

---

**RECONHECIMENTO DE IMAGEM PARA IDENTIFICAÇÃO DE PISOS  
TÁTEIS EM AMBIENTES URBANOS**

**IMAGE RECOGNITION FOR TACTILE SURFACE IDENTIFICATION IN URBAN  
ENVIRONMENTS**

Daniel Cunha Stock<sup>1</sup>  
Simone Sawasaki Tanaka<sup>2</sup>

**RESUMO**

A locomoção de pessoas com algum tipo de deficiência sensorial visual em ambientes urbanos é um desafio à independência destes indivíduos. Como alternativa para garantia de segurança, orientação e mobilidade, é definido pela Associação Brasileira de Normas Técnicas (ABNT) um padrão arquitetônico de sinalização tátil no piso. Neste trabalho realizou-se um estudo sobre o funcionamento de redes neurais artificiais, sua estrutura, parametrizações, estratégias de treinamento e avaliação dos resultados. São apresentadas as camadas básicas, de redes neurais convolucionais, para utilização em problemas de classificação de imagens com o auxílio do framework de aprendizado de máquina PyTorch, criação de um dataset de imagens especializado no tema do trabalho e ferramentas para acompanhamento das épocas de aprendizado do algoritmo e avaliação dos resultados.

35

**Palavras-chave:** acessibilidade; deep learning; classificação de imagem; redes neurais convolucionais.

**ABSTRACT**

The mobility of people with some type of visual sensory impairment in urban environments is a challenge to working conditions. As an alternative to safety guarantee the Brazilian Association of Technical Norms (ABNT) defined a tactile surface standard. In this work, a study was carried out on the functioning of artificial neural networks, their structuring, parameterizations, training strategies and evaluation. The basic layers of convolutional neural networks are presented for use in image classification problems with the help of the machine learning framework PyTorch. Also it covers the creation of an image dataset specialized in the subject of the work and tools for monitoring the algorithm learning epochs and evaluating its results.

**Keywords:** accessibility; deep learning; image classification; convolutional neural networks.

---

<sup>1</sup> Graduado em Engenharia de Software, Centro Universitário Filadélfia – UniFil. Departamento de Computação. Londrina – Paraná – Brasil. 86020-000 – daniel.stock@edu.unifil.br

<sup>2</sup> Docente do Centro Universitário Filadélfia - Unifil. Departamento de Computação Londrina - Paraná - Brasil. 86020-000 - simone.tanaka@unifil.br

## 1 INTRODUÇÃO

Pessoas com algum tipo de deficiência sensorial visual enfrentam desafios diários de locomoção em ambientes dinâmicos e desconhecidos, como ruas e ambientes comerciais. (GUERREIRO *et al.*, 2019). Como alternativa para garantia de gradual independência à Pessoa com Deficiência (PCD) sensorial visual, é definido pela Associação Brasileira de Normas Técnicas (ABNT) um padrão arquitetônico de sinalização tátil no piso. (ABNT, 2016).

Apesar de definidos padrões de acessibilidade, esforços são necessários para a verificação da aplicação dos mesmos. Redes neurais convolucionais, (Convolutional Neural Networks - CNN), oferecem um grande suporte para o reconhecimento e classificação de imagens, alcançando o estado da arte em várias implementações (YAN *et al.*, 2016).

Neste projeto foi desenvolvida uma CNN, com a função de detectar a presença de piso tátil. Os dados para treinamento foram coletados do software de rotas Google Street View (GSV) e a estrutura da ANN foi implementada na plataforma de desenvolvimento Google Colaboratory, com auxílio do framework de aprendizado de máquina PyTorch.

Os resultados obtidos ao final do estudo apontaram a necessidade de mais treinamento da rede neural, em decorrência de *datasets* de treinamento pequenos e de mescla de categorias entre os dados de treinamento. Entretanto, os modelos de CNN implementados no trabalho podem servir de apoio a estudos semelhantes.

## 2 DESENVOLVIMENTO

Redes neurais artificiais (ANN) tentam simular o funcionamento do cérebro biológico, espelhando-se no comportamento de neurônios (NORIEGA, 2005). Analogamente a redes neurais biológicas, em que neurônios são responsáveis pelas conexões e reações, em redes artificiais, estruturas chamadas de perceptrons assumem responsabilidade de reconhecer estímulos, transmitir informações e conceder respostas (ROSENBLATT, 1958).

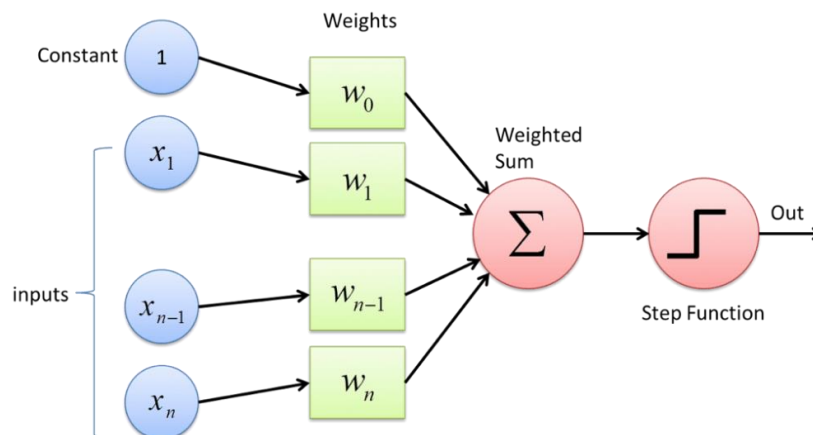
Segundo Kanal (2003), perceptrons são conjuntos de unidades sensoriais (S),

de associação (A) e de resposta (R). As unidades (S) recebem os dados de entrada e se ligam a unidades (A), que transformam os dados de acordo com pesos e a sua função objetivo. Os pesos são definidos durante o treinamento do algoritmo, após uma sequência suficiente de iterações (FREUND; SCHAPIRE, 1999).

Esses pesos variam entre  $[-1, 1]$  e são multiplicados pela resposta (R) que a unidade (A) definir. A partir da entrada de um conjunto de dados  $x_i$ , onde  $i = 1, 2, 3, \dots, N$ , é computada a soma dos pesos. Tendo como base um limite  $\theta$  (threshold), a ativação  $y$  do perceptron depende da soma exceder ou não o limite  $\theta$  definido (KANAL, 2003).

Em uma rede neural, cada perceptron é capaz de realizar uma ativação, com base no dado de entrada que ela recebe. Essa ativação pode ser passada para outra célula como entrada ou ser a saída final da rede neural (GALLANT, 1990). A Figura 1 apresenta as operações de processamento de um conjunto de entrada  $x_i$ .

**Figura 1 - Perceptron.**



Fonte: Sharma (2017).

Noriega (2005) descreve os pesos representados como uma matriz  $w_{io}$ , em que  $i$  são as linhas de dados de entrada e  $o$  são as colunas de possíveis dados de saída. Assim, o dado é multiplicado pela camada de pesos e enviado à camada de saída para ser processado pela função de ativação.

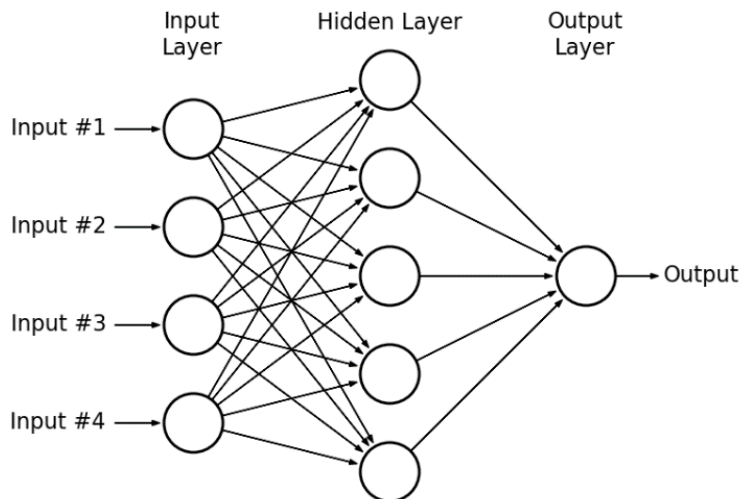
Funções de ativação têm o principal objetivo de converter o processamento de um perceptron, ou neurônio de uma rede neural artificial, em sinais de saída (FALCAO *et al.*, 2019).

A ReLU é uma função de ativação que tem provado ser uma solução muito eficiente no treinamento de redes neurais (HUSSEIN; ABDALMAGEED, 2021). Um comportamento dessa função que explica os seus resultados é a forma como trata valores negativos, que são igualados a zero, enquanto valores positivos permanecem iguais. O que reduz a complexidade e o tempo de treinamento (DUBEY; SINGH; CHAUDHURI, 2021).

A utilização de perceptrons passa a ser uma escolha ineficiente quando são enfrentados problemas não linearmente separáveis (NORIEGA, 2005). Para esse tipo de problema são utilizados multilayer perceptrons (MLP), estruturas na forma de grafos direcionados com vértices e arestas, denominados neurônios e sinapses (GÜNTHER; FRITSCH, 2010).

MLP são compostas por uma camada de entrada (*input layer*), múltiplas camadas escondidas (*hidden layer*) e uma camada de saída (*output layer*) (GORJANI *et al.*, 2021). A Figura 2 apresenta a estrutura de camadas padrão de um MLP.

**Figura 2** - Multilayer Perceptron.



**Fonte:** Mohamed *et al.* (2015).

Para cada sinapse, um peso é adicionado, indicando o efeito do neurônio correspondente e a informação é transmitida pela rede neural na forma de sinais. Os sinais são primeiro processados pela função de integração, que realiza a soma de todos os sinais recebidos, e então pela função de ativação, no neurônio da camada de saída (GÜNTHER; FRITSCH, 2010).

Os neurônios podem ser organizados em diversas topologias, entretanto a mais comum é conhecida por *feedforward*, na qual a informação é passada camada por camada até o fim do MLP (GROSSI; BUSCEMA, 2008). De acordo com Grossi (2008), redes neurais artificiais são sistemas adaptativos inspirados no funcionamento do cérebro humano, que ajustam a própria estrutura de acordo com uma função objetiva.

O aprendizado, ou treinamento, de uma rede neural artificial pode ser definido como o processo de identificação de todos os pesos  $w$  que, dado um conjunto de entradas  $x$ , melhor se combinam para fornecer uma saída  $O$  mais próxima do resultado esperado (RAMCHOUN *et al.*, 2017). Durante o treinamento são expostos conjuntos de exemplos aos quais a ANN deve se ajustar (GALLANT, 1990).

O algoritmo conhecido como *backpropagation* funciona com a definição de valores aleatórios para todos os pesos das camadas escondidas do MLP. Então ciclos iterativos, chamados de épocas, de ajustes são aplicados aos pesos, comparando as saídas geradas das saídas esperadas (MOHAMED; ZAHRAN; SAAVEDRA, 2015).

Visando minimizar a distância entre o resultado da ANN e o valor esperado, algumas estratégias de otimização são empregadas para garantir a qualidade da predição final. A essas configurações dá-se o nome de função de custo e função de otimização (RODRIGUES, 2018).

A função de custo tem o papel de avaliar a distância da predição para rótulos informados e retornar um escalar, cujo objetivo é ser minimizado entre as épocas (KEREN, 2017). Entre as funções mais difundidas está a de entropia-cruzada (cross-entropy), que minimiza o erro ao calcular a derivada parcial e verificar o comportamento do Gradiente Descendente ( $\nabla$ ) (RODRIGUES, 2018).

Durante a etapa de aprendizado é muito importante que a ANN não atualize os pesos de forma a fazer o Gradiente Descendente movimentar-se muito rápido, nem muito devagar (TEACHEY; KIPPING, 2021). Para realizar esse controle, existem hiperparâmetros que definem o passo de atualização desses dados, são eles a taxa de aprendizado (*learning rate*) e a decaimento de peso (*weight decay*).

A taxa de aprendizado é um escalar que representa o tamanho do passo de atualização dos pesos  $w$  da ANN (LI *et al.*, 2020). O ajuste e otimização desse valor é essencial para a otimização das épocas, pois determina o quão rápido o modelo

alcançará o mínimo local da função de perda (MAEDA-GUTIÉRREZ *et al.*, 2019).

Assim como redes neurais tradicionais, CNN são divididas em camada de entrada, camadas escondidas e camada de saída. Entretanto no contexto convolucional, essas camadas são chamadas de camada convolucional, camadas de *pooling* (agrupamento) e totalmente conectadas, ou *fully-connected* (FC) (SAKIB *et al.*, 2018). A Figura 3 traz a representação de uma imagem, como dado de entrada da CNN, no formato matricial (altura, largura e profundidade).

As camadas convolucionais da CNN são formadas por um conjunto de filtros, também conhecidos como *kernel*, que são aplicados no dado de entrada (ARAUJO *et al.*, 2017). Esses filtros, quando aplicados na imagem, produzem *feature maps*, que podem ativar determinados neurônios se forem identificados os padrões desejados no dado de entrada (NGUYEN *et al.*, 2021).

**Figura 3** - Imagem de entrada com 3 canais de cores.



Fonte: Araújo *et al.* (2017).

Ao processo de transformação do volume em um *feature map* é dado o nome de convolução e três parâmetros determinam o tamanho da saída desta etapa: profundidade (*depth*), passo (*stride*) e *zero-padding*. A profundidade determina o tamanho do filtro, que possui forma matricial, linhas x colunas, e o passo dita a movimentação, pixel a pixel, que o filtro percorrerá o volume (MURPHY, 2016).

A operação de *zero-padding* é um recurso que permite redimensionar imagens pequenas, adicionando zeros às bordas da matriz. Favorecendo conjuntos de dados que tenham dimensões muito pequenas, em relação ao tamanho do filtro (HASHEMI, 2019).

A equação a seguir expressa a perda de dimensão espacial que a imagem sofre ao passar pelas camadas da CNN. O cálculo é feito com base nos parâmetros definidos na camada convolucional, sendo  $X_{out}$  a dimensão de saída da imagem,  $X_{in}$  a dimensão de entrada,  $F$  o tamanho do filtro convolucional,  $P$  o padding aplicado e  $S$  o passo de deslocamento do filtro (HOSSAIN; SAJIB, 2019).

$$X_{out} = ((X_{in} - F + 2P)/S) + 1$$

A saída da camada de convolução serve de entrada para a camada de agrupamento. Nesta camada normalmente é aplicada uma função de máxima (MAX) (SAKIB *et al.*, 2018). No agrupamento busca-se "reduzir progressivamente a dimensão espacial do volume de entrada, conseqüentemente a redução diminui o custo computacional da rede e evitar o *over-fitting*", que é o treinamento em excesso de particularidades do conjunto de aprendizado. (ARAUJO *et al.*, 2017).

A camada totalmente conectada de uma CNN tem a função de classificar um dado de entrada em uma classe, previamente determinada durante o aprendizado supervisionado. O funcionamento desta camada é exatamente igual ao de redes neurais tradicionais, com a implementação de MLP (ARAUJO *et al.*, 2017).

41

### 3 ESTUDO DE CASO

A implementação da CNN deu-se início pela coleta de imagens e criação de dataset específico para diferenciação de pisos táteis e pisos não-táteis. Posteriormente, foi implementada uma adaptação do modelo LeNet 5.

Com base nas opções de leitura de *datasets* oferecidas pela API de desenvolvimento do PyTorch, optou-se pela criação de um *dataset* a partir de diretórios de imagens. Assim, foram criados dois diretórios, denominados *tactilesurface* (piso-tátil) e *sidewalk* (calçada), e compartilhados com o ambiente de desenvolvimento Google Colab. Cada diretório contém imagens com as dimensões 100x100 pixels, que representam a classe definida pelo seu próprio nome. Foi criado também um segundo conjunto de dados, intitulado *testset*, com a mesma estrutura do primeiro e com a finalidade de validar o treinamento dos modelos.

Os dataset e *testset* da classe *tactilesurface* apresentam, respectivamente, 150 e 50 imagens cada. Para a classe *sidewalk* foram coletadas 50 imagens para o

dataset e 100 imagens para o *testset*.

Como forma de adaptar este modelo às dimensões dos datasets (100x100 pixels), foram adicionadas duas camadas extras à CNN. Uma camada convolucional e uma subsequente de agrupamento. A inserção de novas camadas e alteração das dimensões de entrada estabeleceram a necessidade de revisão das dimensões espaciais das imagens durante o processamento em cada camada. Logo, aplicou-se a Equação (2.1) para atualização das dimensões recebidas por cada camada, onde  $X_{in} = 100$ ,  $F = 5$ ,  $P = 0$  e  $S = 1$  para todas as camadas convolucionais. Em camadas de agrupamento os valores de  $X_{in}$  são determinados pela camada anterior  $F = 2$ ,  $P = 0$  e  $S = 2$ .

Após cada camada convolucional também estão presentes uma função de ativação ReLU e uma camada de normalização de batch, que realizam o tratamento dos dados antes de enviá-los à próxima camada da CNN.

Para as funções de custo e otimização foram aplicadas as funções *cross-entropy*, da biblioteca *torch.nn*, e a função Adam, da biblioteca *torch.optim*, devido a constatação da utilização das mesmas em trabalhos relacionados.

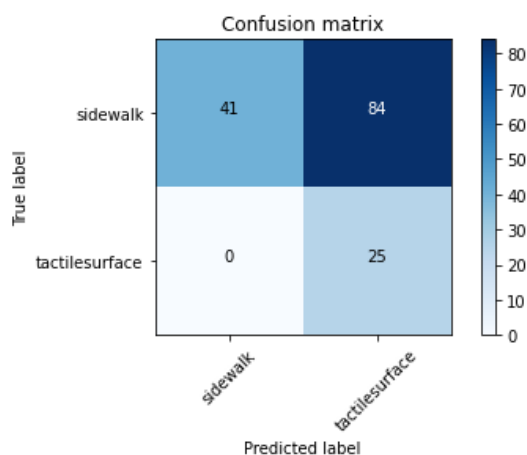
42

#### **4 AVALIAÇÃO DO MODELO**

A primeira avaliação foi realizada na CNN adaptada a partir do modelo LeNet-5. Para este modelo os resultados apresentaram um alto índice de variabilidade. Após a constatação do modelo, as funções de ativação da CNN foram substituídas pela função da Tangente Hiperbólica, original da arquitetura LeNet-5. O acompanhamento da função de custo no decorrer das épocas aponta rendimentos inferiores em comparação ao modelo adaptado, com valores praticamente constantes durante o processo de treinamento. Entretanto a matriz de confusão da arquitetura LeNet-5, Figura 5, expõe um comportamento mais assertivo que o primeiro modelo. O resultado percebido para o modelo ainda assim não é satisfatório, alcançando um nível de confiança de apenas 43% de assertividade.



**Figura 5 -** Matriz de confusão.



**Fonte:** Figura do autor.

## 5 CONCLUSÃO

No presente estudo foram tratados os principais conceitos de ANN e de CNN, essenciais para o desenvolvimento e compreensão de redes neurais genéricas. Também são tratadas as particularidades da elaboração de uma CNN para reconhecimento e classificação de imagens, contemplando os passos de criação de dataset, estruturação da CNN, aprendizado supervisionado e avaliação do modelo.

43

Embora os resultados finais não sejam satisfatórios, pôde-se identificar os maiores ofensores do projeto, com base nos conhecimentos adquiridos ao longo do estudo. Logo, a consolidação dos conhecimentos apresentada neste estudo pode servir de material de apoio à trabalhos que utilizem a mesma tecnologia, independente da finalidade proposta.

Também verificou-se a relevância da empregabilidade de CNN em problemas de classificação e reconhecimento de imagens, através de vários estudos em que essa tecnologia demonstra superioridade em relação a outras ferramentas para resolução dos mesmos problemas. Entretanto, assim como relatado em estudos e comprovado nas experimentações deste trabalho, a qualidade dos dados de entrada e o treinamento impactam fortemente no sucesso do modelo e devem ser as etapas de maior dedicação do projeto.

Para trabalhos futuros, é imprescindível a melhoria e aumento do conjunto de dados de treinamento. Alguns fatores podem ser levados em consideração no aperfeiçoamento desses dados, como a angulação dos elementos, boa resolução da

imagem, presença de apenas uma categoria de classificação por figura e especialização das categorias aprendidas. Ademais, outras estratégias de treinamento podem ser adotadas, com diferentes funções de ativação, número de camadas escondidas, hiperparâmetros, tamanhos de filtros, etc.

## REFERÊNCIAS

ABNT. NBR 16537 - Acessibilidade - Sinalização tátil no piso - Diretrizes para elaboração de projetos e instalação. Rio de Janeiro: ABNT, 2016.

ARAUJO, F. H. D. *et al.* Redes Neurais Convolucionais com Tensorflow: Teoria e Prática. [S.l.]: II Escola Regional de Informática do Piauí, 2017. v. 1.

DUBEY, S. R.; SINGH, S. K.; CHAUDHURI, B. B. A Comprehensive Survey and Performance Analysis of Activation Functions in Deep Learning. 2021.

FALCAO, J. V. R. *et al.* Redes neurais deep learning com tensorflow. Revista Eletrônica Científica de Ciência da Computação, v. 14, n. 1, 2019.

FREUND, Y.; SCHAPIRE, R. Large margin classification using the perceptron algorithm. Machine Learning, Kluwer Academic Publishers, v. 37, p. 277–296, 1999.

GALLANT, S. I. Perceptron-based learning algorithms. IEEE Transactions On Neural Networks, v. 1, n. 2, p. 179–191, 1990.

GORJANI, O. M. *et al.* Human activity classification using multilayer perceptron. Sensors, MDPI, v. 21, 2021.

GROSSI, E.; BUSCEMA, M. Introduction to artificial neural networks. European Journal of Gastroenterology and Hepatology, Research Gate, 2008.

GUERREIRO, J. *et al.* Airport accessibility and navigation assistance for people with visual impairments. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19. [S.l.]: ACM Press, 2019.

GÜNTHER, F.; FRITSCH, S. Neuralnet: Training of neural networks. The R Journal, v. 2/1, 2010.

HASHEMI, M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. Journal of Big Data, 2019.

HOSSAIN, M. A.; SAJIB, M. Classification of image using convolutional neural network (cnn). Global Journal of Computer Science and Technology, v. 19, p. 13–18, 05 2019.

HUSSEIN, M. E.; ABDALMAGEED, W. Introducing the DOME Activation Functions. 2021.

KANAL, L. N. Perceptron. Encyclopedia of Computer Science, p. 1383–1385, 2003.

KEREN, G. A Comparison of Loss Functions in Deep Embedding. Tese (Doutorado), 06 2017.

LI, Y. *et al.* Learning rate optimization in convolutional neural networks for medical images classification. [S.l.: s.n.], 2020.

MAEDA-GUTIÉRREZ, V. *et al.* Comparison of convolutional neural network architectures for classification of tomato plant diseases. Applied Sciences, MDPI, 2019.

MOHAMED, H.; ZAHRAN, M.; SAAVEDRA, O. Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: Case study el burullus lake. [S.l.: s.n.], 2015.

MURPHY, J. An Overview of Convolutional Neural Network Architectures for Deep Learning. [S.l.], 2016.

NGUYEN, A. *et al.* Explaining Convolutional Neural Networks by Tagging Filters. 2021.

45

NORIEGA, L. Multilayer perceptron tutorial. v. 2/1, 2005.

RAMCHOUN, H. *et al.* Multilayer perceptron: Architecture optimization and training with mixed activation functions. In: Proceedings of the 2nd International Conference on Big Data, Cloud and Applications. New York, NY, USA: Association for Computing Machinery, 2017. (BDCA'17). ISBN 9781450348522. Disponível em: <https://doi.org/10.1145/3090354.3090427>.

RODRIGUES, L. F. **Comparação entre redes neurais convolucionais e técnicas de pré-processamento para classificar células HEP-2 em imagens de imunofluorescência.** Dissertação (Mestrado) - Universidade Federal de Viçosa, 2018

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, Cornell Aeronautical Laboratory, v. 65, n. 6, 1958.

SAKIB, S. *et al.* An overview of convolutional neural network: Its architecture and applications. [S.l.: s.n.], 2018.

SHARMA, S. The Fundamentals of Neural Networks. 2017. Disponível em: <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>.

TEACHEY, A.; KIPPING, D. Identifying potential exomoon signals with convolutional neural networks. Monthly Notices of the Royal Astronomical Society, Oxford University Press (OUP), Sep 2021. Disponível em: <http://dx.doi.org/10.1093/mnras/stab2694>

YAN, K. *et al.* CNN vs. SIFT for image retrieval. In: Proceedings of the 2016 ACM on Multimedia Conference - MM '16. [S.l.]: ACM Press, 2016.