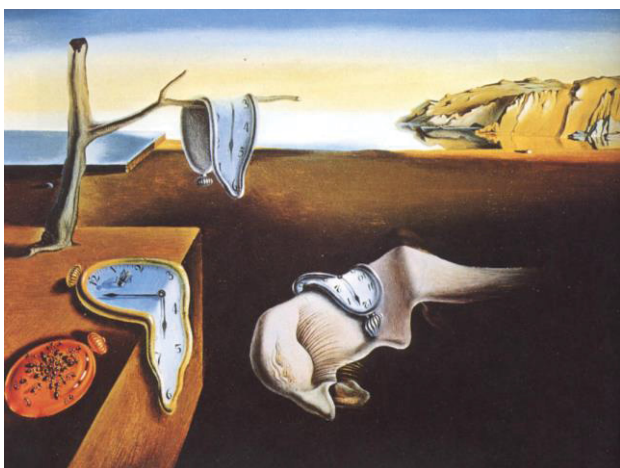


Fig. 10. “A Persistência da Memória” por Salvador Dalí (1931). Fonte: adaptado de WIKIART [24].

Dentre as principais características deste movimento, destacam-se a colagem, a escrita automática, a representação do abstrato e do inconsciente e a não obrigatoriedade com a lógica e a razão.

### B. Aprendizado de máquina

*Machine Learning* (ML) ou, traduzindo para o português, Aprendizado de Máquina, é uma ramificação ou subcampo de estudo dentro da Inteligência Artificial. Ela é responsável por ensinar ao software a execução de uma tarefa, sem depender de trechos de código fixos para tomadas de decisão. Segundo Carleo et al. [9], a proposta geral do ML é reconhecer padrões nos dados, que informam como problemas não vistos são tratados. Como por exemplo os carros autônomos, onde grandes quantidades de dados são captados por sensores, que se tornam decisões de como o automóvel será controlado. Essas escolhas são feitas através de vários algoritmos que aprendem a reconhecer os padrões com base nos dados apresentados.

Tendo isso em mente, para uma melhor noção de quais problemas o ML consegue resolver, podemos dividi-lo em duas categorias principais: aprendizado supervisionado e aprendizado não-supervisionado.

#### 1) Aprendizado supervisionado

O aprendizado supervisionado depende de dados com categorias associadas a eles para o treinamento do modelo de *Machine Learning*. De acordo com Cunningham, Sarah e Delany [10], a característica que define o que é o aprendizado supervisionado é a disponibilidade de dados de treinamento rotulados.

Algoritmos de aprendizado supervisionado são treinados com dados já rotulados, para gerar modelos de ML que aprendam a classificar o mesmo tipo de dado, porém que não tenha uma categoria associada.

Pode-se dividir o aprendizado supervisionado em duas tarefas: classificação e regressão:

- A classificação tem o objetivo de descobrir a função mais aproximada para separar os dados em classes de saída. Essas variáveis de saída geralmente são chamadas de categorias ou classes. A função mapeada prevê a categoria para uma determinada observação.
- A regressão, assim como a classificação tem o objetivo de retornar um modelo ou função que consiga distinguir os dados, porém, em contraponto com a classificação, a regressão não retorna categorias, mas sim valores reais e contínuos. Um exemplo seria aplicar a regressão para retornar uma função que mapeie a probabilidade de um cliente comprar produtos específicos, com base em suas compras cotidianas.

Dentre os algoritmos mais conhecidos no aprendizado supervisionado encontram-se:

- Regressão linear;
- Árvores de decisão;
- *Support Vector Machine* (SVM);
- *K-Nearest Neighbors* (KNN);
- Random forest;
- Rede neural artificial (RNA).

#### 2) Aprendizado não supervisionado

O aprendizado não-supervisionado tem como objetivo aprender os padrões nos dados não categorizados, afim de agrupá-los ou organizá-los em similaridades entre si. Assim pode-se segmentar uma grande quantidade de dados não categóricos e extrair informações importantes deles.

Existem diversos métodos para esse tipo de aprendizado, sendo o mais famoso deles a clusterização. Baxter et al. [11] diz que o objetivo da clusterização é de separar um conjunto de objetos em grupos, para que os membros de um grupo se diferenciem o menos possível de outro grupo, de acordo com o critério estabelecido.

### C. Redes Neurais Artificiais

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos. São definidos por um conjunto de unidades de processamento, também chamados de neurônios artificiais ou nós, interligados por uma gama de interconexões ou sinapses. Essas redes aprendem por meio da alteração de pesos entre suas conexões.

A arquitetura básica de uma RNA consiste em três tipos de camadas de neurônios: camadas de entrada, escondida e saída (figura 11).

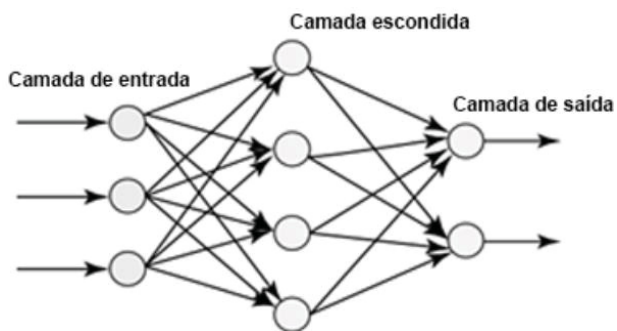


Fig. 11. Exemplo de arquitetura básica de uma Rede Neural Artificial. Fonte: Adaptado de Artificial Neural Networks [12].

Uma propriedade importante de uma rede neural artificial e o que a diferencia de outros métodos de aprendizado de máquina é o fato desta técnica fazer inferências sobre o que aprendeu. Com base nisso, algumas ligações entre neurônios são reforçadas enquanto outras são enfraquecidas, melhorando seu desempenho.

Dentre os recursos mais úteis em aplicações que utilizam a rede neural artificial estão a adaptação pela experiência de novos dados, a capacidade de aprendizado, a capacidade de generalização, a organização dos dados, a tolerância a falhas, o armazenamento distribuído e a facilidade para criar prototipagens [13].

### 1) Arquitetura de uma Rede Neural Artificial

A arquitetura básica de uma rede neural consiste em três camadas: as camadas de entrada, escondidas e de saída:

- A camada de entrada é responsável por receber as informações (dados), recursos e sinais de um ambiente externo. Esses dados recebidos (amostras) são geralmente normalizados pela função de ativação, para que a rede neural artificial tenha uma precisão maior nas operações realizadas em cima da amostragem;
- A camada escondida é composta por um ou mais neurônios que tem a responsabilidade de extrair informações (*features*) associadas aos dados sendo analisados. São eles quem processam a maior quantidade de dados em uma RNA;
- A camada de saída é composta por um ou mais neurônios responsáveis pela saída da informação de uma RNA. Essa camada processa a informação pelos neurônios nas camadas anteriores e retorna dados apresentáveis a partir da configuração de sua função de ativação.

Dependendo da aplicação da rede neural artificial, as camadas podem conter diferentes quantidades de neurônios. O neurônio, por sua vez, geralmente possui três estruturas lógicas: um conjunto de entradas com seus respectivos pesos, um somador dos sinais de entrada ponderados e a função de ativação.

É de responsabilidade do neurônio ligar as camadas através de pesos sinápticos. Assim, as entradas são propagadas pela rede neural, sendo transformadas pela sua função de ativação.

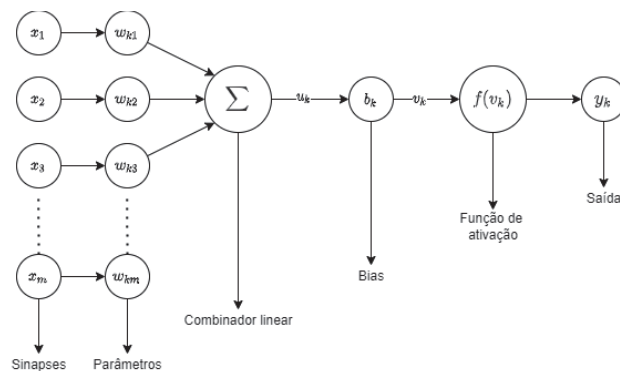


Fig. 12. Arquitetura de um neurônio artificial. Fonte: Elaborado pelo autor.

A figura 12 mostra a arquitetura de um neurônio artificial, sendo que pode ser segmentada em:

- Sinapse, que é o sinal de entrada do neurônio. Todo neurônio em uma rede neural artificial espera uma sinapse;
- Parâmetro, ou também chamado de peso, são os valores aos quais as sinapses serão multiplicadas na fase de aprendizado. Cada sinapse tem seu respectivo parâmetro. O parâmetro é representado como  $w_{kn}$ , onde  $k$  representa o neurônio e  $j$  o índice do parâmetro;
- Combinador linear, ou também chamado de somador, é onde os neurônios serão somados depois de serem multiplicados com seus respectivos pesos. Pode ser representado pela seguinte equação:

$$u_k = \sum_{j=1}^m x_{kj} w_{kj}$$

Onde  $u_k$  é a saída do somador, onde  $k$  representa o neurônio em si;

- O *bias* é um valor externo que é adicionado à saída do combinador linear. Serve para diminuir ou aumentar o valor de saída do combinador linear, com o intuito de deslocar a curva de aprendizado de uma rede neural artificial, pode ser representado pela seguinte equação:

$$v_k = b_k + u_k$$

Que é basicamente a soma entre o *bias* e a saída do combinador linear;

- Por último têm-se a função de ativação (capítulo D-2) que, dependendo da função, pode normalizar a saída de um neurônio de diferentes formas, como por exemplo, em intervalos de  $[0, 1]$  ou  $[-1, 1]$ .

A saída do neurônio é igual ao resultado da função de ativação aplicada à combinação das sinapses:

$$y_k = f(v_k)$$

Entre as arquiteturas de redes neurais artificiais mais consolidadas estão as redes neurais *feed-forward*, redes perceptrons multicamadas (MLP), redes neurais recorrentes, long short-term memory (LSTM) e a rede neural convolucional (CNN).

### a) Redes neurais feed-forward

A rede neural *feed-forward* (figura 11) permite que o sinal trafegue em um único sentido, da entrada para a saída [47]. São caracterizadas por 3 camadas principais: a camada de entrada, oculta e a de saída. Sendo assim, quando existe mais de uma camada oculta, a rede neural é chamada de profunda, assim surge o termo *Deep Learning*, ou aprendizado profundo.

Nesta arquitetura de rede neural não existe recorrência, ou seja, a saída de uma camada não afeta a si mesma. Dessa forma, redes neurais *feed-forward* tendem a ser diretas, associando as entradas com as saídas do modelo. Outra característica dessa arquitetura é que os neurônios podem estar parcialmente ou totalmente conectados.

### b) Redes neurais recorrentes

Diferente das redes neurais *feed-forward*, na arquitetura de rede neural recorrente é possível que o sinal trafegue nas duas direções por meio de *loops* ou recorrências na rede neural.

Assim, os neurônios das camadas escondidas de redes neurais recorrentes também podem receber como entrada o resultado da operação matemática que realizaram no período temporal anterior.

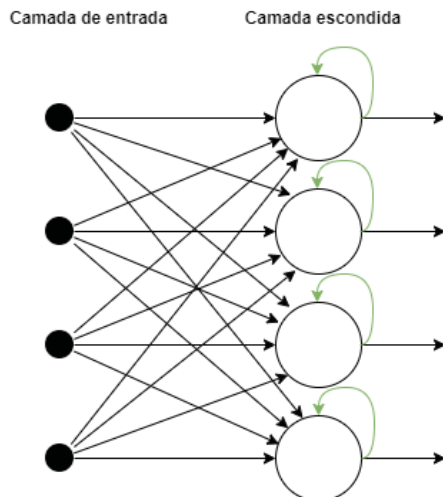


Fig. 13. Arquitetura básica de uma Rede Neural Recorrente. Fonte: Elaborado pelo autor.

Rede neural recorrente é uma arquitetura que atingiu o estado da arte de performance em modelos de linguagens, reconhecimento de fala e tradução de máquina [48].

## 2) Aprendizado e treinamento de uma Rede Neural Artificial

A principal propriedade de uma rede neural é sua habilidade de aprender a partir de dados de entrada (ou ambiente) e de sempre melhorar seu desempenho através da aprendizagem. Sendo assim, a RNA aprende sobre determinado assunto através de um processo de ajustes aplicados aos pesos sinápticos dos neurônios e seus respectivos *bias* (como explicado no capítulo C-1). Nessa perspectiva, Queguiner [30] explica que treinar uma rede neural artificial significa achar os pesos apropriados para as conexões neurais.

Todos os neurônios de uma camada específica geram uma saída, mas eles não têm o mesmo peso para a próxima camada.

A cada saída de neurônio e a cada camada em que os dados vão evoluindo, pesos são associados à entrada. Com isso, a rede neural consegue diferenciar entradas com pesos maiores e menores, assim descartando os pesos menores para a próxima camada. Assim pode ser criada uma forte associação entre os neurônios na conexão neural.

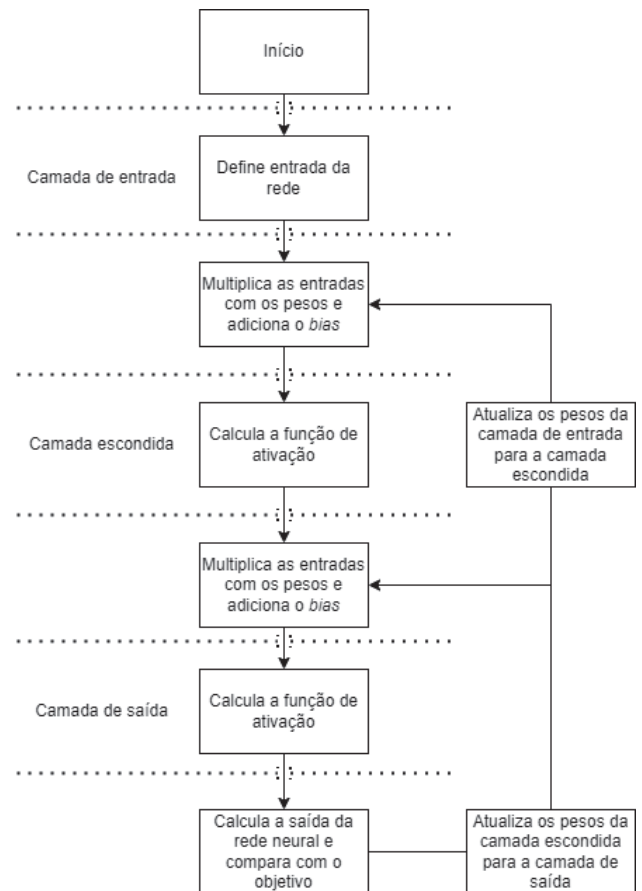


Fig. 14. Fluxograma do treinamento de uma rede neural artificial. Fonte: Elaborado pelo autor.

O aprendizado pode ocorrer quando uma rede neural atinge uma solução para determinado problema. Uma vez que todos os ajustes de pesos sinápticos têm o objetivo final de formar um vetor de saída que coincida com um valor desejado para cada vetor de entrada.

No final da etapa de treinamento, a RNA terá adquirido conhecimento sobre o ambiente, ou sobre os dados de entrada, que serão armazenados em seus parâmetros.

Além disso, a RNA pode ser treinada e validada utilizando o critério de erro mínimo que, segundo Ambrósio (2002), significa que os pesos das conexões entre os neurônios da rede neural artificial serão ajustados baseados nos erros encontrados nos sinais de saída, com o objetivo de minimizar este erro. O erro consiste na diferença entre o resultado calculado pelo neurônio e o resultado desejado. Para isso, é preciso fornecer à rede as respostas desejadas para cada padrão de entrada durante a fase de treinamento, que pode ser definido pela seguinte fórmula:

$$e_j = d_j(n) - y_j(n)$$

Onde o sinal de erro  $e$  do neurônio  $j$  no momento  $n$  é igual à saída desejada  $d$  para este neurônio, menos a sua saída obtida.

### 3) Backpropagation

*Backpropagation* ou retropropagação é o algoritmo de aprendizado que ajusta os pesos de uma rede neural artificial baseado na taxa de perda da iteração anterior. Ajustar os pesos corretamente acarreta em menores índices de erro, e com isso o modelo fica mais confiável aumentando sua generalização [32].

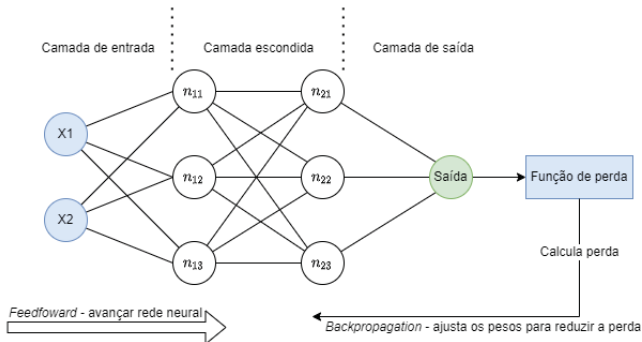


Fig. 15. Backpropagation em uma rede neural artificial. Fonte: Elaborado pelo autor.

Em resumo, a função da retropropagação é alimentar a perda para os neurônios anteriores, de maneira que seja possível fazer os ajustes finos nos pesos com base nesta perda. A função de otimização que gera o *backpropagation* ajuda a encontrar os pesos ideais e, com isso, uma perda menor para cada iteração.

### 4) Métodos de otimização

Os métodos de otimização são algoritmos usados para minimizar a função de perda e maximizar a eficiência do modelo de rede neural. Essas funções matemáticas são dependentes dos pesos e *bias* do modelo. É com base nelas que ocorre o *backpropagation* explicado na sessão anterior, ajustando os *bias* e pesos dos neurônios recorrentemente afim de diminuir a função de perda e otimizar o modelo de rede neural o máximo possível.

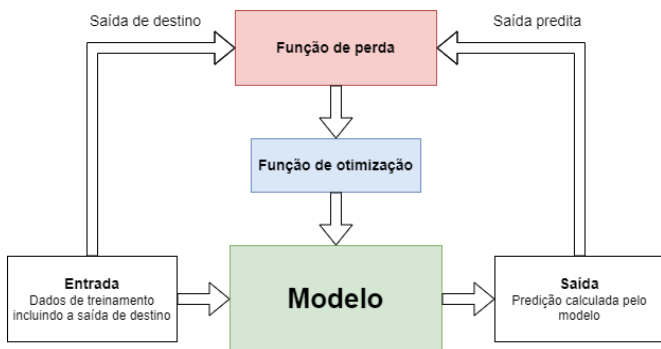


Fig. 16. Fluxograma da função de otimização em um modelo de rede neural artificial. Fonte: Elaborado pelo autor.

Existem diversos tipos de funções de otimização, a seguir são explicadas e exemplificadas as duas funções de otimização utilizadas para a elaboração deste artigo: gradiente descendente e, o *Adam Optimizer* (II-C-b).

### a) Gradiente Descendente

É o método de otimização mais comumente usada em redes neurais artificiais. Ela é baseada em uma função convexa e calcula como os pesos do modelo têm que ser alterados para que a perda chega à mínima [34].

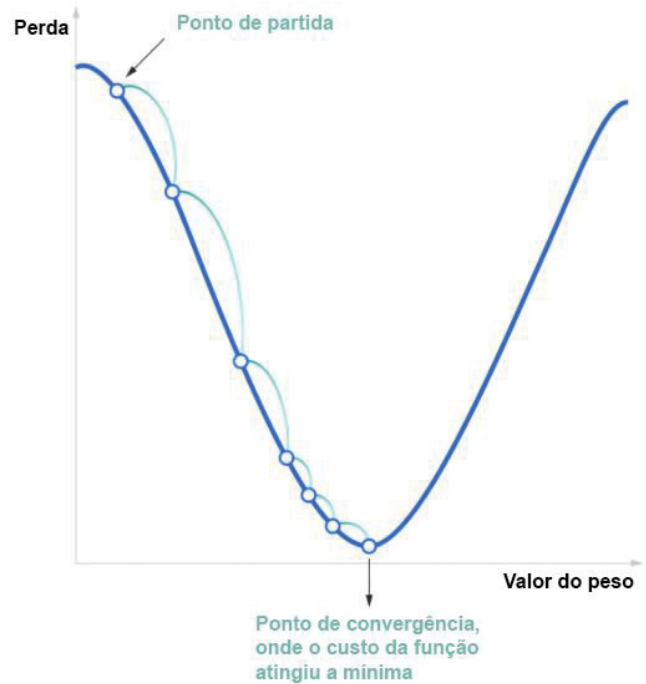


Fig. 17. Função convexa ao qual o gradiente descendente se baseia. Fonte: Adaptado de IBM [35].

Através da retropropagação, a perda é passada de uma camada para a outra e os parâmetros de peso do modelo são modificados de acordo com essas perdas, para que o erro seja minimizado.

$$W_{novo} = W_{velho} - \alpha * \Delta f(W_{velho})$$

A fórmula acima descreve o que um gradiente descendente faz de forma resumida: o  $W_{novo}$  representa a próxima posição do neurônio,  $W_{velho}$  representa a posição atual. O sinal de subtração faz referência para a parte de minimização do gradiente descendente. O  $\alpha$  é um fator de espera e a função de gradiente  $\Delta f(W_{velho})$  é a direção da descendência mais profunda. Sendo que a ideia dessa fórmula é achar a próxima posição ao qual o peso tem que ir, que é a posição da descendência mais profunda.

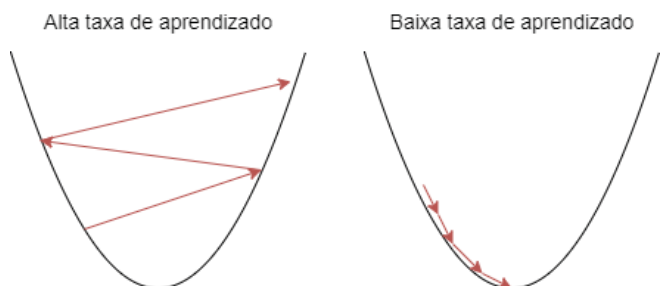


Fig. 18. Alta taxa de aprendizado e baixa taxa de aprendizado na função de otimização. Fonte: Elaborado pelo autor.

Para isso, é preciso configurar a taxa de aprendizado do gradiente descendente (figura 18), onde uma taxa de aprendizado muito grande pode passar da mínima da função,

enquanto uma taxa de aprendizado pequena demais eventualmente o gradiente descendente chegará na mínima, porém a quantidade de processamento será muito maior e por isso levará mais tempo para processar.

Dentre suas vantagens estão:

- Processamento leve;
- Fácil de implementar;
- Fácil de entender.

Dentre suas desvantagens estão:

- Pode travar na mínima local e não necessariamente a mínima global de toda a função;
- Se o *dataset* é muito grande, pode ser que leve um bom tempo para computar, já que os pesos são alterados após calcular o gradiente de todo o *dataset*.
- Utiliza muita memória para calcular o gradiente de todo o *dataset*.

#### b) Adam

O *Adaptive Moment Estimation*, traduzindo do inglês Estimativa de Momento Adaptativo, mais conhecido como Adam é um método de otimização que computa taxas de aprendizado adaptáveis para cada parâmetro [36]. Além disso, ele armazena uma média exponencialmente decrescente de gradientes quadrados anteriores.

$$m_t = \beta_1 m_t - 1 + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_t - 1 + (1 - \beta_2) g_t^2$$

Na fórmula acima, os hiper-parâmetros  $\beta_1$  e  $\beta_2$  controlam a taxa de decaimento exponencial das médias móveis. Assim,  $m_t$  e  $v_t$  são estimativas do primeiro momento (a média) e o segundo momento (a variância) dos gradientes [33].

Dentre suas vantagens estão:

- É bem rápido e converge rapidamente;
- Alta variação;

Dentre suas desvantagens estão:

- Custo computacional.

#### D. Redes Neurais Convolucionais (CNN)

Como explicado anteriormente, Redes Neurais Artificiais recebem dados na camada de entrada, processam esses dados na camada escondida e retornam o resultado final do processamento dos dados na camada de saída. Mas e se os dados fornecidos para a rede neurais fossem imagens? Para isto existe a Rede Neural Convolutiva (CNN).

As CNNs, como qualquer outra rede neural artificial, têm uma estrutura dividida em camadas, mas entre essas camadas estão as chamadas camadas de convolução. Porém, para entender a camada de convolução, e como ela processa as imagens, é preciso entender como a imagem se comporta como um objeto de entrada, ou um dado fornecido à rede neural.

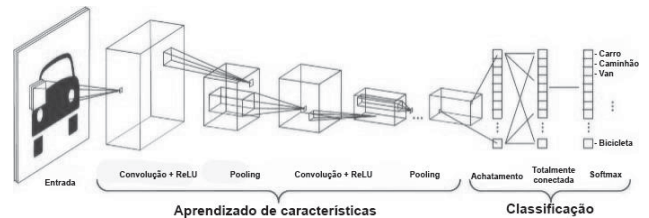


Fig. 19. Etapas de uma Rede Neural Convolutiva. Fonte: Adaptado Medium [14].

A imagem é um tipo especial de dado, já que, ao serem inseridas em uma rede neural elas são interpretadas como um objeto de *array* de valores. Esses objetos podem ser bidimensionais se a imagem for em preto e branco ou tridimensionais caso a imagem seja colorida. Por exemplo, dependendo do tamanho e resolução da imagem, o computador poderia interpretá-la como um *array* de números de tamanho 64x64x3 (altura, largura e as cores RGB da imagem). Cada um desses números contém números que vão de 0 a 255, que representam a intensidade do pixel naquele ponto da imagem.

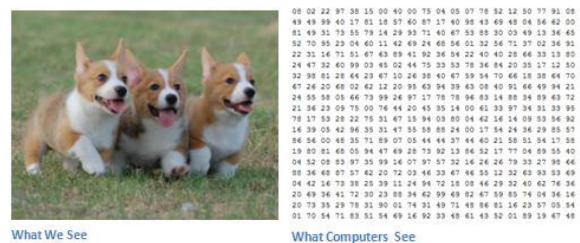


Fig. 20. O que o ser-humano enxerga (esquerda) e o que o computador enxerga (direita). Fonte: DESHPANDE [15].

Ao levar isso em consideração, as camadas de convolução possuem filtros (ou *kernels*) responsáveis por extrair informações da imagem, classificando-as no decorrer das camadas de convolução. Essas operações são realizadas diversas vezes para interpretar a imagem com o objetivo de categorizá-la na camada de saída.

Ao final de cada camada de convolução, tipicamente temos uma camada não-linear ou camada de ativação, que tem o propósito de diminuir a dimensão da saída das camadas de convolução: a *pooling*.

Sendo assim, podem-se separar os passos de uma CNN em três etapas para melhor entendimento: etapa de convolução, funções de ativação, *pooling* e camada totalmente conectada.

##### 1) Camadas de convolução

A camada de convolução é a etapa após a camada de entrada da imagem. Ela é uma fórmula matemática em duas funções para produzir uma terceira função que expressa como o formato da primeira é modificado pela segunda [16].

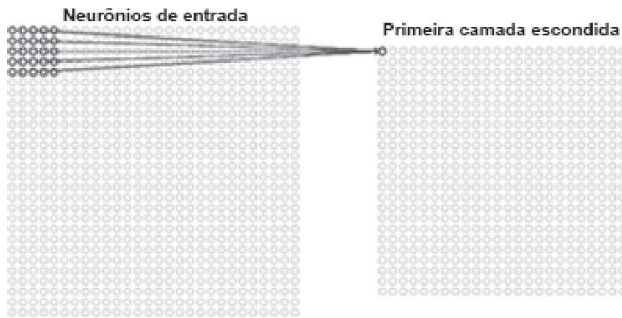


Fig. 21. Visualização de um filtro 5x5 fazendo a convolução em uma imagem de input e produzindo um mapa de ativação. Fonte: DESHPANDE [15].

A figura 21 mostra o processo onde o filtro seleciona os pixels da imagem, faz a convolução afim de gerar o produto sobre elas, passando-a para a próxima camada escondida.

Do mesmo modo, ao realizar uma convolução, a matriz de entrada da imagem diminui em tamanho, dependendo do *padding*. Nas figuras 22 e 23 pode-se ver um exemplo disso, onde é feita a convolução de uma imagem de matriz 6x6 em preto e branco, com um filtro (*kernel*) de tamanho 3x3. No final será produzido uma matriz de tamanho 4x4 pois, como explicado anteriormente e descrito na figura 22, o filtro 3x3 passa por toda a imagem de input da esquerda para direita, de cima para baixo, e cada produto da convolução é salvo na matriz resultante 4x4 ou o que chama-se de mapa de ativação.

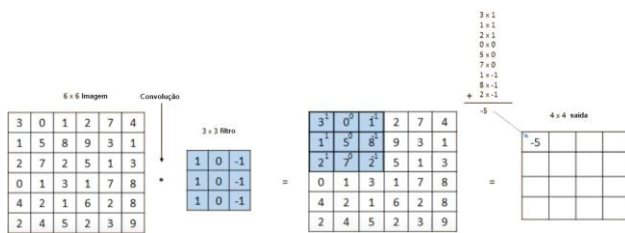


Fig. 22. Imagem de tamanho 6x6 e um filtro de tamanho 3x3. Fonte: Adaptado de RIZWAN [16].

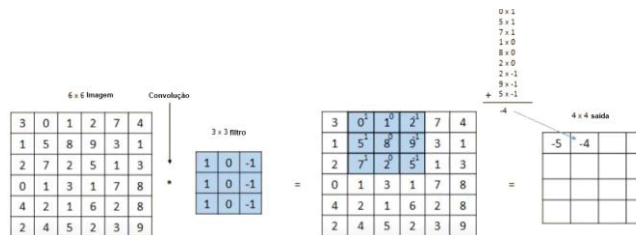


Fig. 23. Segunda convolução na imagem de tamanho 6x6 com filtro de tamanho 3x3. Fonte: Adaptado de RIZWAN [16].

Um dos problemas de continuar com as convoluções é a perda de dados, uma vez que a imagem diminui de tamanho a cada camada de convolução.

Para isso temos o chamado *padding*, que adiciona zeros na borda da matriz da imagem, e serve para controlarmos o tamanho espacial da imagem de saída ou mapa de ativação.

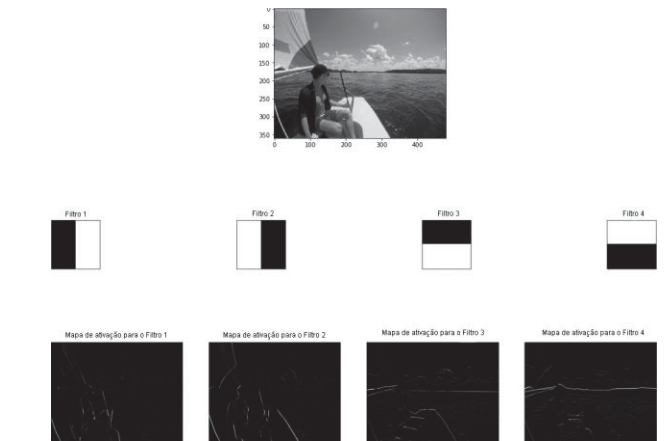


Fig. 24. Exemplo de mapas de ativação utilizando 4 filtros diferentes. Fonte: Adaptado de JORDAN [17].

Neste sentido, quando ocorre o processo de convolução em toda a matriz da imagem, é gerado uma nova matriz da imagem, também chamada de mapa de características ou *feature map*. Este mapa de ativação é gerado para cada filtro usado na camada de convolução e representa as características encontradas na imagem. Portanto, de modo geral, quanto mais filtros, maior quantidade de informação pode-se extrair de uma imagem [16].

É importante salientar que o tamanho do filtro de uma imagem colorida tem que ter a mesma profundidade da imagem. Então uma imagem 64x64x3 não poderia ter um filtro 3x3x2, mas sim um filtro 3x3x3, uma vez que a dimensionalidade de uma imagem colorida geralmente representa as cores RGB (vermelho, verde e azul), assim como uma imagem em preto e branco geralmente é representada por uma matriz de duas dimensões, somente com a altura e largura da imagem. Da mesma forma, não é necessário informar exatamente os filtros que a CNN irá utilizar, para isto é feita a parametrização dos filtros utilizando hiperparâmetros e é papel da rede neural aprender quais são os melhores filtros durante o treinamento [17]. Apesar de ainda ser necessário informar a quantidade de filtros que a rede neural irá aplicar.

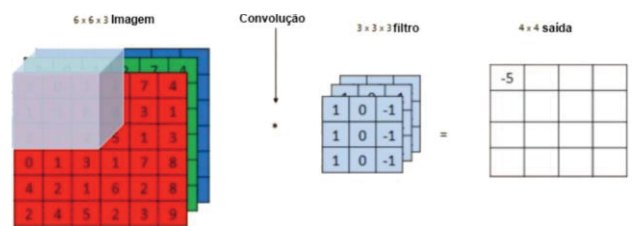


Fig. 25. Convolução em uma imagem colorida de tamanho 6x6x3 e filtro de tamanho 3x3x3. Fonte: Adaptado de RIZWAN [16].

Vale destacar que, a partir da segunda camada de convolução da CNN, as entradas não são mais a imagem original, mas sim os mapas de ativação da camada anterior. Para isso, o filtro passa a ter profundidade igual ao número de mapas de ativação gerados na camada anterior.

A medida em que as camadas de convolução vão evoluindo e com cada uma utilizando o mapa de ativação da camada anterior, fica cada vez mais difícil do ser humano conseguir interpretar que aquela matriz é uma imagem. Isso porque a cada camada, a representação da imagem fica mais abstrata.

## 2) Funções de ativação

A função de ativação é responsável por determinar a frequência de saída dos neurônios artificiais, podendo ser tanto linear quanto não linear. Para funções de ativação lineares, a camada escondida mapeia a função de entrada para a saída, antes da predição final do modelo. A saída de uma rede neural artificial é, por natureza linear. Portanto, funções de ativação não lineares são necessárias para converter as entradas lineares em saídas não lineares [52].

Assim, em modelos de classificação não lineares como a classificação de imagens, é papel da função de ativação transformar a rede neural artificial em um modelo não-linear. Essa não linearidade é necessária para classificar as imagens fornecidas na entrada em mais de uma categoria.

Além disso, utilizar uma camada de ativação após cada camada de convolução, faz com que o modelo consiga criar associações mais complexas entre as entradas e saídas da rede neural, sendo essencial para aprendizados complexos como a interpretação e segmentação de imagens.

Em resumo, a função de ativação decide se o neurônio deve ou não ser ativado, ou seja, se a informação que o neurônio está passando é relevante ou não para a saída desejada. Pode ser definida genericamente pela fórmula:

$$y = \text{Ativação}(\sum(\text{peso} * \text{entrada}) + \text{bias})$$

Sem a função de ativação, os pesos e os bias de um neurônio dariam uma saída linear – pode-se resumir em sendo um modelo de regressão linear –, incapaz de resolver problemas complexos de múltiplas categorias.

Dentre as funções de ativação mais conhecidas para funções não-lineares, têm-se:

### a) ReLU

ReLU, do inglês *Rectified Linear Units* é a função de ativação mais utilizada em redes neurais artificiais, e especialmente em redes neurais convolucionais.

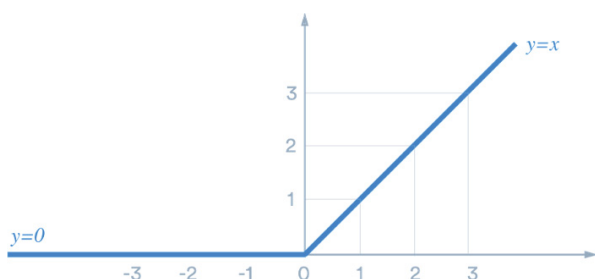


Fig. 26. Função de ativação ReLU. Fonte: CLAPPIS [18].

Assim como Liu (2017) comenta em seu artigo: ela se comporta de forma linear para valores positivos e zero para valores negativos (figura 26). Assim, ela possui algumas vantagens:

- Por não ter uma matemática complexa, é fácil de computar, assim o modelo leva menos tempo para executar;
- Possui uma conversão mais rápida devido sua linearidade, o que significa que quando o valor 'x' começa a ficar muito grande, ele não "satura" seu valor ou entra em um platô;

- Como para valores negativos a função de ativação ReLU substitui por zero, isso significa que ela não é ativada em todas as saídas.

A função de ativação ReLU pode ser expressada pela seguinte fórmula:

$$y = \max(0, x)$$

### b) Sigmóide

De acordo com Sharma (2017), a especialidade da função de ativação Sigmóide é a de ser usada em modelos onde é preciso prever uma probabilidade como saída, já que probabilidade é algo que existe em um intervalo de 0 a 1. Ela é usada comumente em modelos com duas classes de saída.

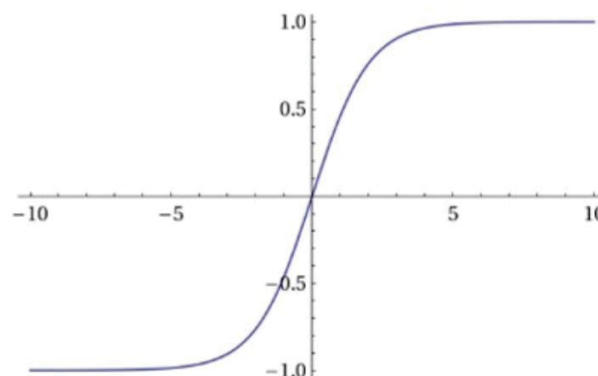


Fig. 27. Função de ativação Sigmóide. Fonte: Adaptado de BOLTON [39].

Como a figura 27 mostra, a função de ativação sigmoide tem como saída valores entre 0 e 1. Por isso ela é uma função diferenciável, ou seja, é possível achar a inclinação da curva em dois pontos distintos.

Essa função de ativação pode ser representada pela seguinte fórmula:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

### c) Softmax

Softmax é uma função de ativação que converte um vetor de números em um vetor de probabilidades, onde a probabilidade de cada valor é proporcional à escala relativa de cada valor no vetor [40]. Ela poderia ser representada por um desenho idêntico ao da função de ativação Sigmóide (figura 27), porém para saídas de múltiplas classes.

Cada saída da função Softmax é interpretada como a probabilidade para cada classe, ao converter a soma dos pesos das classes em probabilidade. E, diferentemente da função de ativação Sigmóide, a função de ativação Softmax é usada em modelos cuja saída contém diversas classes.

Essa função de ativação pode ser representada pela seguinte fórmula:

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)}$$

## 3) Camadas de pooling

A camada de *pooling* condensa o tamanho espacial da imagem, ao fazer uma redução das dimensões e do número de parâmetros da imagem. O processo de *pooling* reduz a dimensionalidade de cada mapa de ativação da imagem, mas retém as informações importantes [14].

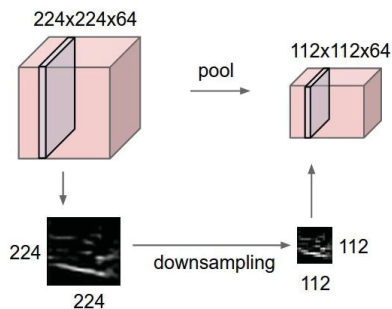


Fig. 28. Processo de *downsampling* na camada de *pooling*. Fonte: CLAPPIS [18].

Essa camada é similar à camada de convolução pelo fato de também definições de tamanho do filtro, porém seu propósito não é extrair as *features* da imagem, e sim condensar informações e diminuir o tamanho dos dados de entrada. Em resumo, a camada de *pooling* tem o papel de simplificar a saída após uma convolução na imagem [20].

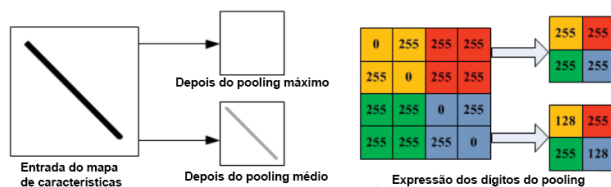


Fig. 29. Diferenças entre o *max pooling* e o *average pooling*. Fonte: Adaptado de Y et al. [21].

Existem dois tipos principais de *pooling*:

- *Max pooling*, onde é retornado o maior valor dentro da matriz de entrada para cada região percorrida pela janela do filtro. Ele é o mais comum dos tipos de *pooling*;
- *Average pooling*, que retorna a média dos valores de dentro da matriz de entrada, para cada região percorrida pela janela do filtro.

#### 4) Camada totalmente conectada

A camada totalmente conectada tem o objetivo de retornar a classificação da imagem como saída de uma CNN. Isso é possível pois ela recebe como entrada a saída da camada anterior que foi transformada em um vetor, e, com base nisso, determina quais características mais se correlacionam com os mapas de ativação [16].

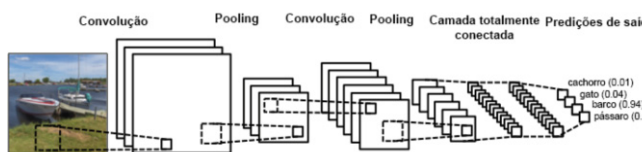


Fig. 30. Camada totalmente conectada classificando a imagem de input nas categorias de cachorro, gato, barco e pássaro. Fonte: Adaptado de JORDAN [17].

Antes de passar os dados para a camada totalmente conectada, é preciso fazer o achatamento, ou *flattening*, da camada anterior. Isso porque até agora os mapas de ativação eram somente representados por matrizes N-dimensionais. Essa matriz deve ser transformada em um vetor unidimensional para servir de entrada para a camada totalmente conectada.

Assim, ela converte a matriz gerada pelas camadas de convolução e pooling (que por sua vez foram achatadas), em um vetor de classificação, com as categorias especificadas no treinamento.

#### E. Métricas de avaliação

Para avaliar o desempenho dos modelos de Rede Neural Convolutiva, podem ser utilizadas as métricas de acurácia do modelo, as curvas de perda, perda nos dados de validação e a matriz de confusão do modelo com base nas 9 classes (movimentos artísticos) utilizadas como *output* do modelo.

##### 1) Matriz de confusão

A matriz de confusão é tipicamente usada em aprendizado de máquina para avaliar ou visualizar o comportamento de modelos de classificação supervisionada [50]. É uma matriz  $n \times n$  onde as linhas representam a classe atual das instâncias e as colunas representam as classes previstas.

Por exemplo, ao fazer uma classificação binária, então a matriz de confusão será uma matriz 2x2 que corresponde ao número de *verdadeiros positivos* (VP), *verdadeiros negativos* (VN), *falsos positivos* (FP) e *falsos negativos* (FN), como a seguir:

$$\begin{bmatrix} VP & FN \\ FP & VN \end{bmatrix}$$

- Verdadeiro positivo: ocorre quando o modelo previu a classe positiva corretamente;
- Falso positivo: ocorre quando o modelo previu a classe incorretamente, ao fornecer um valor de saída positivo, porém o resultado real foi negativo;
- Verdadeiro negativo: ocorre quando o modelo previu a classe negativa corretamente;
- Falso negativo: ocorre quando o modelo previu a classe incorretamente, ao fornecer um valor de saída negativo, porém o resultado real foi positivo.

De acordo com Caelen (2019), a matriz de confusão contém toda a informação bruta sobre as predições feitas por um modelo de classificação em uma determinada base de dados. E que, para avaliar a acurácia geral do modelo, é comum se utilizar de uma base de testes separada da base de treinamento do modelo.

Muitos indicadores de performance podem ser extraídos da matriz de confusão, como a acurácia, *precision*, *recall* e *F-score*.

##### 2) Acurácia

Essa métrica é geralmente determinada depois do treinamento do modelo ter sido finalizado. Ela funciona com base nos dados de teste, que são fornecidos ao modelo e o

número de erros e acertos que o modelo gera são registrados, por último uma porcentagem com a taxa de erros e acertos é computada, gerando a taxa de acurácia (do inglês *accuracy*).

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN}$$

### 3) Precision

*Precision* ou precisão é calculada como a razão entre a quantidade de verdadeiros positivos e a soma de falsos e verdadeiro positivos.

$$Precision = \frac{VP}{VP + FP}$$

Ela mede a acurácia do modelo em classificar os *samples* como positivo.

A precisão será alta quando:

- O modelo faz várias predições corretamente positivas (maximiza verdadeiro positivo);
- O modelo faz poucas predições incorretamente positivas (minimiza falsos positivos).

Em resumo, a precisão nos mostra o quão confiável o modelo é em classificar as amostras como positivos. É aconselhável utilizar o *precision* como métrica quando o problema é sensível em classificar amostras como positivo, mesmo se for falso positivo.

### 4) Recall

*Recall* ou revocação é calculado como a razão entre a quantidade de verdadeiros positivos e a soma de verdadeiros positivos e falsos negativos.

$$Recall = \frac{VP}{VP + FN}$$

Quando um modelo classifica todas as amostras como positivo, então o *recall* será 100%, mesmo se todas as amostras negativas estejam incorretamente classificadas como positivo. Isso porque o *recall* só se importa como as amostras positivas são classificadas.

É aconselhável utilizar o *recall* quando o objetivo do modelo é identificar todos os *samples* positivos, sem se importar com os falsos positivos.

### 5) F1-Score

*F1-Score* mede a média harmônica entre *Precision* e *Recall*. Sendo que quando o *F1-score* está baixo, é provável que a *precision* ou *recall* está com valor baixo.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Esse método retorna uma métrica melhor para classes desbalanceadas, como é o caso do estudo deste artigo, uma vez que cada pintura se diferencia muito umas das outras.

### 6) Perdas

Em resumo, quanto menor a perda (ou *loss*), melhor o modelo, com a exceção do *overfitting*. Isso porque a perda é o somatório dos erros cometidos por cada exemplo nos dados de validação e treinamento fornecidos ao modelo.

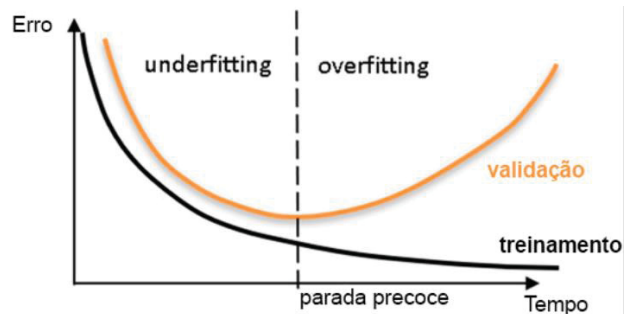


Fig. 31. Gráfico de perda nos dados de validação e treinamento utilizando a *cross-validation*. Fonte: Adaptado de ABRAMBES et al. (2018).

## F. Transfer Learning

*Transfer Learning* ou transferência de aprendizado é uma abordagem que visa a reutilização de modelos de *deep learning* já treinados para resolver novos problemas de forma mais rápida e efetiva, já que poupa recursos computacionais e tempo no treinamento de redes neurais.

Para isto, é preciso armazenar as camadas de uma rede neural já treinada. Assim elas podem ser reutilizadas por outro modelo que, por sua vez, precisa adicionar novas camadas em cima da rede neural já treinada para resolver o problema que a solução aborda.

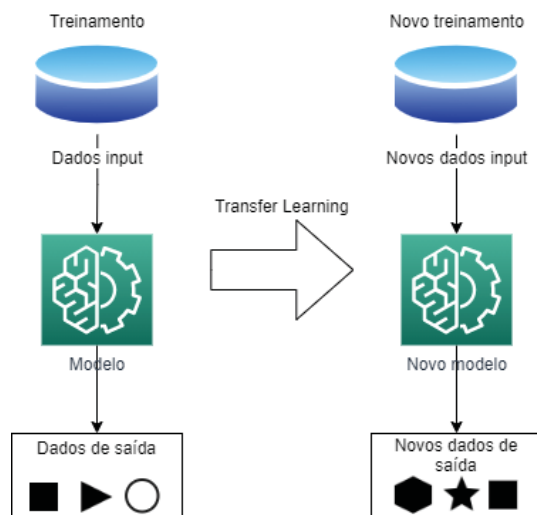


Fig. 32. Ilustração do processo de aprendizado por transferência. Fonte: Elaborado pelo autor.

Como ilustrado na figura 32, o lado esquerdo representa um treinamento tradicional de *machine learning*. Por outro lado, à direita têm-se um modelo que se utiliza dos dados já treinados do modelo da esquerda, e em cima disso adiciona seus próprios dados de treinamento, assim como novas camadas e saída desejada.

Esse processo é necessário para resolver um dos maiores problemas da aprendizagem profunda: a dependência de dados. De acordo com Sculley et al. (2015), a dependência de dados pode ser dividida em:

- Dependência de dados instáveis: significa que alguns dados de entrada qualitativamente ou quantitativamente podem mudar seu comportamento com o decorrer do tempo. Isso pode acontecer quando os sinais de entrada vêm implicitamente, como por

exemplo, de outro modelo de aprendizado de máquina [53], como também pode acontecer explicitamente, quando o dono dos dados de origem é separado do modelo que os consome;

- Dependência de dados subutilizados: são dados na maioria das vezes desnecessários, ou melhor, são sinais de entrada que não fornecem um benefício ao modelo de aprendizado de máquina. Dependência de dados subutilizados podem se infiltrar no modelo de várias maneiras, como por características (*features*) ultrapassadas ou redundantes, características agrupadas e a correlação de características onde a menos impactante pode ser utilizada pelo modelo, mesmo não sendo a ideal [53].

Se comparado aos modelos tradicionais de aprendizado de máquina, as redes neurais têm uma dependência muito mais forte com os dados de treinamento, já que geralmente, uma rede neural precisa de uma grande quantidade de dados para poder entender os possíveis padrões entre os dados [41]. Muitos modelos que utilizam rede neural sofrem de falta de dados de treinamento, devido à essa forte dependência com a quantidade de insumos.

Em resumo, o aprendizado por transferência resolve o problema de insuficiência da quantidade de dados de treinamento. Já que, o conhecimento adquirido de treinamentos anteriores é transferido do modelo de origem para o modelo atual, uma vez que assume que os dados de treinamento e teste não precisam ser identicamente distribuídos. De acordo com Tan, et al. isso acarreta em efeitos extremamente positivos em muitos modelos difíceis de melhorar, devido a insuficiência de dados ou poder de processamento.

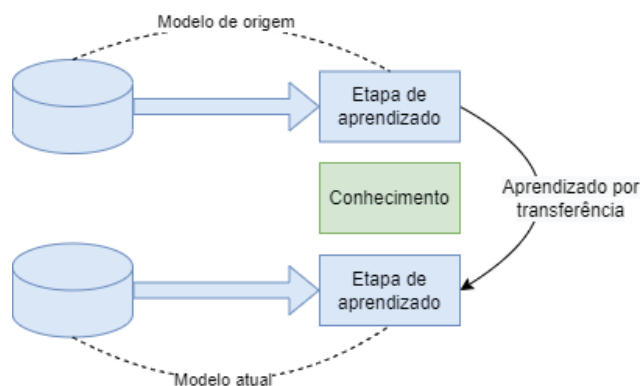


Fig. 33. Resumo do propósito do aprendizado por transferência. Fonte: Elaborado pelo autor.

De acordo com Pan et al. (2010), com base no conceito de transferência de aprendizado, e seu uso em aprendizado de máquina tradicional, é possível categorizá-la em três subgrupos: transferência de aprendizado indutivo, transferência de aprendizado transdutivo e transferência de aprendizado não supervisionado [46].

Na transferência de aprendizado indutivo, a tarefa atual é diferente da tarefa de origem. Neste caso, alguns dados categorizados no domínio atual são requeridos para induzir um objetivo preditivo no modelo.

Diferente da transferência indutiva, a transferência transdutiva explora o conjunto categorizado de treinamento e um conjunto não categorizado de teste para treinar o modelo,

afim de inferir as categorias no conjunto de teste não categorizado [51].

Ainda de acordo com Rezaei *et al.*, para uma nova amostra, o algoritmo de transferência de aprendizado transdutivo usa todos os seus dados para treinar o modelo, incluindo os dados da nova amostra. Como por exemplo, em classificação de sequência biológica, as próximas amostras não categorizadas com diferentes distribuições de características precisa ser categorizada de acordo com os experimentos anteriores.

Similar à transferência de aprendizado indutiva, a transferência de aprendizado não supervisionada a tarefa atual é diferente, porém é correlacionada com a tarefa de origem. Contudo, como o nome já diz, sua principal aplicação está em resolver tarefas de aprendizado não supervisionado, como clusterização e redução de dimensionalidade [46].

### G. Data Augmentation

De acordo com Shorten *et al.* [43], Redes neurais convolucionais têm uma performance alta quando se trata de tarefas de visão computacional. Contudo as CNNs são altamente dependentes de grande quantidade de dados (como explicado no capítulo II-E), para evitar o *overfitting* – fenômeno que ocorre quando a rede neural aprende uma função com uma alta variância, como um modelo perfeitamente treinado somente para os dados de treinamento.

O conceito de *Data Augmentation* soluciona o problema de limitação dos dados para o treinamento de redes neurais, uma vez que utiliza técnicas para aumentar a qualidade e quantidade dos dados de treinamento. Essas técnicas incluem o conjunto de algoritmos que constroem dados sintéticos a partir dos dados disponíveis [44].

Muitas estratégias para aumentar a performance generalizada de modelos de aprendizado de máquina profunda focam na arquitetura do modelo em si. Com isso surgiram arquiteturas progressivamente mais complexas, como *AlexNet* e *Resnet*. Assim como outras soluções como o aprendizado por transferência ou a regularização por *drouput* surgiram para estender o uso de redes neurais para modelos com uma pequena quantidade de dados.

Em contraste com as técnicas mencionadas acima, o *Data Augmentation* aborda o problema do *overfitting* em sua causa raiz: o *dataset* de treinamento do modelo. Para isso, assume-se que quanto mais informação puder ser extraída da base de dados por meio do aumento programático dos dados, seja por transformações, filtros ou distorções, melhor será para o modelo.

É importante salientar sobre a segurança nas transformações das imagens, que dependem do domínio do modelo e qual o objetivo de sua predição. Por exemplo, no caso do filtro de *flipping* (II-F1), que pode ser aplicado para o enriquecimento de bases de dados para utilizados para o reconhecimento de cachorros e gatos, não poderia ser aplicado de forma segura para um modelo de reconhecimento de números, onde a orientação dos números 6 e 9 são extremamente importantes para sua interpretação.

A seguir, algumas técnicas de *data augmentation* que podem ser usadas para enriquecer a base de dados de imagens para o treinamento do modelo.

### 1) Flipping

É um dos acréscimos a serem feitos nas bases de dados mais fáceis de implementar. Nele a imagem é rotacionada em seu eixo, sendo que a rotação no eixo horizontal é mais comum que a rotação no eixo vertical. Não é seguro dependendo da aplicação do modelo.

### 2) Cor no espaço

Geralmente as imagens são codificadas como tensores de dimensão (altura x largura x canais de cor). Fazer transformações nos espaços dos canais de cor é uma estratégia de melhora dos dados bem prática. A mais básica das melhoras no espaço de cor seria isolar um único canal de cor da imagem, como o vermelho, verde ou azul (*red, green, blue, RGB*).

### 3) Corte

O corte pode ser usado para imagens com diferentes dimensões, cortando uma parte central e principal de cada imagem. Assim como a tradução (II-F5), pode ser usada cortes aleatórios na imagem, porém, diferentemente da tradução, o corte não mantém o tamanho original da imagem.

### 4) Rotacionar

Responsável por rotacionar a imagem para a esquerda ou direito no seu eixo, entre 1° e 359°. Assim como o *flipping*, a rotação não é segura dependendo da aplicação do modelo.

### 5) Tradução

Deslocar imagens para a esquerda, direita, cima ou baixo pode ser muito efetiva para evitar um *bias* de posição nos dados. Por exemplo, um modelo de reconhecimento facial onde todas as imagens estão centralizadas no centro do rosto da pessoa, o modelo ficaria totalmente dependente de uma imagem perfeitamente centralizada.

Quando a imagem é deslocada, os espaços restantes são preenchidos com 0s, 255s ou com ruídos Gaussianos.

### 6) Injeção de ruídos

Essa técnica para o aumento dos dados consiste em injetar uma matriz de valores randômicos na imagem, geralmente esses valores randômicos são pegos da distribuição Gaussiana.

### 7) Transformação de cor no espaço

Essa técnica consiste em transformar as cores na matriz de uma imagem. Com isso é possível fazer filtros como aumentar o contraste, equalização da imagem, cores negativas, efeitos de *sharpen, balance*, entre muitos outros.

### 8) Filtros de kernel

Os filtros de *kernel* são técnicas muito populares para aplicar efeitos nas imagens.

Esses filtros funcionam ao deslizar uma matriz  $n \times n$  entre toda a matriz da imagem com a matriz desejada para a saída desejada.

## III. MATERIAIS E MÉTODOS

Nesse artigo foram utilizados dois *datasets* contendo obras de arte de mais de 50 artistas referências em seus respectivos movimentos artísticos.

Nas sessões a seguir serão apresentados detalhadamente os *datasets* utilizados e separação dos dados. Assim como os motivos pelos quais o modelo não foi treinado utilizando todos os movimentos artísticos já estudados. Por fim, serão apresentados como o modelo foi treinado e a forma de avaliação empregada no estudo.

## A. Datasets utilizados

Foi utilizado como ponto de partida o *dataset* “Best Artwork of All Time” [22]. Esse repositório contém imagens de obras de arte de mais de 50 artistas de qualquer região do mundo e período na história. Além disso, ele conta com uma tabela que lista a quais movimentos artísticos determinado artista pertenceu, sua nacionalidade, período em que viveu e uma breve biografia sobre cada um.

TABELA I. QUANTIDADE DE PINTURAS POR ARTISTA NA BASE DE DADOS “BEST ARTWORKS OF ALL TIME ORDENADOS” PELO MOVIMENTO ARTÍSTICO

Nome do artista	Movimento artístico	Quantidade de pinturas
Caravaggio	Barroco	55
Rembrandt	Barroco	262
Diego Velazquez	Barroco	128
Peter Paul Rubens	Barroco	141
Pablo Picasso	Cubismo	439
Paul Klee	Expressionismo	188
Amedeo Modigliani	Expressionismo	193
Vasily Kandinsky	Expressionismo	88
Henri Matisse	Impressionismo	186
Edgar Degas	Impressionismo	702
Claude Monet	Impressionismo	73
Edouard Manet	Impressionismo	90
Alfred Sisley	Impressionismo	259
Pierre-Auguste Renoir	Impressionismo	336
Camille Pissarro	Impressionismo	91
El Greco	Maneirismo	87
Piet Mondrian	Neoplasticismo	84
Andy Warhol	Arte Pop	181
Georges Seurat	Pós-Impressionismo	43
Henri de Toulouse-Lautrec	Pós-Impressionismo	81
Paul Cezanne	Pós-Impressionismo	47
Vincent van Gogh	Pós-Impressionismo	877
Henri Rousseau	Primitivismo	70
Marc Chagall	Primitivismo	239
Gustave Courbet	Realismo	59
Jan van Eyck	Renascimento	81
Titian	Renascimento	255
Michelangelo	Renascimento	49
Leonardo da Vinci	Renascimento	143
Raphael	Renascimento	109
Sandro Botticelli	Renascimento	164
Giotto di Bondone	Renascimento	119
Pieter Bruegel	Renascimento	134
Albrecht Dürer	Renascimento	328
Hieronymus Bosch	Renascimento	137
Francisco Goya	Romantismo	291
Eugene Delacroix	Romantismo	31
William Turner	Romantismo	66
Kazimir Malevich	Suprematismo	126
Rene Magritte	Surrealismo	194
Salvador Dali	Surrealismo	139
Joan Miro	Surrealismo	102
Frida Kahlo	Surrealismo	120
Edvard Munch	Simbolismo	67
Mikhail Vrubel	Simbolismo	171
Paul Gauguin	Simbolismo	311
Gustav Klimt	Simbolismo	117

As imagens contidas neste conjunto de dados não estão totalmente estruturadas. Isso porque elas estão em diferentes tamanhos e não existe um padrão quanto à imagem ser colorida ou preto e branca. Um dos desafios desse trabalho foi estruturar essa base de dados para que seja possível otimizar ainda mais o modelo de CNN.

Outro desafio foi o fato de alguns artistas pertencerem à mais de um movimento artístico. Por isso foi preciso dividir as categorias ou fazer uma pesquisa mais aprofundada sobre o artista para determinar a qual movimento artístico ele mais contribuiu, e utilizar o que fizer mais sentido.

TABELA II. QUANTIDADE DE PINTURAS PELO MOVIMENTO ARTÍSTICO NA BASE DE DADOS “BEST ARTWORKS OF ALL TIME ORDENADOS”

Movimento artístico	Quantidade de pinturas
Barroco	586
Cubismo	439
Expressionismo	469
Impressionismo	1737
Mancirismo	87
Neoplasticismo	84
Arte Pop	181
Pós-Impressionismo	1048
Primitivismo	309
Realismo	59
Renascimento	1519
Romantismo	388
Suprematismo	126
Surrealismo	555
Simbolismo	666

Da mesma forma, foram utilizados somente movimentos artísticos a partir do Renascimento (século XIV - século XVII) até as vanguardas modernistas (século XIX - século XX). Isso porque, a partir da metade do século XX, a classificação entre movimentos artísticos começou a ficar cada vez mais complexa, até que nos dias atuais não se tem mais como enquadrar as obras em movimentos artísticos distintos.



Fig. 34. Afresco barroco da igreja de Santo Inácio, em Roma, por Andrea Pozzo. Fonte: Saber Cultural [29] (acesso em 2021).

Ainda mais, as obras utilizadas para o treinamento do modelo foram delimitadas somente na Europa (com algumas exceções descritas na sessão a seguir), já que os movimentos artísticos se desenvolveram de forma heterogênea e com características diferentes entre as diversas regiões do mundo.

Por exemplo, o Barroco, que teve seu início na Europa no final do século XVI e só chegou ao Brasil no final do século XVII. Do mesmo modo, o Barroco europeu (figura 34) ostentava muitos materiais caros devido ao apoio financeiro da Igreja Católica, como mármore e ouro. No Brasil, por outro lado, eram utilizados materiais rudimentares (figura

35), como madeira e tintas de menor qualidade. Em resumo, por mais que ambos pertençam ao mesmo movimento artístico, é difícil comparar uma obra com a outra, pois estão inseridas em contextos diferentes.



Fig. 35. Assunção da Virgem, pintura barroca no teto da igreja de São Francisco de Ouro Preto, por Mestre Ataíde. Fonte: Imagem de domínio público (acesso em 2021).

Durante as análises do *dataset*, foi observado que havia uma grande disparidade de quantidade de amostra entre os movimentos artísticos. Sendo que, por exemplo, o Renascimento tinha mais de 1750 imagens, o Cubismo tinha somente 500.



Fig. 36. Quantidade de imagens por movimento artístico no dataset Best Artworks of All Time. Fonte: Elaborado pelo autor.

Para normalizar a quantidade de amostras, foi também utilizado o *dataset* Pandora [23], que consiste em 7724 imagens de 12 movimentos artísticos: cerâmicas da Grécia antiga, iconoclastia, alto renascimento, barroco, rococó, romântico, impressionismo, realismo, cubismo, fauvismo, expressionismo abstrato e surrealismo. Como mostrado na tabela III.

TABELA III. ESTRUTURA DO BANCO DE DADOS PANDORA

Movimento artístico	Nº de imagens
Cerâmicas da Grécia antiga	350
Iconoclastia	665
Alto Renascimento	812
Barroco	960
Rococó	844
Romanticismo	874
Impressionismo	984
Realismo	307
Cubismo	920
Fauvismo	340
Expressionismo abstrato	426
Surrealismo	242

Com isso foi possível normalizar a base de dados e separar em 9 categorias (figura 37) –movimentos artísticos– a serem treinados, foram elas: expressionismo, impressionismo, surrealismo, pós-impressionismo, simbolismo, renascimento, cubismo, barroco e romantismo.

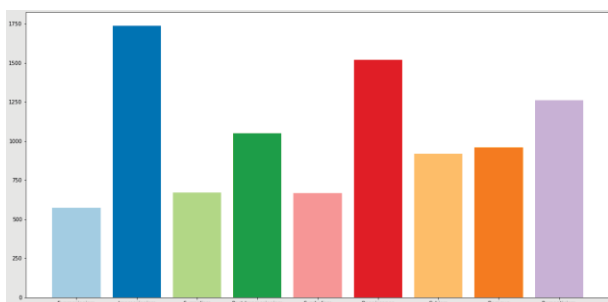


Fig. 37. Quantidade de imagens por movimento artístico no dataset Best Artworks of All Time em junção com a Pandora Database. Fonte: Elaborado pelo autor.

Além disso, foi utilizado o *Data Augmentation* para enriquecer as imagens do *dataset*, assim como aumentar a quantidade de imagens úteis para o treinamento do modelo, sendo que o *Data Augmentation* adiciona imagens em diferentes perspectivas e até filtros à base de dados de treino e validação. No caso deste artigo, foram adicionadas imagens rotacionadas no eixo horizontal, vertical e imagens espelhadas consigo mesmas, assim como mostra a figura 38.



Fig. 38. Pintura barroca após passar pelo filtro de *flipping* no *Data Augmentation*. Do lado esquerdo observa-se a imagem original, e do lado direito é a mesma imagem espelhada. Fonte: Elaborado pelo autor.

Em suma, com a junção dos *datasets* “Best Artworks of All Time” e “Pandora”, assim como o uso do *Data Augmentation*, foi atingido a marca de 9.357 imagens, divididas em 9 classes, como mostra a tabela IV.

TABELA IV. NÚMERO DE IMAGENS POR MOVIMENTO ARTÍSTICO NA BASE DE DADOS FINAL PARA O TREINAMENTO E VALIDAÇÃO DO MODELO

Movimento artístico	Nº de imagens
Barroco	960
Cubismo	920
Expressionismo	573
Impressionismo	1737
Pós-impressionismo	1048
Renascimento	1519
Romantismo	1262
Surrealismo	672
Simbolismo	666

### B. Data augmentation

Para enriquecer a base de dados, foi utilizado o conceito de *data augmentation* explicado no capítulo II-F. E para isso, foi utilizado a classe *ImageDataGenerator* [25] da biblioteca *Keras*, que possibilita aplicar diversas transformações nas imagens a serem treinadas, de forma aleatória.

### C. Treinamento dos modelos

Nessa etapa, foi utilizada algumas configurações de Redes Neurais Convolucionais com *Transfer Learning* para utilizar de parâmetros já treinados pela ResNet50 [26].

#### 1) ResNet50

É uma CNN com 50 camadas de profundidade e que foi treinada com mais de um milhão de imagens da base de dados ImageNet [27].

Esta Rede Neural Convolutiva pré-treinada, pode classificar imagens em mais de 1000 categorias, como caneta, lápis e vários animais. Por conta disso, a rede aprendeu várias representações em uma grande quantidade de imagens.

Ao treinar o modelo sem o uso dessa rede neural pré-treinada, não foi possível fazer com que o modelo aprenda sobre as características das diversas pinturas, uma vez que elas têm características muito distintas uma das outras.

#### 2) Hiperparâmetros

Para o *Data Augmentation*, explicado na sessão III-B, foram utilizados o *input shape* das imagens de (224, 224, 3), *batch size* de 16, filtros para virar horizontalmente e verticalmente as imagens, assim como e *shear range* (ou distorção das imagens em seu eixo) de 5. Tudo isso gerado aleatoriamente a cada *batch*.

Ao final da CNN ResNet50 (III-C1), foram adicionadas uma camada de achatamento, uma camada densa com 512 filtros, dropout de 0.5 e função de ativação ReLU, outra camada densa com 16 filtros e dropout de 0.5 para evitar o *overfitting* e função de ativação ReLU. Por último foi adicionado uma camada de *output* densa com 9 categorias (os 9 movimentos artísticos que o modelo quer prever) e função de ativação *softmax* (explicado na sessão II-D2). O número de épocas utilizadas no treinamento do modelo foi de 50, mas com o critério de parada explicado na sessão III-C4.

#### 3) Otimizador

Otimizadores são algoritmos usados para variar os atributos de redes neurais como a largura e índice de aprendizado do modelo, afim de reduzir as perdas.

Para este estudo foi utilizado o Adam Optimizer, que é um algoritmo de descida de gradiente que calcula as taxas de aprendizagem individuais de cada parâmetro e também o primeiro e segundo momento dos gradientes.

#### 4) Critério de parada

Grande parte do esforço desse estudo foi parar o treinamento do modelo no momento certo, uma vez que a base de dados utilizada no treinamento desta rede neural convolucional sofre com algumas limitações, como a quantidade de amostras e a falta de características marcantes entre cada movimento artístico. Por conta disso, o modelo pode facilmente ser demasiadamente treinado, assim ocorrendo o *overfitting*, que é quando o modelo aprende muito bem somente para os dados de treinamento, mas no momento de prever dados do mundo real o resultado não é satisfatório.

Sendo assim, o critério de parada tem a função de parar o treinamento do modelo quando a métrica monitorada para de melhorar. Assim é possível evitar o *overfitting* ao barrar o treinamento do modelo quando a perda ainda é baixa, ou a acurácia está em seu ponto mais alto.

Para o estudo, foi utilizado um critério de parada com base na perda para os dados de validação, ou *val loss*, utilizando uma paciência (ou número de épocas sem melhora no modelo) de 5.

Além disso foi utilizado a classes do Keras *ReduceLROnPlateau*, com o intuito de reduzir a taxa de aprendizado do modelo quando a métrica de *val loss* ter parado de aprender. Para esta, foram utilizadas as métricas de parada com base na *val loss*, paciência de 3 e fator (em quanto a taxa de aprendizagem será reduzida) de 0.1.

#### 5) Hardware

Para o treinamento do modelo foi utilizado uma máquina com as seguintes configurações:

- CPU: Intel Core i7-7700K, com 4.20GHz até 4.50GHz de velocidade de *clock*, 4 núcleos e 8 threads;
- Memória RAM: 16gb de memória com 2400MHz de frequência;
- Placa de vídeo: NVIDIA GeForce GTX 1060 GDDR5 com 6gb de memória de vídeo dedicada, 8008MHz de frequência de memória e 1506MHz de velocidade gráfica.

#### 6) Versões

Para o treinamento do modelo foi utilizado a linguagem de programação Python na versão 3.7.6. Como IDE foi utilizado o *Jupyter Notebook* e o Conda na versão 2.1.4 como gerenciador de pacotes e ambiente.

Para o uso da placa de vídeo nos treinamentos dos modelos, foi utilizado o CUDA, acrônimo de arquitetura de dispositivo unificado de computação, na versão 11.2. O CUDA é uma engine de computação criada pela NVIDIA para utilizar todos os recursos da placa de vídeo (GPU) para computação em paralelo, afim de aumentar drasticamente a performance nos treinamentos de modelo se comparado ao uso de uma CPU.

TABELA V. PRINCIPAIS BIBLIOTECAS UTILIZADAS PARA A ANÁLISE E TREINAMENTO DO MODELO DE REDE NEURAL CONVOLUCIONAL UTILIZADO NESTE ARTIGO

Nome da biblioteca	Versão
pandas	1.0.1
numpy	1.19.5
tensorflow	2.4.1
keras	2.4.3
scikit-learn	1.0
matplotlib	3.1.3
seaborn	0.11.2
tqdm	4.42.1

## IV. RESULTADOS E DISCUSSÕES

Neste capítulo são descritos os resultados obtidos ao usar os conceitos e metodologias descritos nas sessões anteriores. Vale ressaltar que todos os modelos treinados que obtiveram resultados expressivos utilizaram da rede neural convolucional ResNet50 como base, mudando somente a quantidade de amostras e os hiperparâmetros para cada treinamento.

A primeira rede neural convolucional treinada não utilizou a transferência de aprendizado da ResNet50, mas somente uma estrutura com 3 camadas de convolução, com 128, 64 e 32 filtros respectivamente, seguida por 3 camadas de achatamento, intercaladas com duas camadas densas com função de ativação ReLU e a camada de saída com a função de ativação Softmax. Esta mesma rede neural utilizou um critério de parada com paciência 5, a função de otimização Adam e como principal métrica a acurácia. O resultado não foi nada expressivo, com uma acurácia de 0.37 e F1-Score de 0.09, como mostra a tabela VI.

O próximo passo foi treinar uma a rede neural convolucional utilizando a transferência de aprendizado do modelo ResNet50 e a totalidade de amostras que a base descrita na sessão III-A contém. Comparado com o treinamento anterior o modelo se saiu melhor, porém, como era de se esperar pelo fato da base de dados não estar normalizada, o modelo teve um grande viés pendendo para pinturas impressionistas e renascentistas que continham mais amostras (figura 40).

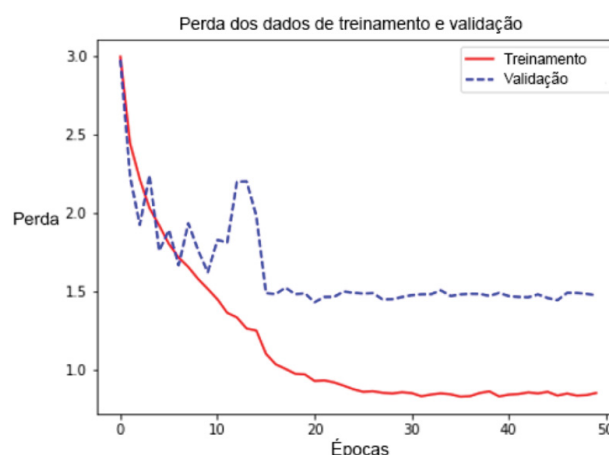


Fig. 39. Gráfico de perda para o primeiro modelo treinado, utilizando a database completa. Fonte: Elaborado pelo autor.

O resultado da figura 39 mostra que, para o conjunto dos dados fornecidos, o modelo se mostrou mais eficaz para os dados de treinamento, uma vez que a perda do treinamento ficou muito abaixo da perda dos dados de validação, o que

significa que para os dados que o modelo não foi treinado ele não se saiu tão bem. Além disso, enquanto o Renascimento ficou com uma acurácia de 87%, a escola artística do Cubismo não foi sequer relacionada entre as categorias preditas, como mostra a figura 40. Da mesma forma, na figura 40, as cores representam o mapa de calor do quão próximos de 100% de acurácia os movimentos artísticos estão sendo preditos.

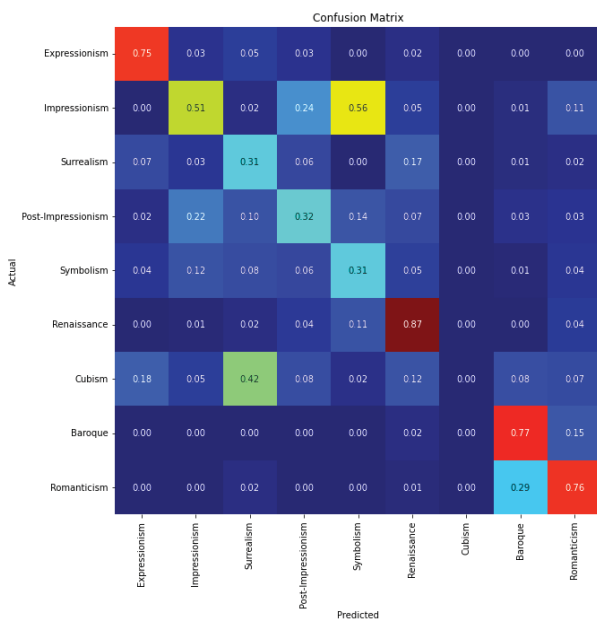


Fig. 40. Matriz de confusão do primeiro modelo treinado, utilizando a database completa. Fonte: Elaborado pelo autor.

Com base nisto foi possível concluir que é preciso normalizar a quantidade de amostras da base de dados (III-A), e partir da premissa de que todas as categorias têm quantidades iguais de imagens.

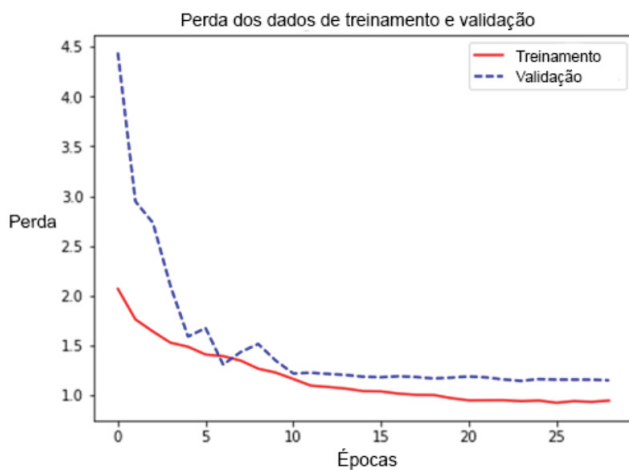


Fig. 41. Gráfico de perda para o modelo com melhor resultado, utilizando a database com quantidades de imagens iguais para cada movimento artístico. Fonte: Elaborado pelo autor.

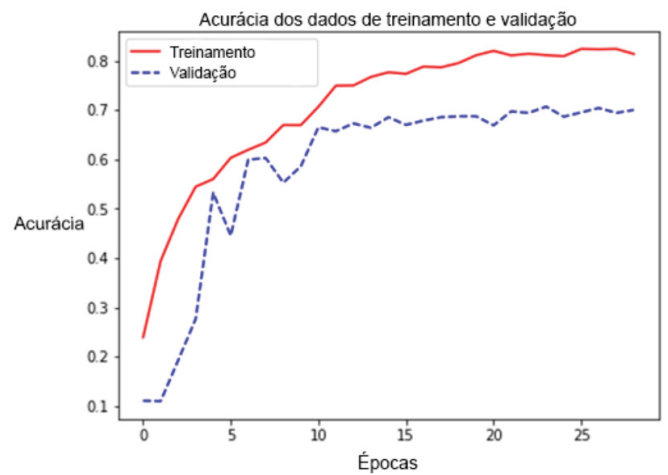


Fig. 42. Gráfico da acurácia para o modelo com melhor resultado, utilizando a database com quantidades de imagens iguais para cada movimento artístico. Fonte: Elaborado pelo autor.

A partir do balanceamento proporcional da quantidade de amostras para cada movimento artístico, foi possível treinar modelos eficazes, parametrizando seus hiperparâmetros, afim de atingir o melhor resultado possível. Para atingir este melhor resultado, foram treinados mais de 20 modelos de rede neural convolucional, sendo que na tabela VI estão descritos os modelos com melhores resultados, além do primeiro modelo treinado que, por mais que não tenha tido um resultado satisfatório, foi importante catalogar com o objetivo de mostrar o benefício da transferência de aprendizado.

Para chegar nos melhores hiperparâmetros, foram realizados testes começando com 5 camadas densas com 256, 128, 64, 32 e 16 filtros respectivamente, *dropout* de 0.5 e com função de ativação ReLU ao final de cada camada densa. A função de ativação de saída utilizada para todos os treinamentos foi a *Softmax*. Com isso, foi possível confirmar qual modelo se mostrou mais eficaz com uma menor quantidade de camadas densas, dessa forma foi possível afunilar qual a melhor quantidade de filtros em cada camada. Além disso, o critério de parada se mostrou mais eficaz com uma paciência menor, já que começou a parar o treinamento do modelo em menos épocas, poupando tempo e recursos.

O modelo com o melhor resultado, com os hiperparâmetros descritos no capítulo III-C, atingiu uma acurácia de 69% nos dados de validação e, a figura 41 mostra que a melhora na perda deste modelo foi significativa se comparado aos treinamentos anteriores. Além disso, pelo uso do critério de parada, foram necessários somente 26 épocas para atingir a melhor perda e acurácia no treinamento do modelo (figura 41), ao invés das 50 épocas utilizadas em experimentos anteriores, como apontado na figura 39.

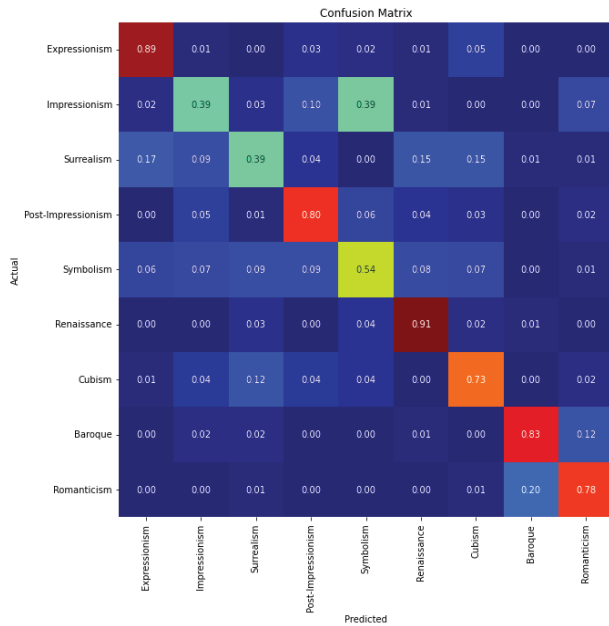


Fig. 43. Matriz de confusão do modelo com melhor resultado, utilizando a database com quantidades de imagens iguais para cada movimento artístico. Fonte: Elaborado pelo autor.

Além disso, percebe-se o aumento da acurácia na matriz de confusão da figura 43, já que desta vez não existe um viés para movimentos artísticos como Renascimento e Impressionismo (por conter maior quantidade de amostras na base de dados). E a escola artística do Cubismo conseguiu ser relacionada entre as categorias preditas pelo modelo.

	precision	recall	f1-score	support
Expressionism	0.78	0.89	0.83	114
Impressionism	0.58	0.39	0.46	114
Surrealism	0.57	0.39	0.47	114
Post-Impressionism	0.73	0.80	0.76	113
Symbolism	0.50	0.54	0.51	114
Renaissance	0.76	0.91	0.83	114
Cubism	0.69	0.73	0.71	114
Baroque	0.79	0.83	0.81	113
Romanticism	0.76	0.78	0.77	114
accuracy			0.69	1024
macro avg	0.68	0.69	0.68	1024
weighted avg	0.68	0.69	0.68	1024

Fig. 44. Precision, recall, f1-score e accuracy de cada movimento artístico categorizado pelo modelo. Fonte: Elaborado pelo autor.

O *F1-score* ficou acima da média para a maioria das categorias, porém ficou abaixo para os movimentos artísticos impressionismo e surrealismo. Faz sentido isto acontecer se for levada em consideração a diferença técnica entre pinturas da mesma escola artística, uma vez que essas três escolas têm diferenças consideráveis entre si mesmas.



Fig. 45. Categorização das pinturas do *dataset* de validação pelo modelo. Fonte: Elaborado pelo autor.

A tabela VI mostra os melhores modelos de rede neural artificial treinados e avaliados para este artigo, suas respectivas métricas de performance e tempo de execução, assim como o modelo que obteve o melhor resultado em negrito.

TABELA VI. PRINCIPAIS MODELOS DE REDE NEURAL ARTIFICIAL TREINADOS PARA A PROPOSTA DO ARTIGO

Modelo	Acurácia	F1-Score	Tempo de treinamento
CNN - 1	0.37	0.09	8min 8s
Transfer Learning - ResNet50 - 2	0.53	0.42	43min 32s
Transfer Learning - ResNet50 - 3	0.51	0.43	1h 32min 44s
<b>Transfer Learning - ResNet50 - 4</b>	<b>0.69</b>	<b>0.68</b>	<b>39min 2s</b>
Transfer Learning - ResNet50 - 5	0.23	0.15	15min 46s
Transfer Learning - ResNet50 - 6	0.68	0.66	46min 58s

Por fim, por se tratar de imagens com pouca relação entre si, já que as diferenças técnicas entre elas são pouco perceptíveis, se não for levado em consideração o período histórico e, muitas vezes a subjetividade que os autores quiseram passar ao espectador, o modelo se saiu bem.

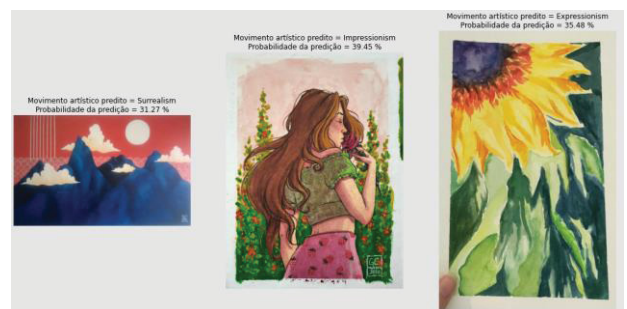


Fig. 46. Categorização de pinturas pela artista Giulia Camargo Marchi pelo modelo de rede neural artificial. Fonte: Elaborado pelo autor.

## V. CONSIDERAÇÕES FINAIS

Este trabalho teve um objetivo ambicioso, uma vez que classificar movimentos artísticos é uma tarefa difícil até mesmo para estudiosos do assunto. Isso porquê, como elaborado nos capítulos I e II, para classificar uma pintura em sua respectiva escola artística são levados em consideração vários critérios, como período histórico, região geográfica e

momento na vida do artista e sim, a técnica utilizada quando a pintura foi criada.

Sabendo disto, o modelo de aprendizado de máquina criado para este artigo levou em consideração somente a técnica empregada nas pinturas, como os padrões e escalas de linhas e objetos, cores e incidências de objetos, animais e pessoas nas obras.

Primeiramente o modelo foi treinado utilizando a base de dados completa, onde tinha-se uma disparidade grande entre a quantidade de imagens por movimento artístico, e isso levava o modelo a um viés muito grande em relação a qual categoria tinha mais amostras. Ao normalizar a quantidade de imagens por movimento artístico, apesar de diminuir consideravelmente a quantidade de amostras para o treinamento do modelo, o resultado obtido foi bem mais satisfatório, já estava ocorrendo menos *overfitting* e os dados de treinamento e validação estavam se correlacionando entre si.

Dentre as dezenas de treinamentos do modelo de Rede Neural Convolutiva e os *tunnings* de hiperparâmetros, pouco a pouco foi possível chegar a um resultado satisfatório. O modelo atingiu 69% de acurácia, com um viés balanceado entre as categorias e um gráfico de perdas respeitável.

É importante observar que, com um estudo aprofundado sobre quais obras mais representam seus respectivos movimentos artísticos, é possível aumentar a correlação e a acurácia do modelo consideravelmente. Uma vez que, para normalizar a quantidade de *samples* de cada escola artística, as imagens foram escolhidas aleatoriamente, sem critério nenhum além do movimento artístico ao qual ela é categorizada.

Em suma, o objetivo do estudo foi atingido, e por mais que o modelo ainda possa ser melhorado com a ajuda de um(a) especialista na área artística, esse modelo já poderia ser usado em escolas e cursos como um incentivo para a criação de pinturas e obras de arte.

#### A. Trabalhos futuros

O modelo classificatório de pinturas e desenhos que este artigo propôs explorou diversos conceitos de aprendizado de máquina, para aplicar a melhor categorização possível de movimentos artísticos nas obras fornecidas. Porém, a arte em geral, mas especialmente a arte visual é subjetiva, ou seja, a sua interpretação não vem somente do que é possível enxergar, mas também do contexto histórico e por quem a obra foi criada, onde geograficamente a pintura foi produzida e qual técnica de pintura foi utilizada em sua criação.

Da mesma forma, às vezes obras com o mesmo contexto histórico e classificada por especialistas como sendo do mesmo movimento artístico não tem correlação técnica, sendo que o principal objetivo da rede neural convolutiva aplicada neste artigo foi de analisar e descobrir padrões técnicos entre as pinturas para classificá-las corretamente em seus respectivos movimentos artísticos.



Fig. 47. Obra “Morte de Sardanápalo” (1827) pelo artista romantista Eugène Delacroix. Fonte: adaptado de WIKIART [24].



Fig. 48. Obra “Archway with Trees by the Sea” (1828) pelo artista romantista William Turner. Fonte: adaptado de WIKIART [24].

Nas figuras 47 e 48, é possível ver as diferenças técnicas entre duas pinturas românticas que prova que, se for levar em consideração somente padrões visuais, não é possível chegar a uma conclusão acurada de qual movimento artístico a obra se encaixa.

Levando tudo isto em consideração, e entendendo que o modelo que o artigo propôs só leva em consideração os padrões visuais das obras, e que isso não é o suficiente para ter uma acurácia alta em relação à identificação de obras artísticas, os próximos passos seriam treinar um modelo que leve em consideração características além do visual. Como o ano, e país em que a obra foi produzida.

Quanto aos próximos passos com o modelo que foi produzido neste artigo, já seria o suficiente para aplicá-lo em um sistema ou aplicativo educacional para ser usado em escolas e estudos de arte.

#### REFERÊNCIAS

- [1] WEITZ, Morris. The Role of Theory in Aesthetics. *The Journal of Aesthetics and Art Criticism*, v. 15, No. 1, p. 27-35, set. 1956.
- [2] ANDRADE, D. Q. S.; ARANTES, A. R. V.; A HISTÓRIA DO ENSINO DA ARTE NO BRASIL: tendências e concepções. *Revista de Magistro de Filosofia*, p. 107-120, set. 2016.
- [3] BRASIL, Lei nº 9.394 Art. 35-A, § 2º, de 20 de dezembro de 1996. O ensino médio, etapa final da educação básica, com duração mínima de três anos. Disponível em [http://www.planalto.gov.br/ccivil\\_03/leis/19394.htm](http://www.planalto.gov.br/ccivil_03/leis/19394.htm). Acesso em 10 de out. 2021.

- [4] FARIA, Diomira M.C.P. . Investigação sobre o visitante de museus de arte: uma comparação Brasil e Espanha. *Via@ Tourism Review* , v. 1, p. 1-18, 2015.
- [5] IPEA Instituto de Pesquisa Econômica Aplicada. Disponível em: [https://www.ipea.gov.br/portal/index.php?option=com\\_alphacontent&ordering=3&limitstart=4510&limit=10](https://www.ipea.gov.br/portal/index.php?option=com_alphacontent&ordering=3&limitstart=4510&limit=10). Acesso em 10 out. 2021.
- [6] BAHETI, Bhakti; INNANI, Shubham; GAJRE, Suhas; TALBAR, Sanjay. Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment. *IEEE Xplore*, Nanded, v. 28, p. 3286-3300, jul. 2020.
- [7] BADRINARAYANAN, Vijay; KENDALL, Alex; CIPOLLA, Roberto. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. *IEEE Xplore*, v. 39, p. 1-14, dec. 2017.
- [8] BAPPY, Jawadul H; SIMONS, Cody; NATARAJ, Lakshmanan; MANJUNATH, B.S; ROY-CHOWDHURY, Amit K. Hybrid LSTM and Encoder-Decoder Architecture for Detection of Image Forgeries. *IEEE Xplore*, v. 28, p. 3286-3300, jan. 2017.
- [9] CARLEO, Giuseppe; CIRAC, Ignacio; CRANMER, Kyle; DAUDET, Laurent; SCHULD, Maria; TISHBY, Naftali; VOGT-MARANTO, Leslie; ZDEBOROVÁ, Lenka. Machine learning and the physical sciences. *Rev. Mod. Phys.* p. 91, 2019.
- [10] CUNNINGHAM, Pádraig; CORD, Matthieu; DELANY, Sarah, Jane. Supervised Learning. *Machine learning techniques for multimedia*, Sprinder, Berlin, p. 21-49, 2008.
- [11] OLIVER, Jonathan J; BAXTER, Rohan A; WALLACE, Chris S. Unsupervised Learning Using MML. *ICML'96: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, p. 364-372, jul. 1996.
- [12] ABRAHAM, Ajith. Artificial Neural Networks. *Handbook of Measuring System Design*, jul. 2005.
- [13] SILVA, Ivan, Nunes; SPATTI, Danilo, Hernane; FLAUZINO, Rogerio, Andrade; LIBONI, Luisa, Helena, Bartocci; ALVES, Silas, Franco, Reis. *Artificial Neural Networks A Practical Course*. 24 ago. 2016.
- [14] MEDIUM. Disponível em: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>. Acesso em 03 out. 2021.
- [15] ADESHPANDE3. Disponível em: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>. Acesso em 03 out. 2021.
- [16] MEDIUM. Disponível em: <https://medium.com/coinmonks/convolutional-neural-network-in-a-nut-shell-107b5b9364ab>. Acesso em 03 out. 2021.
- [17] JEREMYJORDAN. Disponível em: <https://www.jeremyjordan.me/convolutional-neural-networks/>. Acesso em 03 out. 2021.
- [18] MEDIUM. Disponível em <https://medium.com/data-hackers/uma-introdu%C3%A7%C3%A3o-as-redes-neurais-convolucionais-utilizando-o-keras-41ee8dccc033e>. Acesso em 28 ago. 2021.
- [19] MACHINE Learning Mastery. Disponível em: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>. Acesso em 28 ago. 2021.
- [20] CHAUHAN, Rahul; GHANSHALA, K; JOSHI, R. Convolutional Neural Network (CNN) for Image Detection and Recognition. *First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. 2018.
- [21] YU, Dingjun; WANG, Hanli; CHEN, Peiqiu; WEI, Zhihua. Mixed Pooling for Convolutional Neural Networks. *The 9th International Conference on Rough Sets and Knowledge Technology*, Shanghai, China, out. 2014.
- [22] KAGGLE. Disponível em: <https://www.kaggle.com/ikarus777/best-works-of-all-time>. Acesso em 29 ago. 2021.
- [23] FLOREA, Corneliu; CONDOROVICI, Razvan; VERTAN, Constantin; BUTNARU, Raluca; FLOREA, Laura; VRANCEANU, Ruxandra. Pandora: Description of a Painting Database for Art Movement Recognition with Baselines and Perspectives. *Image Processing and Analysis Laboratory, University Politehnica of Bucharest*, Bucharest, v. 3, p. 1-5, 2016.
- [24] WIKIART. Disponível em: <https://www.wikiart.org/>. Acesso em 05 de out. 2021.
- [25] TENSORFLOW. Disponível em: [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator). Acesso em 23 de out. 2021.
- [26] MATHWORKS. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/resnet50.html>. Acesso em 23 de out. 2021.
- [27] IMAGENET. Disponível em: <http://www.image-net.org>. Acesso em 23 de out. 2021.
- [28] ABAMBRES, Miguel; RAJANA, Komal; TSAVDARIDIS, Konstantinos; RIBEIRO, Tiago Pinto. Neural Network-Based Formula for the Buckling Load Prediction of I-Section Cellular Steel Beams. *Research & Development, Abambres' Lab, 1600-275 Lisbon, Portugal*, dez. 2018.
- [29] CULTURAL, Saber. Disponível em: <http://www.sabercultural.com/template/especiais/IgrejaSantoInacio.html>. Acesso em 05 de out. 2021.
- [30] OVHCloud. Disponível em: <https://blog.ovhcloud.com/what-does-training-neural-networks-mean/>. Acesso em 20 de jan. de 2022.
- [31] AMBRÓSIO, P. E. Redes neurais artificiais no apoio ao diagnóstico diferencial de lesões intersticiais pulmonares. 2002. *Ribeirão Preto – SP. Dissertação (Mestrado) – Faculdade de Filosofia, Universidade de São Paulo*.
- [32] TOWARDSDATASCIENCE. Disponível em: <https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7>. Acesso em 20 de jan. de 2022.
- [33] KDNUGGETS. Disponível em: <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>. Acesso em 21 de jan. de 2022.
- [34] TOWARDSDATASCIENCE. Disponível em: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>. Acesso em 21 de jan. de 2022.
- [35] IBM. Disponível em: <https://www.ibm.com/cloud/learn/gradient-descent>. Acesso em 21 de jan. de 2022.
- [36] MEDIUM. Disponível em: <https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0>. Acesso em 21 de jan. de 2022.
- [37] LIU, D. A Practical Guide to ReLU. 2017. Disponível em <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>. Acesso em 6 mar 2022.
- [38] SHAMAR, Sagar. Activation Functions in Neural Networks. 2017. Disponível em <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>. Acesso em 6 mar 2022.
- [39] BONTON, Kris. A QUICK INTRODUCTION TO ARTIFICIAL NEURAL NETWORKS (PART 2). 2018. Disponível em <https://krisbolton.com/a-quick-introduction-to-artificial-neural-networks-part-2>. Acesso em 6 mar 2022.
- [40] BROWNLEE, Jason. Softmax Activation Function with Python. 2020. Disponível em <https://machinelearningmastery.com/softmax-activation-function-with-python/>. Acesso em 6 mar 2022.
- [41] TAN, Chuanqi; SUN, Fuchun; KONG, Tao; ZHANG, Wenchang; YANG, Chao; LIU, Chunfang. A Survey on Deep Transfer Learning. *Lecture Notes in Computer Science*, vol 11141. Springer, Cham, set. 2018.
- [42] ZHUANG, Fuzhen; QIL, Zhiyuan; DUAN, Keyuu; XI, Dongbo; ZHU, Yongchun; ZHUU, Hengshu; XIONG, Hui; HE, Qing. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, v. 109, p. 43-76, jan. 2021.
- [43] SHORTEN, C.; KHOSHGOFTAAR, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, v.60, jul. 2019.
- [44] SHORTEN, C.; KHOSHGOFTAAR, T.M.; FURHT, B. Text Data Augmentation for Deep Learning. *J Big Data*, v. 8, p.101, jul. 2021.
- [45] HAYKIN, S. *Redes Neurais- Princípios e Práticas*. BOOKMAN, São Paulo, 2ª ed. 2001. 900 p.
- [46] PAN, S.J; YANG, Qiang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, v.22, n.10, out 2010.
- [47] SVOZIL, Daniel; KVASNICKA, Vladimír; POSPÍČHAL, JIRÍ. Introduction to multi-layer feed-forward neural networks, v. 39, p.43-62, jun. 1997.
- [48] ZAREMBA, Wojciech; SUTSKEVER, Ilya; VINYALS, Oriol. Recurrent Neural Network Regularization, *arXiv*, set. 2014.

- [50] CAELEN, Olivier. A Bayesian Interpretation of the Confusion Matrix, *Ann Math Artif Intell*, vol. 81, p. 429-450, set. 2019.
- [51] REZAEI, Samaneh; TAHMORESNEZHAD, Jafar; SOOLOUK, Vahid. A transductive transfer learning approach for image classification. *International Journal of Machine Learning and Cybernetics*, vol. 12, p. 747-762, set. 2020.
- [52] NWANKPA, C., E.; IJOMAH, W.; GACHAGAN, A.; MARSHALL, S. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *2nd International Conference on Computational Sciences and Technology*, p. 124 - 133, 2021.
- [53] SCULLEY, D.; HOLT, Gary; GOLOVIN, Daniel; DAVYDOV, Eugene; PHILLIPS, Todd; EBNER, Dietmar; CHAUDHARY, Vinay; YOUNG, Michael; CRESPO, Jean-François; DENNISON, Dan. Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems*, vol. 28, 2015.