DOI: 10.5748/19CONTECSI/REX/DSC/6975

ANÁLISE DE ALGORITMOS DE RECONHECIMENTO FACIAL: HAAR CASCADE (VIOLA-JONES) E CNN (CONVOLUTIONAL NEURAL NETWORKS)

Dilermando Piva Jr ; https://orcid.org/0000-0002-2534-9618
Fatec Itu / Fatec Sorocaba

André Santos Alckmin de Carvalho ; https://orcid.org/0000-0003-3891-0803
Fatec Itu

ANÁLISE DE ALGORITMOS DE RECONHECIMENTO FACIAL: HAAR CASCADE (VIOLA-JONES) E CNN (CONVOLUTIONAL NEURAL NETWORKS)

ANALYSIS OF FACIAL RECOGNITION ALGORITHMS: HAAR CASCADE (VIOLA-JONES) AND CNN (CONVOLUTIONAL NEURAL NETWORKS)

RESUMO: A Visão Computacional vem ganhando destaque pela sua premissa de capacitar máquinas a verem o mundo como humanos fazem, sendo capazes de entender de maneira similar e utilizar o conhecimento adquirido para diversas tarefas, como Detecção e Reconhecimento de Imagem e Vídeo, Sistemas de Recomendação, Processamento de Linguagem Natural etc. O objetivo deste trabalho é descrever dois dos principais métodos e algoritmos de detecção e reconhecimento facial: Haar Cascade (algoritmo apresentado em 2001, é um *framework* proposto por Paul Viola e Michael Jones cujo objetivo é detecção de objetos prática e rapidamente) e Redes Neurais Convolucionais (algoritmo introduzido por Yann Lecun em 1998 que, por sua vez, possui foco na detecção e reconhecimento de objetos, sendo amplamente utilizando em contextos diversos). Os testes para comparação e análise dos algoritmos foram realizados em uma máquina com as seguintes especificações técnicas: Processador Amd Ryzen 5 1600 AF, Placa Mãe Asus B450M Gaming, Memória RAM 16GB 3200MHz DDR4, SSD NVMe 256GB, Placa de Vídeo RX 580 4GB. Além disso, vale notar que o desenvolvimento da aplicação foi realizado através de funções individuais criadas na linguagem de programação Python versão 3.9.6 que, através da mesma, analisaram 1000 imagens obtidas das bases de imagens YaleFaces, Labeled Faces in the Wild (LFW) e Dogs vs Cats (possuindo 600 imagens positivas 400 negativas). Como resultado, pode-se observar que o algoritmo CNN chegou a uma acurácia de 98,9%, contra 81,7 do algoritmo Haar Cascade.

ABSTRACT: Computer Vision has been gaining prominence for its premise of enabling machines to see the world as humans do, being able to understand in a similar way and use the acquired knowledge for different tasks, such as Image and Video Detection and Recognition, Recommender Systems, Processing of Natural Language etc. The objective of this work is to describe two of the main methods and algorithms for detection and facial recognition: Haar Cascade (an algorithm presented in 2001, it is a framework proposed by Paul Viola and Michael Jones whose objective is to detect objects in a practical and fast way) and Convolutional Neural Networks. (algorithm introduced by Yann Lecun in 1998 which, in turn, focuses on the detection and recognition of objects, being widely used in different contexts). The tests for comparison and analysis of the algorithms were carried out on a machine with the following technical specifications: AMD Ryzen 5 1600 AF Processor, Asus B450M Gaming Motherboard, RAM 16GB 3200MHz DDR4, SSD NVMe 256GB, Video Card RX 580 4GB. Furthermore, it is worth noting that the development of the application was carried out through individual functions created in the Python programming language version 3.9.6 which, through it, analyzed 1000 images obtained from the YaleFaces, Labeled Faces in the Wild (LFW) image databases. and Dogs vs Cats (having 600 positive and 400 negative images). As a result, it can be seen that the CNN algorithm reached an accuracy of 98.9%, against 81.7 for the Haar Cascade algorithm.

PALAVRAS-CHAVE: Visão Computacional. Haar Cascade. Redes Neurais Convolucionais. Aprendizagem Profunda. Aprendizado de Máquina.

KEYWORD: Computer Vision. Haar Cascade. Convolutional Neural Networks. Deep Learning. Machine Learning.

1 INTRODUÇÃO

A Visão Computacional vem ganhando destaque pela sua premissa de capacitar máquinas a verem o mundo como humanos fazem, sendo capazes de entender de maneira similar e utilizar o conhecimento adquirido para diversas tarefas, como Detecção e Reconhecimento de Imagem e Vídeo, Sistemas de Recomendação, Processamento de Linguagem Natural etc. O objetivo deste trabalho é descrever dois dos principais métodos e algoritmos de detecção facial: Haar Cascade e Redes Neurais Convolucionais. Os testes serão realizados através de funções individuais criadas na linguagem de programação Python. Os dados retornados pelas funções serão explorados e detalhados, trazendo informações como tempo de processamento de cada imagem, acurácia e representações gráficas, por exemplo.

2 METODOLOGIA

Para comparar dois algoritmos, deve-se, inicialmente, preparar um ambiente onde não existem fatores externos que irão interferir no que se deseja medir. Neste caso, a variável desejada é o desempenho. Para isso, foi preparado uma máquina com as seguintes especificações técnicas: Processador Amd Ryzen 5 1600 AF, Placa Mãe Asus B450M Gaming, Memória RAM 16GB 3200MHz DDR4, SSD NVMe 256GB, Placa de Vídeo RX 580 4GB. Além disso, também foi utilizada uma base de imagens que constitui um conjunto das bases de imagens YaleFaces, Labeled Faces in the Wild (LFW) e Dogs vs Cats (possuindo 600 imagens positivas 400 negativas).

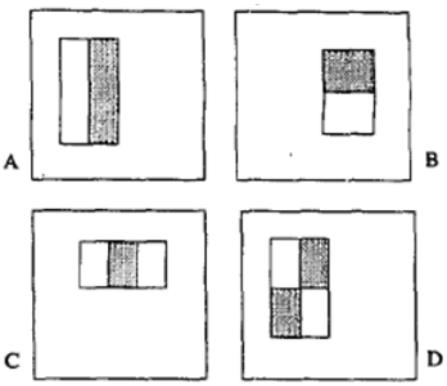
3 DESENVOLVIMENTO

3.1 HAAR CASCADE (*Viola-Jones*)

O framework desenvolvido por Paul Viola e Michael Jones, Detecção Rápida de Objetos utilizando uma Cascata Impulsionada de Recursos Simples, tem como base três contribuições-chave introduzidas pelos autores. A primeira é a Imagem Integral, uma representação de imagem criada para realizar cálculos eficientemente. A segunda é um algoritmo de aprendizado baseado em AdaBoost, que seleciona um pequeno grupo de recursos críticos de um grupo maior [Bartlett, Freund, Lee e Schapire 1998]. A terceira é um método de combinar classificadores cada vez mais complexos numa estrutura de cascata, aumentando a velocidade do detector focando somente nas regiões promissoras da imagem [VIOLA e Jones 2001]. A cascata pode ser vista como um mecanismo de foco de atenção específico do objeto que, ao contrário de abordagens de detecção de objetos anteriores, é capaz de fornecer garantias estatísticas de que regiões descartadas não possuem, de fato, o objeto de interesse. Resumidamente, o processo de análise de um objeto utilizando o framework Viola-Jones ocorreria da seguinte forma: o algoritmo recebe uma imagem de entrada. A partir desta, é calculada sua Imagem Integral. Em sequência, serão aplicados retângulos (recursos) em uma ordem já previamente definida que, a partir da diferença de luminosidade, indicarão se aquele fragmento da imagem possui chances de ser uma face ou não. Conforme os recursos indicarem que há chance de o fragmento da imagem possuir um rosto presente, serão aplicados recursos cada vez mais complexos e específicos, aumentando a probabilidade de acerto.

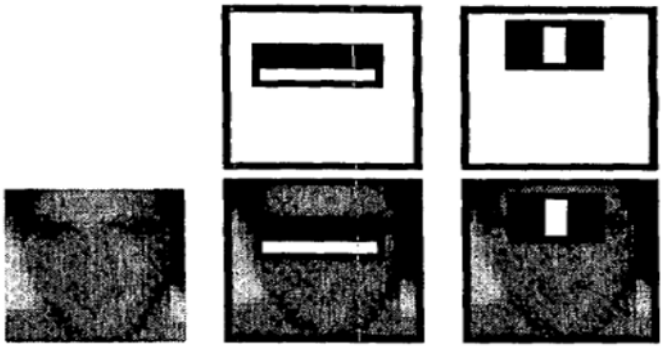
Os principais Recursos Haar são apresentados na Figura 1. A aplicação destes recursos em locais específicos e pré-determinados da imagem, por sua vez, é responsável pelo cálculo d Imagem Integral. É possível observar que os Recursos Haar A e C foram aplicados na face presente na Figura 2.

Figura 1 – Recursos Haar.



Fonte: (VIOLA; JONES, 2001, p. 2).

Figura 2 – Principais aplicações dos Recursos Haar.

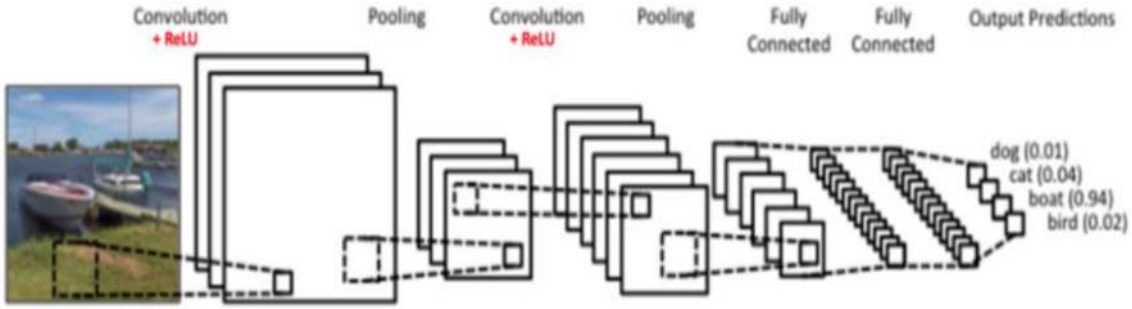


Fonte: *(VIOLA; JONES, 2001, p. 4)*.

3.2 CNN (*Convolutional Neural Networks*)

Uma CNN ou ConvNet (abreviação de Convolutional Neural Network — Rede Neural Convolucional, em português) é um algoritmo de Aprendizado Profundo capaz de captar uma determinada entrada, atribuir importância através de filtros a diferentes aspectos e diferenciar entre seus principais elementos. Elaborado por Yann Lecun em 1998, é uma das obras-primas da matemática aplicada a redes neurais artificiais e reconhecimento de padrões. Para avaliar uma imagem, por exemplo, a CNN utiliza recursos (*features*, em inglês). Um recurso é, em sua forma mais básica, um fragmento da imagem. Ao encontrar o mesmo recurso em regiões próximas da imagem, a Rede é capaz de entender a similaridade entre as duas regiões mais facilmente do que comparando imagens inteiras. A Rede, inicialmente, aplica recursos em todas as posições possíveis, formando um filtro. A matemática utilizada neste processo é chamada de convolução. Quando uma imagem de entrada é apresentada, os recursos críticos (aqueles que diferem eficientemente imagens negativas de positivas) são devidamente alinhados acima da mesma. Em sequência, seus valores de pixel são multiplicados entre si, criando uma matriz bidimensional. Se os pixels forem iguais, o resultado será 1, do contrário, 0. Em sequência, somam-se os valores da matriz. Finalmente, a soma é dividida pela quantidade de pixels presentes no recurso utilizado. Se todos os pixels forem iguais entre si, o resultado será 1. Da mesma forma, se todos os pixels forem diferentes entre si, o resultado será -1 (1 negativo). Os valores obtidos após a convolução devem ser armazenados em uma nova matriz bidimensional. Em sequência, novas matrizes serão criadas de acordo com as camadas que forem aplicadas. Por exemplo, a Camada de Agrupamento (*Pooling*, em inglês) é responsável por reduzir as dimensões de imagens enquanto armazena suas principais informações. Em outras palavras, recebe uma matriz e, através de cálculos, comprime as informações em uma nova matriz bidimensional. Após a imagem de entrada percorrer todas as camadas presentes da ConvNet, será produzido um achatamento (camada completamente conectada) e com base nessa última camada, é então produzida uma saída contendo a probabilidade de existir uma face. A Figura 3 representa visualmente os processos que uma Rede Neural Convolucional realiza. Em termos básicos, diferentes fragmentos da imagem seriam selecionados sequencialmente, onde as camadas presentes na ConvNet aplicariam suas respectivas funções. Por exemplo, a camada de *pooling* seria responsável por reduzir as dimensões da matriz (fragmento de imagem) recebida. Finalizando, após a aplicação de todas as camadas, é gerada a saída que, por sua vez, deve ser utilizada conforme a necessidade da aplicação. A quantidade de camadas de convolução e de agrupamento são variáveis e dependem também do objetivo pretendido e da aplicação.

Figura 3 – Representação dos processos de uma ConvNet.



Fonte: (Soares, 2018, p. 92.

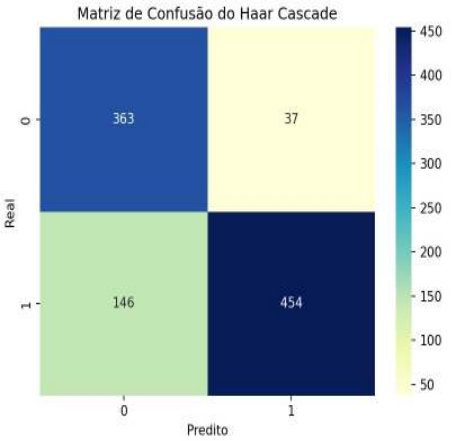
4 RESULTADOS OBTIDOS

4.1 Resultados *Haar Cascade*

As Figuras 4, 5 e 6 mostram os resultados obtidos ao utilizar o algoritmo Haar Cascade através da matriz de confusão, curva ROC e um gráfico apresentando os tempos de detecção em função do tempo, respectivamente.

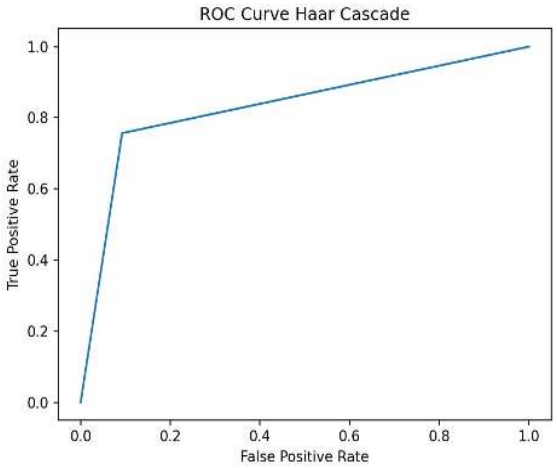
A matriz de confusão pode ser utilizada para avaliações de modelos de classificação em Aprendizado de Máquina, assim como foi utilizada neste relatório. Em termos básicos, recebe como entrada a possibilidade de uma situação ser verdade ou não e, baseando-se nesta, o resultado real do teste da aplicação. Ou seja, é uma tabela que apresenta a taxa de verdadeiro positivos (onde os dados obtidos são verdadeiros em ambas as ocasiões), falso positivos (dado real é falso, mas o obtido é verdadeiro), falso verdadeiro (ambos os dados são falsos) e falso negativo (dado real é verdadeiro, mas o obtido é falso). Analisando a Figura 4, nota-se que os resultados obtidos ao executar o algoritmo Haar Cascade foram 454 imagens verdadeiro positivas, 146 imagens falso positivas, 363 imagens falso verdadeiras e 37 imagens falso negativas. A curva ROC (abreviação de *Receiver Operating Characteristic Curve* — Curva Característica de Operação do Receptor, em português), segundo Avelar (2019), a curva ROC é uma curva de probabilidade, criada traçando a taxa de verdadeiro positivos contra a taxa de falso positivos. Em outras palavras, o número de vezes que o classificador acertou contra o número de vezes que o classificador errou a predição. Tipicamente, representa a taxa de positivo verdadeiros no eixo das ordenadas e a taxa de falso positivos no eixo das abcissas. Ou seja, o canto superior esquerdo do gráfico é o ponto ideal, onde a taxa de falso positivos é mínima (0) e a taxa de verdadeiro positivos é máxima (1). Neste trabalho, a curva é utilizada para indicar a acurácia dos algoritmos. Ao analisar a Figura 5, onde a curva ROC apresenta a acurácia do algoritmo Haar Cascade, observa-se que o resultado obtido foi de 81,7%. Ou seja, a soma de verdadeiro positivos com verdadeiro negativos dividido pela soma de verdadeiro positivos, verdadeiro negativos, falso positivos e falso negativos resulta em 81,7%. Finalizando, o gráfico apresentado na Figura 6 demonstra o tempo de processamento resultante da detecção individual de imagens. Nota-se que houve um pico inicial, onde se obteve uma máxima de aproximadamente 0,38 segundos. Em compensação, o tempo de detecção seguiu numa média de 0,05 segundos de tempo de processamento por imagem.

Figura 4 – Matriz de Confusão do Haar Cascade.



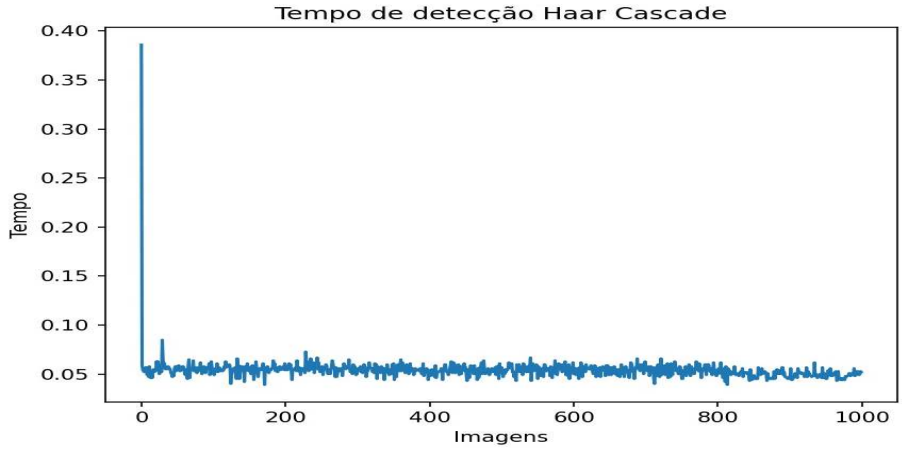
Fonte: Próprios autores.

Figura 5 – Curva ROC do Haar Cascade.



Fonte: Próprios autores.

Figura 6 – Tempo de detecção do Haar Cascade.



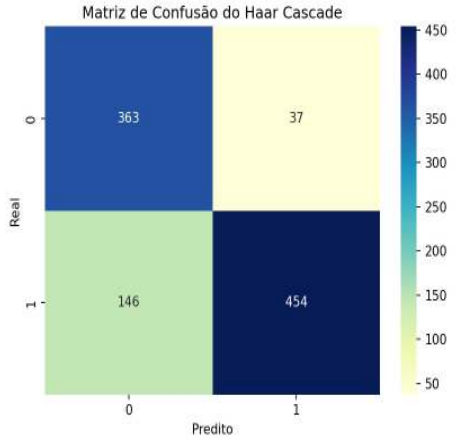
Fonte: Próprios autores.

4.2 Resultados Convolutional Neural Networks

As Figuras 7, 8 e 9, por sua vez, também apresentam os resultados obtidos através das mesmas técnicas ao utilizar o algoritmo CNN.

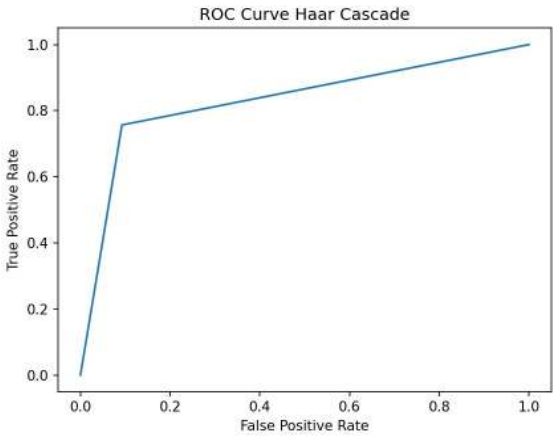
Observando a Figura 7, os resultados obtidos foram 589 imagens verdadeiro positivas, 11 imagens falso positivas, 400 imagens falso verdadeiras e 0 imagens falso negativas. Somando o total de valores de cada algoritmo, é obtido individualmente o valor 1000 (quantidade total de imagens). Analisando a curva ROC do algoritmo CNN, conforme Figura 8, nota-se que a acurácia é aproximadamente 100% (exatamente 98,9%), apresentando um resultado muito superior àquele obtido através do algoritmo Haar Cascade (onde sua acurácia foi de 81,7%). Finalmente, ao observar o gráfico apresentado na Figura 9, é possível notar o tempo de detecção diminuiu drasticamente após o marco de 400 imagens. Isso ocorre, pois, imagens positivas necessitam de menos processamento computacional, já que possuem características claras de que a imagem se trata de uma face.

Figura 7 – Matriz de Confusão do CNN.



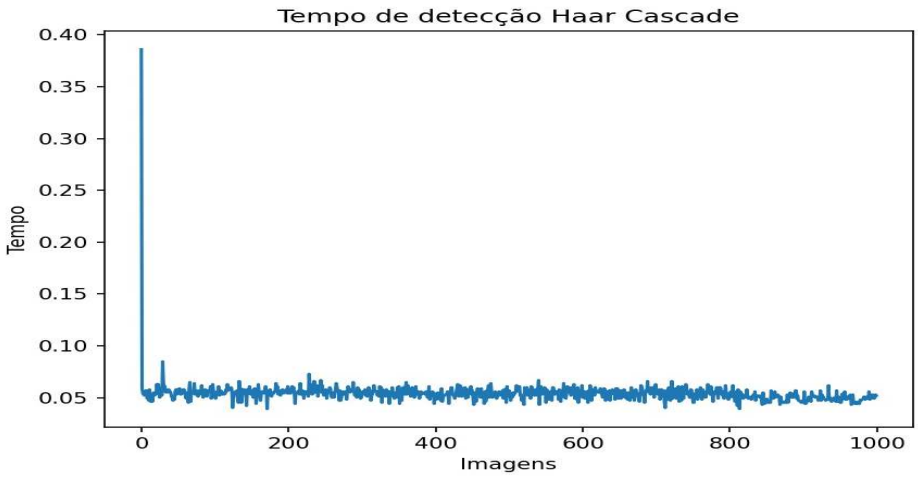
Fonte: Próprio autor.

Figura 8 – Curva ROC do CNN.



Fonte: Próprio autor.

Figura 9 – Tempo de detecção do CNN.



Fonte: *Próprio autor*.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, foram apresentados os seguintes itens: introdução sobre a subárea da Inteligência Artificial conhecida como Visão Computacional; explicação e funcionamento de dois métodos e algoritmos focados em Detecção Facial (Haar Cascade e Redes Neurais Convolucionais); análise individual detalhada dos algoritmos apresentados através de diferentes métodos e técnicas (matriz de confusão, curva ROC e gráfico de tempo de processamento) e descrição e aplicação dos algoritmos em linguagem Python. Comparando a velocidade dos dois algoritmos apresentados, nota-se que o Haar Cascade (baseado somente em cálculos matemáticos) obteve resultados muito mais rápidos, com uma média de 0,05 segundos. Observando a Figura 9, fica claro que a média do tempo de processamento deve ser analisada a fundo, considerando a brusca diferença entre detecção de imagens positivas e negativas. Em compensação, em ambos os casos a detecção foi baseada em segundos completos, diferentemente do primeiro algoritmo, onde esta foi baseada em milissegundos.

Em compensação, em uma análise geral, a acurácia do Haar Cascade foi extremamente inferior àquela obtida através do CNN (baseado em redes neurais). As matrizes de confusão apresentadas evidenciam este fato, já que a utilização de redes neurais possibilitou resultados muito mais precisos, onde a taxa de falso positivos e falso negativos foi baixa (aproximadamente 0). Ainda sobre a acurácia dos algoritmos, a curva ROC gerada pelo algoritmo CNN forma um ângulo de aproximadamente 90° graus, evidenciando, portanto, maior acurácia dentre as duas apresentadas. Logicamente, ao observar a curva obtida pelo algoritmo Haar Cascade, nota-se um ângulo muito mais acentuado, demonstração gráfica da acurácia obtida (81,7%). Dessa forma, constata-se o grande avanço que as redes neurais trouxeram para a subárea da Visão Computacional. Permitindo resultados muito mais acurados, promissores e possíveis de se depender.

REFERÊNCIAS

COMPUTERPHILE. YouTube, 2016. **CNN: Convolutional Neural Networks Explained** - Computerphile. Disponível em: https://www.youtube.com/watch?v=py5byOOHZM8 . Acesso em 22 mar. 2022.

IBM TECHNOLOGY. YouTube, 2021. **What are Convolutional Neural Networks (CNNs)**?. Disponível em: https://www.youtube.com/watch?v=QzY57FaENXg . Acesso em 22 mar. 2022.

ROHRER, Brandon. YouTube, 2016. **How Convolutional Neural Networks work**. Disponível em: https://www.youtube.com/watch?v=FmpDIaiMIeA . Acesso em 22 mar. 2022.

Schapire, R. E., Freund, Y., Bartlett, P., e Lee, W. S. (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. **The Annals of Statistics,** 26(5), 1651–1686. doi:10.1214/aos/1024691352.

SHANMUGAMANI, R. **Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras**. Birmingham, 2018.

Soares, A. (2018). **Redes Neurais Profundas – Deep Learning**, http://ww2.inf.ufg.br/~anderson/deeplearning/20181/Aula%20-%20Redes%20Neurais%20Convolucionais%20Parte%20I.pdf . Julho.

STATQUEST WITH JOSH STARMER. YouTube, 2021. **Neural Networks Part 8: Image Classification with Convolutional Neural Networks**. Disponível em: https://www.youtube.com/watch?v=HGwBXDKFk9I . Acesso em 22 mar. 2022.

Viola, P., e Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001**. doi:10.1109/cvpr.2001.990517.