



EFICIÊNCIA DE REDES NEURAIS CONVOLUCIONAIS NA IDENTIFICAÇÃO DE EDIFÍCIOS EM CONDIÇÕES DESAFIADORAS DE IMAGENS

Acadêmico do Curso de Engenharia de Computação: Caio Emanuel Pontes Sousa
Acadêmico do Curso de Engenharia de Computação: Igor Arruda Freitas Silva
Acadêmico do Curso de Engenharia de Computação: João Gilberto dos Santos
Docente Orientador do Curso de Engenharia de Computação: Marcelo Lucas

Universidade de Uberaba - Uniube, Uberaba – MG, Brasil
E-mail do autor correspondente: caiopontes@edu.uniube.br

RESUMO

Esta pesquisa concentra-se na eficácia das redes neurais convolucionais (CNN – Convolutional Neural Network) na identificação de edifícios em imagens desafiadoras. O objetivo é comparar e melhorar o desempenho do modelo com o uso de CNNs. A metodologia incluiu o pré-processamento de imagens, aplicação de camadas convolucionais densas com funções de ativação denominadas unidades lineares retificadas (ReLU - Rectified Linear Unit), **utilizar o Softmax como classificador e por fim realizar o treinamento com validação cruzada**. Os resultados indicaram que o **uso das redes neurais convolucionais** pode aprimorar significativamente a capacidade de redes neurais em lidar com variações de ângulo, condições de iluminação e obstruções nas imagens. As CNNs são ferramentas eficazes no reconhecimento de padrões visuais em construções, destacando seu potencial para aprimorar práticas de georreferenciamento e outras aplicações.

Palavras-chave: Redes Neurais Convolucionais. Reconhecimento de Imagens. Deep Learning. Inteligência Artificial.

ABSTRACT

This research focuses on the effectiveness of CNNs in identifying buildings in challenging images. The objective is to compare and improve model performance using CNNs. The methodology included image preprocessing, application of convolutional and dense layers with ReLU and Softmax activation functions, and training with cross-validation. The results indicated that the integration of algorithms can significantly enhance the ability of neural networks to handle variations in angle, lighting conditions, and obstructions in images. CNNs are effective tools in recognizing visual patterns in buildings, highlighting their potential to improve georeferencing practices and other applications.

Keywords: Convolutional Neural Networks. Image Recognition. Deep Learning. Artificial Intelligence.

1. INTRODUÇÃO

No estudo realizado por Silva (2023), foram aplicados métodos avançados de deep learning, como autoencoders empilhados e o classificador Softmax no reconhecimento de imagens de edifícios em condições adversas, incluindo: variações de ângulos, diferentes condições de iluminação e obstruções. Silva e os demais autores deste trabalho obtiveram uma taxa de acerto média de 94,4% nos testes de reconhecimento, evidenciando como a integração de algoritmos pode aprimorar a capacidade de realização de tarefas complexas. O trabalho desenvolvido por eles é uma demonstração clara do potencial da inteligência artificial na área de georreferenciamento, oferecendo percepções importantes sobre a aplicação de técnicas de *deep learning* no reconhecimento de padrões visuais em construções. Este trabalho não só mostra o impacto da inteligência artificial nas práticas tradicionais, mas também destaca o rápido avanço da tecnologia.

Assim, diferentemente do trabalho realizado por Silva (2023), o objetivo deste estudo é usar as redes neurais convolucionais (CNNs) e comparar o seu desempenho com aqueles encontrados no trabalho de Silva (2023) na identificação de edifícios.

2 REFERENCIAL TEÓRICO

Segundo Luger (2004), "As redes neurais convolucionais têm demonstrado ser altamente eficazes em tarefas de reconhecimento de imagens, devido à sua capacidade de detectar características hierárquicas." (p. 422). As redes neurais têm alto poder de reconhecimento, pois detectam bem as principais características das

imagens e, com a ajuda de pré-processamento, ela se torna ainda mais eficiente atingindo elevadas taxas de acerto. As CNN têm sido utilizadas na busca de padrões de imagens graças à sua capacidade de armazenar características importantes de forma automática.

De acordo com Caitano e Araújo (2023), as CNNs são eficazes em tarefas de reconhecimento facial, onde a taxa de acerto e eficiência tem maior relevância. Eles mencionam que uma CNNs foi desenvolvida usando a biblioteca ml5.js [ml5.js](#) ([Friendly Machine Learning for the Web](#), 2025) com um resultado de 72,37% em imagens não utilizadas no treinamento.

"A combinação de grandes conjuntos de dados, poder computacional e algoritmos avançados têm levado a avanços significativos na área"(Carvalho, 2011, p. 138). Ao utilizar grandes volumes de dados para treinamento juntamente com um bom modelo de pré-processamento de imagens e uma rede neural eficiente, conseguimos obter ótimos resultados de percentual de acerto utilizando principalmente as redes CNNs. Segundo Goodfellow (2016), as CNNs são uma classe de redes neurais especialmente eficazes para o processamento imagens. A aplicação de convoluções permite capturar características espaciais e padrões locais. Sendo assim, a escolha das CNNs neste trabalho deve-se à grande quantidade de dados presente nas imagens analisadas.

Conforme relatado por Silva (2023), o reconhecimento de edifícios é um desafio na área de inteligência artificial, devido às condições adversas das imagens usadas. Assim, Silva e os demais autores, utilizaram para resolver o problema das imagens em condições adversas, os autoencoders empilhados para a remoção e redução de dimensionalidade das características das imagens, mantendo as características mais importantes para a classificação, o que é muito útil para imagens de baixa resolução.

O Softmax foi usado para o reconhecimento das imagens, e seu desempenho foi comparado com o de redes perceptron multicamadas (MLP). Os resultados mostraram uma taxa de acerto de 94,9% na classificação das imagens de edifícios, destacando a eficiência do método proposto.

Bademian (2023) utilizou o dataset German Traffic Sign (https://benchmark.ini.rub.de/gtsrb_dataset.html, 2025) para treinar uma CNN com o objetivo de identificar e classificar sinais de trânsito. A arquitetura inclui camadas de convolução, *pooling* e *dropout* para evitar o *overfitting*. Todo esse processo busca selecionar as principais características da imagem antes de ser inserida no modelo para que se obtenha um melhor desempenho.

O *overfitting* pode ser interpretado como o caso em que o modelo aprende de mais com os dados usados na fase de treinamento, ou seja, ele se comporta como se tivesse "decorado" as respostas dos dados disponibilizados. Assim, ao ser testar com um exemplo não usado na fase de treinamento, o sistema pode não conseguir identificar devidamente, pois esse não foi um dos dados que o modelo "decorou" (Bademian, 2023, p. 9). Esta situação é agravada pelo superdimensionamento da rede neural.

Dropout é uma técnica que consiste em desativar aleatoriamente uma porcentagem dos neurônios durante o treinamento. Essa técnica reduz a complexidade do modelo e melhora sua capacidade de generalização.

É notável que a inclusão do processo de dropout, após cada uma das camadas de Pooling do modelo, fez com que a taxa de acerto do processo aumentasse (Bademian, 2023, p. 6). Essa técnica tem como objetivo a melhoria da porcentagem de acertos do modelo e a redução das chances de overfitting.

A escolha da função de ativação dos neurônios é um ponto importante no desempenho de uma rede neural (Rizzo, 2020). A função Relu, $f(x) = \max(0, x)$, quando utilizada no reconhecimento de imagens, permite enfatizar as suas características.

Por outro lado, SoftMax é uma função matemática (1) utilizada para modelos de aprendizado de máquina, por meio da conversão de valores reais probabilidades (Goodfellow, 2016).

Equação 01 – Função softmax

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (1)$$

Fonte: Autores

em que:

x: valores reais.

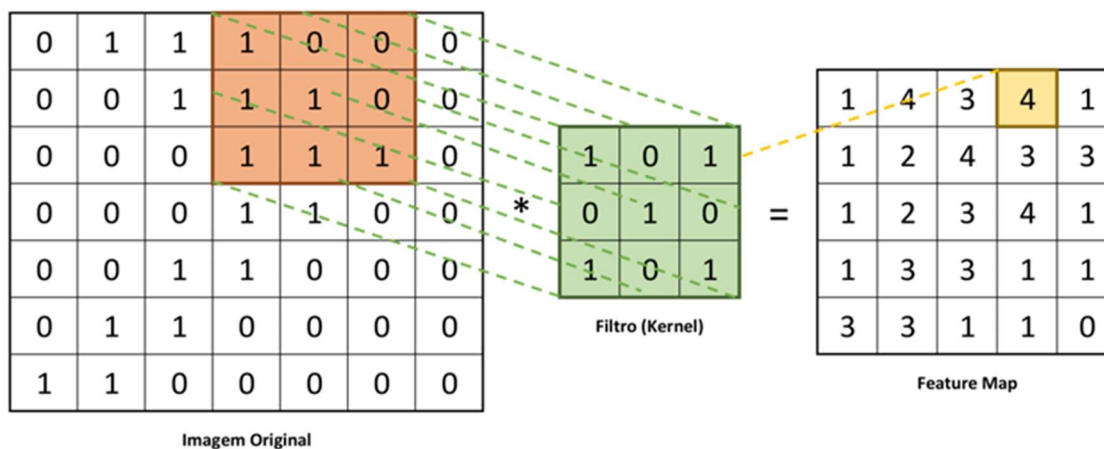
N: quantidade de elementos.

s: probabilidade de ocorrência.

Outro recurso importante que é utilizado neste trabalho é filtro conhecido como kernel que se aplica no **reconhecimento de imagens**, especialmente em **redes neurais convolucionais (CNNs)**. Trata-se de uma pequena matriz de números que varre a imagem para extrair características importantes, como bordas, texturas ou padrões. A figura 1 ilustra a aplicação do kernel a uma imagem digitalizada.

Figura 1 - Cálculo da Camada Convolucional (Granatyr, 2023)

Camada Convolucional



$$(1 * 1) + (0 * 0) + (0 * 1) + (1 * 0) + (1 * 1) + (0 * 0) + (1 * 1) + (1 * 0) + (1 * 1) = 4$$

Fonte: Granatyr; Jones

Além dos recursos ilustrados nesta seção, apresentam-se a seguir os detalhes da metodologia empregada no reconhecimento de edifícios.

3 METODOLOGIA

A abordagem proposta para este trabalho é a utilização de uma Rede Neural Convolucional (CNN) para o reconhecimento de edifícios. Toda a estrutura da rede neural foi desenvolvida em Python utilizando o ambiente de desenvolvimento do Spyder. As bibliotecas do Keras API foram empregadas para construir as camadas de convolução, camadas densas, normalizadores, e para a geração de imagens adicionais para complementar a base de dados. Além disso, o Scikit Learn (Scikit Learn, 2025) foi utilizado para validar o modelo através do método de validação cruzada, e o Numpy (Harrys, 2025) para os cálculos da taxa de acerto.

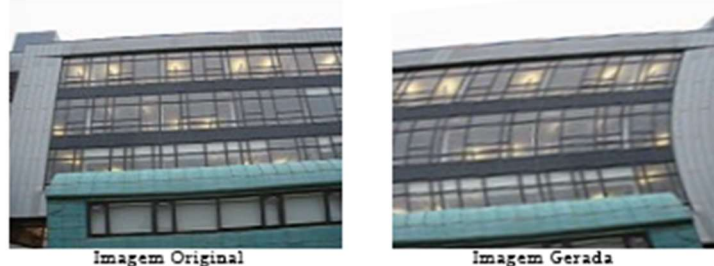
3.1 BASE DE DADOS

A base de dados utilizada SBID (Li, 2009) é composta por imagens de edifícios da Universidade de Sheffield. Ao todo são 3192 imagens em formato JPEG, com resolução de 160x120 pixels, além disso foram geradas 168 imagens adicionais utilizando a ferramenta Image Data Generator da biblioteca do Keras (Chollet, 2015). As imagens geradas são baseadas na base de dados original e sofrem aleatoriamente diferentes mutações, tais como:

- Rotação aleatória em até 7 graus para esquerda ou para direita;
- Espelhamentos horizontais aleatórios;
- Cisalhamentos e/ou distorções de até 20%;
- Deslocamentos verticais em até 7% da altura total da imagem;
- Ampliações aleatórias em até 20%;

A Figura 2 mostra um exemplo destas novas imagens criadas.

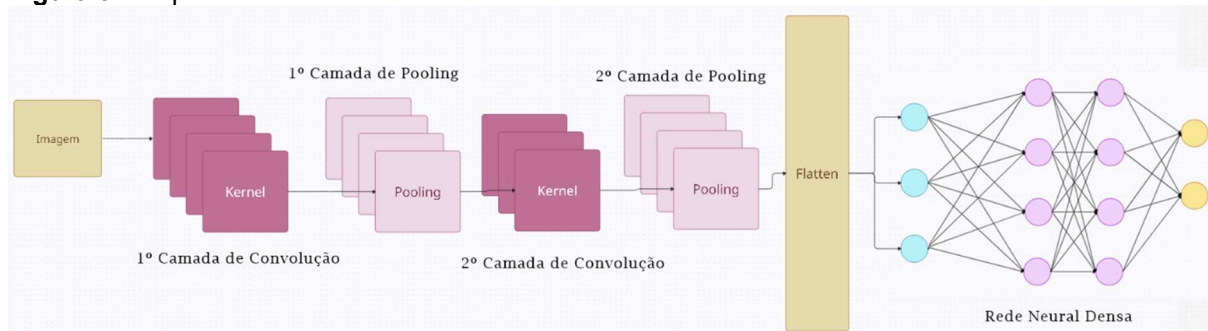
Figura 2 - Comparação de imagens geradas.



Fonte: Autores.

A arquitetura da rede neural, apresentada na Figura 3, é dividida em duas etapas principais: o pré-processamento das imagens e a rede neural densa. A primeira etapa, o pré-processamento, inclui a camada de convolução, cujo objetivo é extrair as características principais da imagem e reduzir a dimensionalidade, preparando-as para a próxima etapa.

Figura 3 - Arquitetura da rede neural convolucional.



Fonte: Autores

3.2 PRÉ-PROCESSAMENTO

Durante o pré-processamento, as imagens são inseridas nos neurônios da camada de entrada da rede convolucional. Esta etapa é fundamental para extrair os padrões e as características essenciais das imagens, facilitando o trabalho das camadas subsequentes. A camada de convolução é subdividida em quatro partes descritas a seguir.

3.2.1 OPERADOR DE CONVOLUÇÃO

A primeira parte trata-se do operador de convolução, utilizado para extrair características de dados estruturados em imagens. Nesse processo, um kernel é aplicado sobre a imagem. Cada pixel da imagem é alterado ponderadamente pelo kernel. Após a aplicação da convolução, a imagem é reduzida em uma matriz menor chamada de mapa de características, contendo apenas os dados considerados importantes e mais relevantes da imagem, é interessante salientar que vários mapas de características são gerados, facilitando assim a identificação de padrões. Identificar qual é o melhor kernel faz parte do treinamento da rede neural.

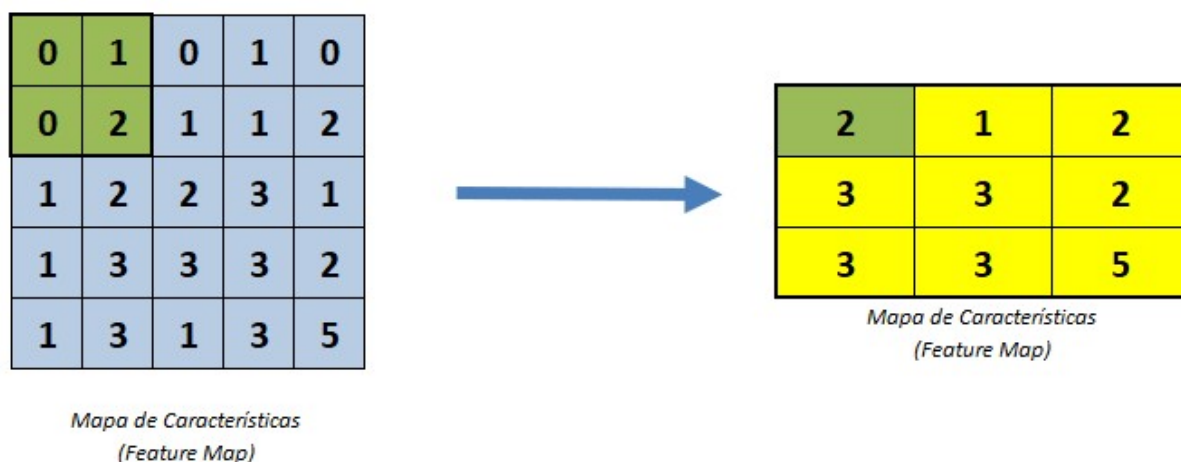
No modelo proposto, a primeira camada de convolução é composta por 64 aplicações de kernel, enquanto a segunda camada é composta por 32, ambos no tamanho 3x3. A função de ativação ReLU é utilizada em ambas as camadas.

Antes de partir para o próximo passo, o mapa de características passa por um processo chamado normalização que consiste em padronizar todos os dados em uma escala entre 0 e 1, aprimorando o processamento da rede neural.

3.2.2 POOLING

O Pooling ou Max Pooling tem como função dividir o mapa de características em blocos menores e de tamanho fixo, normalmente chamados de janela. Para cada iteração a operação de Max Pooling seleciona o valor máximo dos pixels dentro dessa janela. Este valor máximo é utilizado para representar toda a região na matriz de saída do pooling. Em seguida a janela continua se movendo por todo o mapa de características. O pooling é utilizado para enfatizar as características mais importantes da imagem, além de influenciar diretamente na redução de overfitting conforme mostra a Figura 4.

Figura 4 – Aplicação do Pooling.



Fonte: Autores.

A janela do pooling utilizada no modelo tem o tamanho 4x4, e se move em apenas um pixel a cada iteração.

3.2.3 FLATTENING

Após realizar todos os três passos anteriores nas duas camadas de convolução, o último passo do pré-processamento é chamado de Flattening, que basicamente transforma a matriz multidimensional em um vetor unidimensional. Este passo é necessário para preparar os dados para serem inseridos na camada densa.

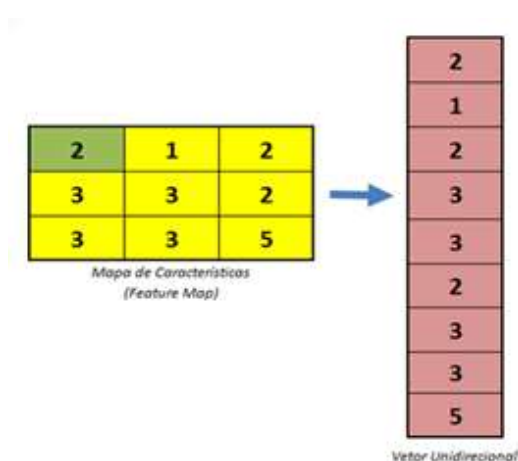


Figura 5 - Transformação do Mapa em um Vetor Unidirecional

3.3 REDE NEURAL DENSA

Nesta etapa, após todo o pré-processamento, os dados são recebidos na camada de entrada de uma rede neural do tipo feedforward. A configuração utilizada

em nosso modelo é composta por duas camadas escondidas com 350 neurônios pois após passar pelo pré-processamento as imagens têm um tamanho estimado de 14x14, ou seja, 196 pixels. Nos testes verificou-se que, adotando-se o número de neurônios das camadas ocultas como o dobro do número de pixels, obtêm-se o melhor desempenho. Isso resultaria 392 neurônios, porém, por tentativa, verificou-se que o uso de 350 neurônios, neste caso, foi a melhor opção. A camada de saída foi composta por 40 neurônios, representando as 40 categorias de edifícios.

Em ambas as camadas escondidas utilizamos a função de ativação ReLU e, na camada de saída, a função utilizada foi a Softmax, para realizar a classificação multiclasse. Um Dropout, responsável por desligar de forma aleatória 20% dos neurônios, foi utilizado nas camadas escondidas com intuito de prevenir o overfitting.

Este modelo faz uso da função de perda denominada entropia cruzada categórica, comumente utilizada em problemas de classificação multiclasse. Ela mede a diferença entre a distribuição de probabilidades prevista pelo modelo e a distribuição real dos dados.

Para uma única camada, a perda é calculada da seguinte forma:

Equação 2 - Fórmula da rede neural de uma camada

$$h = f(Wx - b)$$

Fonte: Autores

Onde:

x é o vetor de entrada;

W é a matriz de pesos que conecta a camada de entrada à camada de saída;

b é o vetor de bias;

f é a função de ativação; e

h é o vetor de saída da camada.

Equação 3 – Fórmula da rede neural com múltiplas amostras

$$h^l = f^l(W^l h^{l-1} - b^l)$$

Fonte: Autores

Para um conjunto de dados com M amostras, a perda é a média das perdas individuais

h^0 é a entrada

W^l é a matriz de pesos da camada l

b^l é o vetor de bias da camada l

f^l é a função de ativação da camada l

h^l é a saída da camada l , que serve como entrada a camada $l + 1$.

3.4 TREINAMENTO

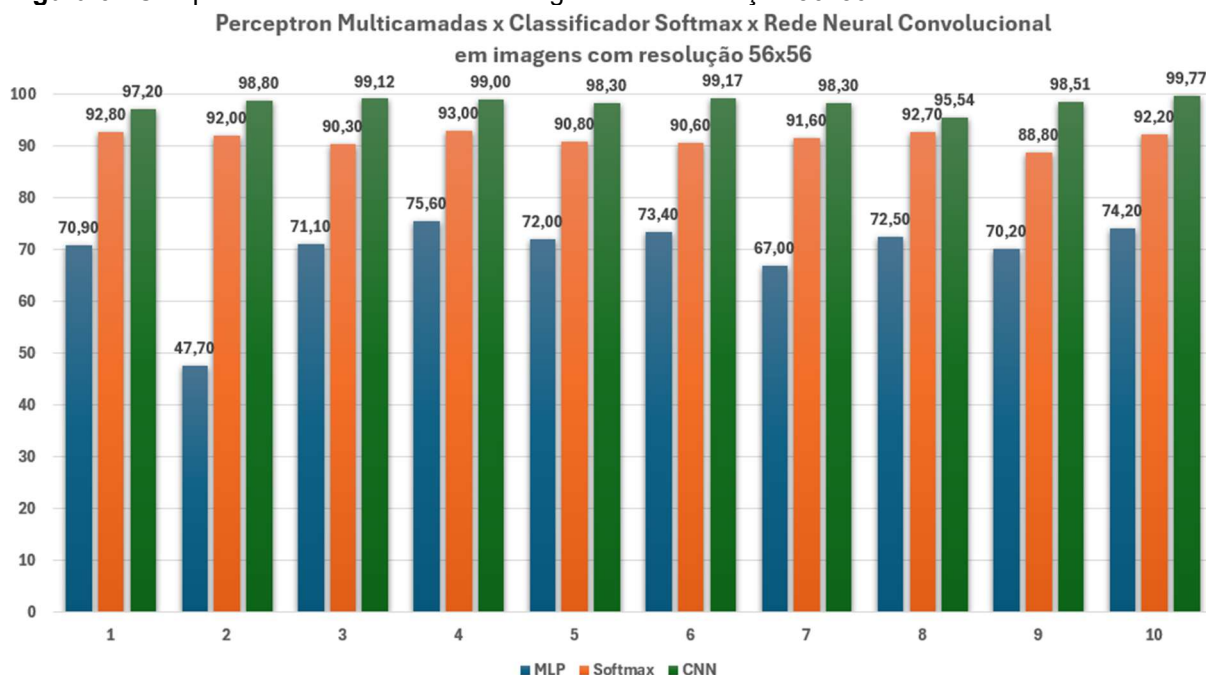
A etapa de treinamento foi realizada em 80 épocas, com um conjunto de 32 amostras. Para validar o modelo foi realizada a validação cruzada, que é uma técnica que divide o conjunto de dados em k partes de tamanhos iguais. A cada iteração, uma das partes é utilizada como um conjunto de teste enquanto as outras são usadas no treinamento. Sendo assim, o modelo é treinado k vezes e, no final, a média de todas as performances é utilizada para fornecer uma estimativa da taxa de acerto do modelo.

4 RESULTADOS E DISCUSSÃO

Para este estudo, a base de dados foi dividida em 10 partes, para diferentes resoluções de imagens: 14 x 14, 28 x 28 e 56 x 56 pixels, além de mais uma rodada de testes mantendo a resolução original das imagens, 160 x 120 pixels. Os testes foram realizados em um ambiente computacional com as seguintes especificações:

- Processador Intel Core i7 11700K, 11ª Geração, 3.6 GHz;
- 32GB de Memória RAM;
- Placa de Vídeo: GTX-1050ti com 4 GB de VRAM;
- SSD M2 de 500gb com leitura de 3500MB/s.

No primeiro teste, a rede neural foi configurada conforme descrito na metodologia e foram utilizadas as imagens na resolução de 56 x 56. Os resultados obtidos foram comparados com o estudo realizado por Silva (2023), como mostrado na Figura 6.

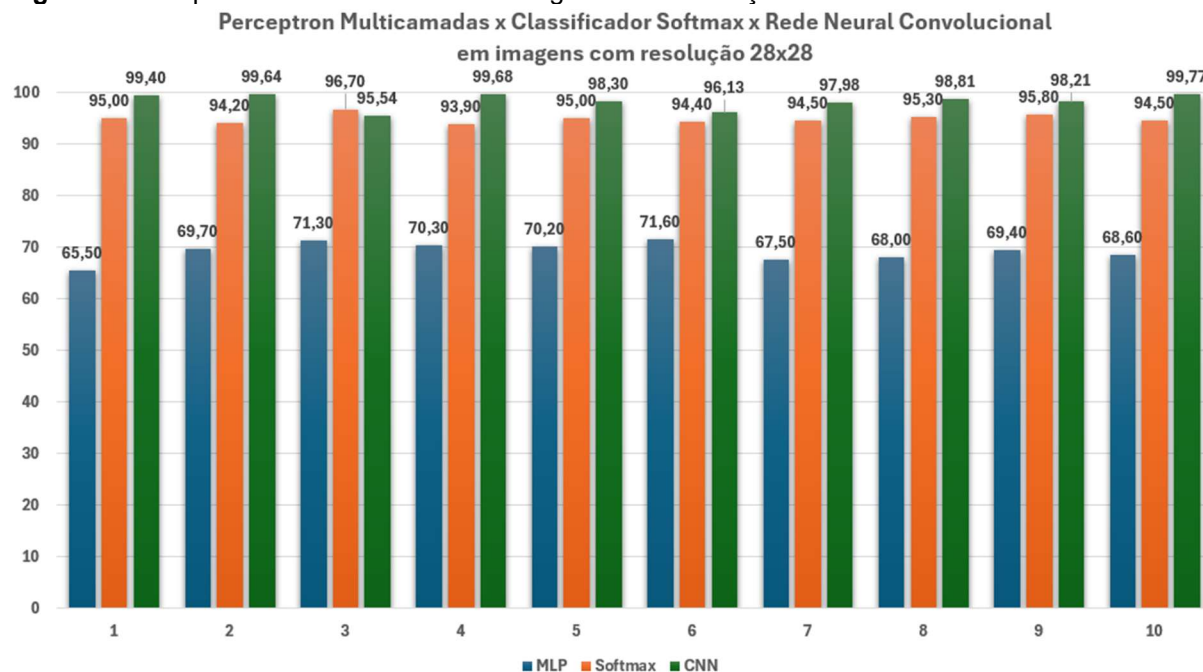
Figura 6 - Comparativo de resultados em imagens com resolução 56x56.

Fonte: Autores

Com a análise da Figura 6, é notável que o Perceptron Multicamadas (MLP) apresentou uma taxa de acerto que varia de 47,7% e 75,6% com uma média de 69,46%. Esses resultados indicam que o MLP tem dificuldades em lidar com a complexidade das imagens dos edifícios. O MLP também apresentou uma variação significativa entre as amostras indicando uma baixa consistência. O classificador Softmax apresentou uma taxa de acerto mais estável, variando entre 88,8% e 93% com uma eficácia média de 91,48%, este método mostrou-se mais robusto do que o MLP, e a sua estabilidade sugere que ele é capaz de generalizar bem os padrões nas imagens. No modelo proposto por este estudo, a rede neural convolucional (CNN) apresentou os melhores resultados com uma taxa de acerto média de 98,37%, variando entre 95,54% e 99,77%. Estes exemplos indicam a eficácia das CNNs para tarefas de reconhecimento de imagem.

Na Figura 7, temos o gráfico de comparação dos resultados obtidos em cada modelo para imagens com resolução 28 x 28

Figura 7 - Comparativo de resultados em imagens com resolução 28x28.

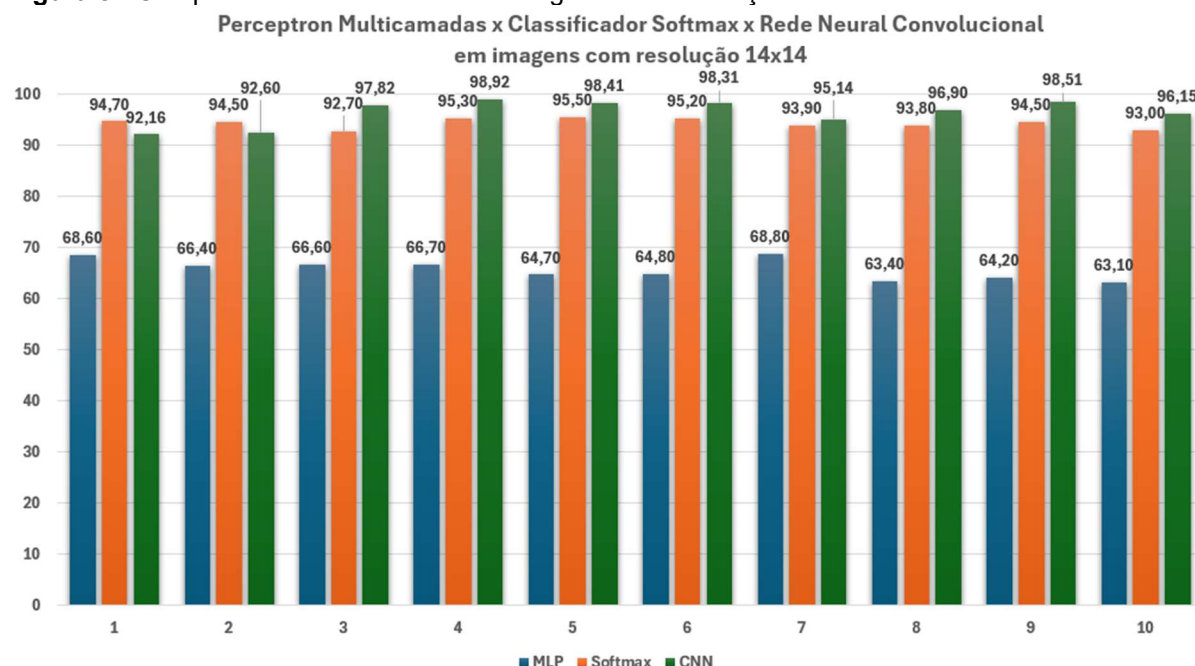


Fonte: Autores

De acordo com a Figura 7, o MLP apresentou uma taxa de acerto média de 69,21% que varia entre 65,5% e 71,6%, embora os resultados apresentados sejam ligeiramente superiores aos resultados anteriores utilizando uma maior resolução, principalmente em relação à estabilidade, ainda é notável sua limitação em classificar imagens complexas. O Softmax mostrou uma taxa de acerto média de 94,93%, variando entre 93,9% e 96,7%, apresentando mais uma vez um desempenho robusto e consistente, porém ainda inferior às CNNs, que obtiveram um desempenho médio de 98,34%, variando entre 95,54% e 99,77%, confirmando mais uma vez sua eficácia em tarefas de reconhecimento de imagens, devido principalmente à sua alta capacidade de aprender características complexas.

A Figura 8 mostra o gráfico que foi obtido a partir dos resultados de imagens com resolução de 14 x 14 pixels.

Figura 8 - Comparativo de resultados em imagens com resolução 14x14



Fonte: Autores

De acordo com a Figura 8, o Perceptron Multicamadas (MLP) apresentou mais uma vez, um desempenho inferior aos demais com uma média de 65,73% e uma boa estabilidade variando entre 63,1% e 68,8%. O classificador Softmax manteve sua robustez variando entre 92,7% e 95,5% com uma taxa de acerto média de 94,31%. A Rede Neural Convolucional (CNN) obteve novamente os melhores resultados, com uma taxa de acerto média de 96,49%, e uma variação entre 92,16% e 98,92%, a CNN mostrou-se inferior ao tentar reconhecer imagens com resoluções menores, comparando as mesmas imagens com resoluções melhores. Porém mesmo com esse desempenho um pouco abaixo do esperado, ela ainda se manteve superior aos outros modelos.

A tabela 1, mostra os resultados obtidos pelo trabalho de Silva et al. (2023) e os obtidos pela CNN proposta neste trabalho.

Tabela 1 – Média da taxa de acerto por resoluções

Resolução	MLP - Silva et al. (2023)	SoftMax - Silva et al. (2023)	CNN
14x14	65,73	94,31	96,49
28x28	69,21	94,93	98,34
56x56	69,46	91,48	98,37

Fonte: Autores.

Analisando a Tabela 1, verifica-se que mesmo para as imagens com resolução mais baixa, como 14 x 14, a rede CNN manteve a sua taxa de reconhecimento superior aos resultados encontrados por Silva et al. (2023). Assim, entende-se que a rede CNN se comportou muito bem na tarefa de reconhecimento de imagens para edifícios, onde as imagens estão em condições de diferentes ângulos, iluminação e obstruções parciais, sendo então mais uma alternativa para essa tarefa. Além disso, os resultados mostram que ela se comportou melhor em imagens com resoluções maiores, o que indica uma ótima capacidade de processamento para conjuntos com muitos dados.

5 CONCLUSÕES

O presente trabalho analisou a eficiência das Redes Neurais Convolucionais (CNNs) na identificação de edifícios sob diversas condições de imagens, comparando-as com outras técnicas de aprendizado de máquina, como o Perceptron Multicamadas (MLP) e o Softmax. Os resultados obtidos são de extrema importância, pois evidenciam a eficácia das CNNs em cenários de reconhecimento de imagens complexas, contribuindo para avanços em diversas aplicações práticas, desde segurança pública até automação industrial. A capacidade das CNNs de aprender características importantes de forma automática e a sua resistência a variações nas condições das imagens são características que as tornam ferramentas valiosas no campo da inteligência artificial. No entanto, é importante considerar algumas limitações e desafios identificados ao longo deste trabalho. A necessidade de grandes volumes de dados para o treinamento das CNNs e o alto custo computacional associado são aspectos que demandam atenção. Além disso, apesar dos resultados promissores, a variação de desempenho em diferentes resoluções sugere que há espaço para melhorias e otimizações adicionais. A investigação de métodos para reduzir o custo computacional e aumentar a eficiência das CNNs também é um campo promissor que pode trazer benefícios significativos para a área. Este estudo nos mostra a relevância das CNNs como uma solução eficaz para problemas de reconhecimento de imagens e abre caminho para novas pesquisas que busquem aprimorar e expandir suas aplicações em diversos setores.

6 REFERÊNCIAS

APPLE. SoftMax function. Disponível em: <https://developer.apple.com/documentation/accelerate/bnnsactivationfunction/2915301-softmax>. Acesso em: 16 jun. 2024.

BADABEMIAN, Gustavo Bulka Bonafé; FRANÇA, Celso Ap. de. **Classificação de placas de trânsito com redes neurais para automação de veículos**. São Carlos: Universidade Federal de São Carlos, Departamento de Engenharia Elétrica, 2023.

CAITANO, Daniel Almeida; ARAÚJO, Ikaro Campos de. **Reconhecimento facial utilizando redes neurais convolucionais para auxiliar na automatização de processos no Campus Santo Antônio de Jesus do IFBA**. 2023. Trabalho de



Conclusão de Curso (Curso de Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Campus Santo Antônio de Jesus, 2023. Disponível em: https://portal.ifba.edu.br/santoantonio/ensino/cursos-tecnologos/arquivos_ads/ifba-saj-ads-tcc-artigo-daniel-reconhecimento_facial.pdf/@download/file/IFBA%20SAJ%20-%20ADS%20-%20TCC%20Artigo%20Daniel%20Reconhecimento_Facial.pdf. Acesso em: 5 abr. 2024.

CARVALHO, André C. P. L. F. de. **Aprendizado de máquina**. São Paulo: Blucher, 2011. Disponível em: <https://www.blucher.com.br/livro/aprendizado-de-maquina-andre-carvalho-124>. Acesso em: 15 mar. 2024.

FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. 2. ed. Rio de Janeiro: Grupo GEN, 2021. Disponível em: <https://www.grupogen.com.br/e-book-inteligencia-artificial-uma-abordagem-de-aprendizado-de-maquina>. Acesso em: 4 abr. 2024.

GHARAT, S. **O quê, por que e qual ?? Funções de ativação**. 2019. Disponível em: <https://medium.com/@snaily16/what-why-and-which-activation-functions-b2bf748c0441>. Acesso em: 13 nov. 2023.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. Cambridge: MIT Press, 2016. Disponível em: <http://www.deeplearningbook.org/>. Acesso em: 12 mar. 2024.

GRANATYR, Jones. **Deep learning com Python de A a Z - o curso completo**. Udemy, 2023. Disponível em: <https://www.udemy.com/course/deep-learning-com-python-az-curso-completo/>.

IBM. **Deep learning**. 2023. Disponível em: <https://www.ibm.com/br-pt/topics/deep-learning>. Acesso em: 10 nov. 2023.

IA EXPERT ACADEMY. **Funções de ativação: definição, características e quando usar cada uma**. 2020. Disponível em: <https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/>. Acesso em: 4 nov. 2023.

LUGER, George F. **Inteligência artificial: estruturas e estratégias para a complexidade humana**. 6. ed. Porto Alegre: Bookman, 2004. Disponível em: <https://livrariapublica.com.br/livros/inteligencia-artificial-george-f-luger/>. Acesso em: 13 mar. 2024.

NIELSEN, Michael. **Neural networks and deep learning**. Determination Press, 2015. Disponível em: <http://neuralnetworksanddeeplearning.com/>. Acesso em: 13 mar. 2024.



RISZO, Igor Vilela; CANATO, Robson Leandro Carvalho. Inteligência artificial: funções de ativação. **Revista Prospectus**, Itapira: FATEC de Itapira "Ogari de Castro Pacheco", v. 2, n. 2, p. 51-65, ago. 2020/fev. 2021. Acesso em: 10 mar. 2024.

SILVA, Leandro Aureliano da et al. Recognition of low-resolution building images using deep learning and softmax. **Revista Observatorio de la Economía Latinoamericana**, Curitiba, v. 21, n. 9, p. 14288-14308, 2023. DOI: 10.55905/oelv21n9-2023.

TERMINAL ROOT. **As 30 melhores bibliotecas e pacotes Python para iniciantes**. 2019. Disponível em: <https://terminalroot.com.br/2019/12/as-30-melhores-bibliotecas-e-pacotes-python-para-iniciantes.html>. Acesso em: 10 nov. 2023.

VICERI. **Arquiteturas de redes neurais convolucionais para reconhecimento de imagens**. Disponível em: <https://viceri.com.br/insights/arquiteturas-de-redes-neurais-convolucionais-para-reconhecimento-de-imagens/>. Acesso em: 4 nov. 2023.

ML5.JS. **Friendly machine learning for the web**. Disponível em: <https://ml5js.org/>. Acesso em: 19 abr. 2025.

SCIKIT-LEARN. Documentation. Disponível em: <https://scikit-learn.org/stable/documentation.html>. Acesso em: 1 maio 2025.

NUMPY. Documentation. Disponível em: <https://numpy.org/doc/stable/>. Acesso em: 1 maio 2025.

HARRIS, C. R. et al. **Array programming with NumPy**. **Nature**, v. 585, n. 7825, p. 357–362, 2020. DOI: <https://doi.org/10.1038/s41586-020-2649-2>.

LI, J.; ALLINSON, N. M. Dimensionality reduction-based building recognition. In: **Proceedings of the Ninth IASTED International Conference on Visualization, Imaging and Image Processing**, Cambridge, UK, 13–15 jul. 2009.

CHOLLET, F. **Keras: the Python deep learning library**. 2015. Disponível em: <https://keras.io>. Acesso em: [informe a data de acesso, se necessário].

GERMAN TRAFFIC SIGN BENCHMARKS. Disponível em: https://benchmark.ini.rub.de/gtsrb_dataset.html. Acesso em: 8 maio 2025.