

16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

IMPLEMENTAÇÃO DE REDES NEURAIS CONVOLUCIONAIS PARA PROCESSAMENTO DE IMAGENS EM TEMPO REAL E CONTROLE DE ROBÔ SEGUIDOR DE LINHA

Paulo C. Da Silva Filho¹, Ricardo F. Nunes²

1 Graduando em Bacharelado em Engenharia Elétrica, Bolsista PIBIFSP, IFSP, Campus Presidente Epitácio, p.clementino@aluno.ifsp.edu.br.

2 Docente em Engenharia Elétrica, IFSP, Campus Presidente Epitácio
Área de conhecimento (Tabela CNPq): 3.04.05.00-9 Engenharia Elétrica

RESUMO: O presente trabalho investiga a aplicação de Robôs Móveis Autônomos (AMRs) no contexto de seguidores de linha, buscando aprimorar sua percepção e tomada de decisão por meio de Redes Neurais Convolucionais (CNNs). Historicamente, robôs seguidores de linha dependem de sensores simples; este estudo propõe a integração de visão computacional, alavancada por CNNs, para conferir maior adaptabilidade a ambientes com variações visuais. A metodologia envolveu a construção de um protótipo equipado com ESP32-CAM para captura de imagens e o desenvolvimento de uma arquitetura CNN. O trabalho em andamento validou o modelo de classificação de imagens por meio da matriz de confusão, taxa de acurácia e testes de desempenho utilizando a ESP32-CAM com transmissão ao vivo (FPS). O foco agora é a implementação da inteligência artificial no sistema embarcado. Apesar da promissora acurácia do modelo, o sucesso final dependerá da superação dos desafios técnicos de latência no processamento em tempo real, um requisito fundamental para a concretização do sistema de controle autônomo proposto.

PALAVRAS-CHAVE: Redes Neurais Convolucionais, Robótica Embarcada, Visão Computacional.

IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORKS FOR REAL-TIME IMAGE PROCESSING AND LINE-FOLLOWER ROBOT CONTROL

ABSTRACT: The current work investigates the application of Autonomous Mobile Robots (AMRs) in the context of line followers, seeking to enhance their perception and decision-making through Convolutional Neural Networks (CNNs). Historically, line follower robots rely on simple sensors; this study proposes the integration of computer vision, leveraged by CNNs, to provide greater adaptability to environments with visual variations. The methodology involved the construction of a prototype equipped with an ESP32-CAM for image capture and the development of CNN architecture. The work in progress validated the image classification model by means of a confusion matrix, accuracy rate, and performance tests using the ESP32-CAM with live streaming (FPS). The focus now is on implementing artificial intelligence within the embedded system. Despite the model's promising accuracy, final success will depend on overcoming the technical challenges of latency in real-time processing, a fundamental requirement for the realization of the proposed autonomous control system.

KEYWORDS: Convolutional Neural Networks, Embedded Robotics, Computer Vision.

INTRODUÇÃO

Os robôs móveis autônomos (AMRs, do inglês *Autonomous Mobile Robots*) realizam tarefas complexas sem intervenção humana direta e são capazes de se deslocar utilizando sensores, atuadores e sistemas de processamento para perceber o ambiente, tomar decisões e executar movimentos. Historicamente, seu desenvolvimento enfrentou desafios como evasão de obstáculos, limitações de hardware e processamento em tempo real, o que dificultava a obtenção de respostas rápidas e precisas (Chang et al., 2019; Rubio, Valero e Llopis-Albert, 2019).

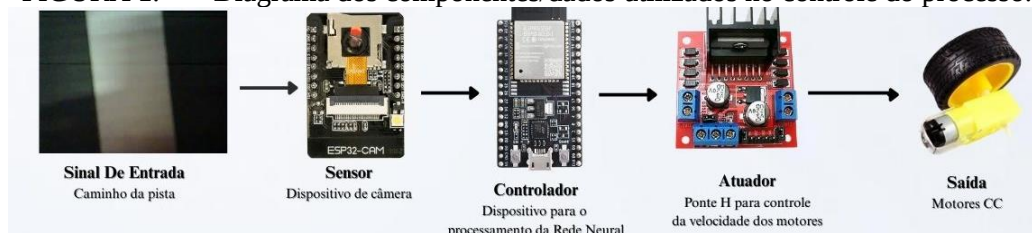
Atualmente, a adoção de AMRs cresce devido à busca por eficiência, redução de custos e otimização de processos, especialmente em ambientes industriais e logísticos. Avanços em visão computacional e inteligência artificial, com redes neurais convolucionais (CNNs) potencializadas por GPUs e *big data*, têm ampliado suas capacidades (O'shea e Nash, 2015; Ušinskis et al., 2025). Apesar disso, prover visão semelhante à humana ainda é desafiador devido ao alto custo computacional, sendo as CNNs essenciais para extrair automaticamente características visuais (Browne, Ghidary e Mayer, 2007; Lawrence et al., 1997).

Com base nesse potencial, o presente estudo propõe aplicar CNNs em robôs seguidores de linha para aprimorar percepção e tomada de decisão. Tradicionalmente dependentes de sensores simples, esses robôs podem alcançar maior adaptabilidade e precisão com aprendizado profundo, mesmo em ambientes com variações ou ruídos visuais. O desenvolvimento contempla a integração de câmeras, a definição da arquitetura de rede, a coleta e o processamento de dados, bem como a implementação do modelo em sistemas embarcados, visando não apenas a melhoria do desempenho, mas também a contribuição para o avanço da robótica educacional e aplicada.

MATERIAL E MÉTODOS

Para a realização do presente trabalho, os principais materiais utilizados na construção do robô incluem: um chassi, que serve como base estrutural; uma bateria LiPo 2S (7,4 V) para alimentação do sistema; dois motores CC (3–6 V, 200 RPM a 6 V) para locomoção do robô; um dispositivo com câmera (ESP32-CAM) responsável pela captura de imagens em tempo real; uma placa baseada em microcontrolador (ESP32) encarregada pela implementação do algoritmo de controle e uma ponte H (L298N) para o acionamento dos motores - além de componentes auxiliares para conexão e fixação de todos os elementos. Essa configuração representa a etapa planejada para a implementação final da rede neural convolucional (CNN), capaz de interpretar as imagens capturadas e determinar autonomamente o comportamento dos motores. O diagrama de blocos do sistema, apresentado na Figura 1, ilustra os principais componentes e a interação entre eles, destacando o fluxo de informações para o controle do movimento do robô.

FIGURA 1. Diagrama dos componentes/dados utilizados no controle do processo.

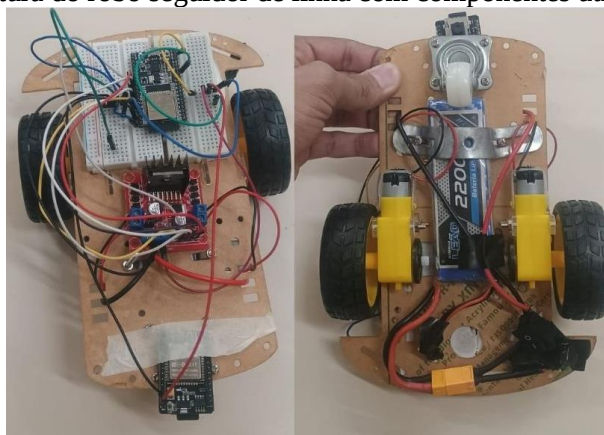


A Figura 2 apresenta a estrutura do robô seguidor de linha com todos os componentes e dispositivos integrados até o momento. O projeto encontra-se em andamento, razão pela qual a ESP32-CAM está fixada de forma provisória, sendo prevista sua fixação definitiva por meio de uma peça impressa em 3D.

Com a estrutura mecânica do robô devidamente definida, procedeu-se à configuração do sistema seguidor de linha pelo método tradicional, utilizando sensores reflexivos para detecção da linha e acionamento dos motores conforme o contraste da pista. Essa etapa teve como finalidade validar o sistema e capturar imagens para a formação do banco de dados destinado ao treinamento, teste e validação da rede neural convolucional (CNN). Para isso, foi desenvolvido um código em MicroPython,

posteriormente implementado no microcontrolador ESP32, enquanto o dispositivo ESP32-CAM foi programado para registrar imagens durante o percurso do robô na pista.

FIGURA 2. Estrutura do robô seguidor de linha com componentes utilizados até o momento



A rede neural convolucional (CNN) foi desenvolvida em Python, utilizando o ambiente de programação PyCharm. Para a implementação, foram empregadas diversas bibliotecas: “os” (manipulação de arquivos), “NumPy” (operações matriciais), “OpenCV” (aquisição e pré-processamento de imagens), “TensorFlow/Keras” (criação e treinamento da CNN), “scikit-learn” (divisão dos conjuntos de dados em treino e teste) e “Matplotlib” (visualização e análise de desempenho).

A Figura 3 ilustra exemplos das imagens capturadas pelo módulo ESP32-CAM durante a execução do robô seguidor de linha, já organizadas conforme suas respectivas classes — frente, esquerda e direita — que representam as possíveis direções de deslocamento adotadas pelo robô.

FIGURA 3. Exemplos de imagens capturadas pela ESP32-CAM com seus respectivos rótulos



A Figura 4 apresenta a rede neural convolucional utilizada, cuja arquitetura segue um modelo sequencial composto por camadas que extraem, reduzem e classificam características das imagens. A camada convolucional inicial, com 32 filtros 3×3 e ativação ReLU, identifica bordas e texturas simples, produzindo mapas de características de dimensões $126 \times 126 \times 32$. Em seguida, a camada *MaxPooling* (2×2) reduz a dimensão para $63 \times 63 \times 32$, concentrando as informações mais relevantes.

Na sequência, uma segunda camada convolucional, com 64 filtros 3×3 e ReLU, extrai padrões mais complexos, resultando em $61 \times 61 \times 64$, seguida por outra *MaxPooling* (2×2) que reduz para $30 \times 30 \times 64$. A camada *Flatten()* transforma esses mapas em um vetor de 57.600 valores, que alimenta uma camada densa (128, ReLU) responsável por combinar as características extraídas, enquanto a técnica de *Dropout* (0,5) reduz o overfitting desativando metade dos neurônios.

Por fim, a última camada densa (com 3 saídas e função Softmax) realiza a classificação final, produzindo probabilidades associadas às três classes de saída. A Tabela 1 apresenta o detalhamento de

cada etapa da rede, especificando a camada correspondente, sua função principal, a dimensão de saída e o número de parâmetros treináveis.

FIGURA 4. Arquitetura da rede neural convolucional utilizada para classificação de imagens.

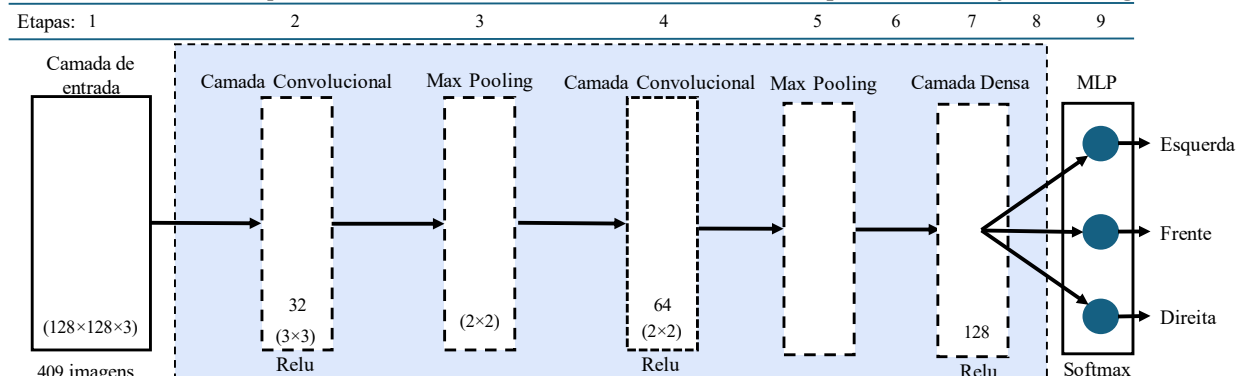


TABELA 1. Detalhamento por etapa da rede neural convolucional utilizada.

Etapa	Camada	Função Principal	Dimensão de Saída	Nº de Parâmetros treináveis
1	Entrada	Recebe o conjunto de 409 imagens RGB ($128 \times 128 \times 3$)	$128 \times 128 \times 3$	—
2	Conv2D (32 filtros, 3×3 , ReLU)	Extraí bordas e padrões simples	$126 \times 126 \times 32$	896
3	MaxPooling2D (2×2)	Reduz a dimensão espacial mantendo as informações principais	$63 \times 63 \times 32$	0
4	Conv2D (64 filtros, 3×3 , ReLU)	Detecta padrões mais complexos e abstratos	$61 \times 61 \times 64$	18.496
5	MaxPooling2D (2×2)	Reduz novamente a dimensão espacial	$30 \times 30 \times 64$	0
6	Flatten()	Converte cada mapa 3D em vetor 1D	57600	0
7	Dense (128, ReLU)	Combina as características extraídas	128	7.372.928
8	Dropout (0.5)	Reduz overfitting desativando neurônios aleatoriamente	128 (ativo na inferência)	0
9	Dense (Softmax, 3)	Classifica cada imagem em 3 categorias	409×3	387

Previamente ao treinamento, identificou-se um desequilíbrio entre as classes, uma vez que a pista possuía maior número de imagens com curvas à esquerda. Para corrigir essa distorção, foi realizado um balanceamento dos dados, de modo que cada classe passasse a conter aproximadamente 140 imagens. As imagens também foram redimensionadas de 1600×1200 para 128×128 pixels, a fim de otimizar o processamento e garantir compatibilidade com as capturas em tempo real da ESP32-CAM.

O conjunto de dados foi dividido em 70% para treinamento e 30% para validação, proporção que contribui para evitar sobreajuste (*overfitting*) e avaliar o desempenho real do modelo. O treinamento da rede neural convolucional foi realizado por 10 épocas.

RESULTADOS E DISCUSSÃO

Até o presente momento, o projeto encontra-se em desenvolvimento, portanto ainda não foi realizado o processo final do trabalho, que é o acionamento dos motores do robô seguidor de linha utilizando uma CNN. Apesar da plataforma ESP32 foi escolhida para controle do sistema, devido às limitações de processamento, a rede neural ainda não foi executada diretamente no dispositivo.

Na etapa atual, além da coleta de dados mencionada anteriormente, o treinamento da rede foi realizado em um computador pessoal, conforme o planejamento inicial. Concluída essa etapa, a implementação na ESP32 será a próxima fase.

O modelo de rede neural convolucional apresentou desempenho satisfatório, com acurácia global de aproximadamente 85%. A Figura 5 ilustra a evolução das acurácias de treino e validação ao longo das épocas, enquanto a Figura 6 apresenta a matriz de confusão do modelo.

FIGURA 5. Gráfico de treino x validação ao decorrer das épocas.
Acurácia durante o treinamento

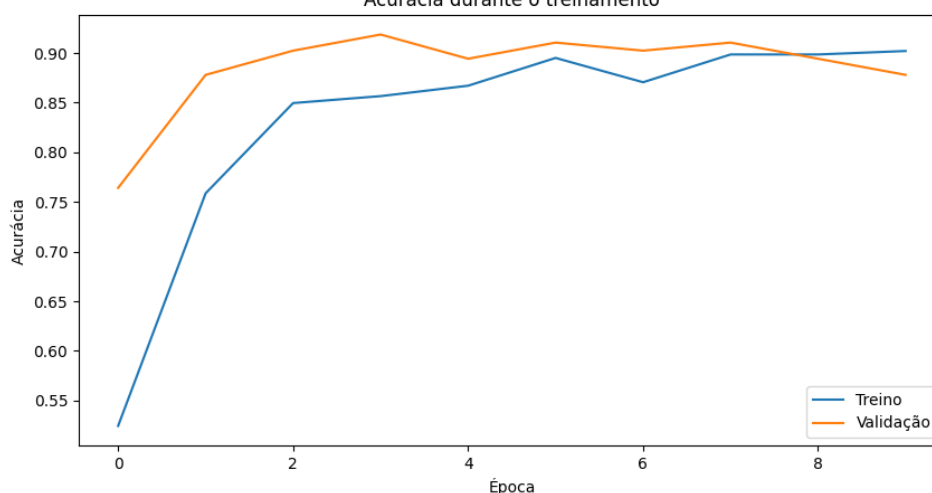
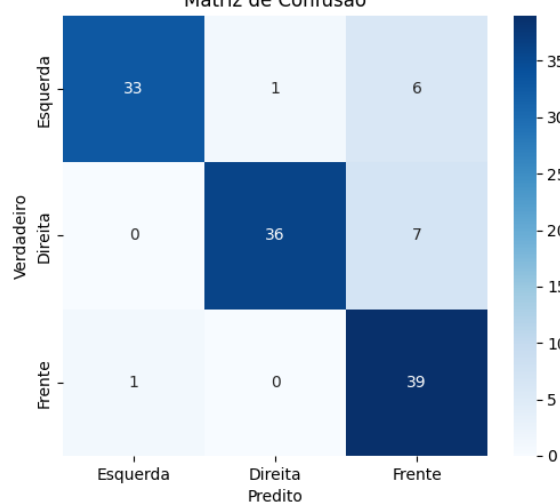


FIGURA 6. Matriz de confusão do modelo.
Matriz de Confusão



Com a acurácia do modelo validada, a etapa seguinte foi avaliar o desempenho do sistema na classificação em tempo real, analisando a taxa de quadros por segundo (FPS) para verificar a eficiência do processamento da rede neural. Observou-se, contudo, uma significativa latência ao integrar a CNN como classificador: a taxa de captura de imagens da ESP32-CAM, que girava em torno de 10 a 60 FPS (com média de 25 FPS) sem o classificador, diminuiu drasticamente para menos de 8 FPS após a implementação da rede neural. A Figura 7 ilustra essa comparação de desempenho. É importante notar que esta análise foi realizada em um computador; consequentemente, a intensa redução de desempenho inviabilizou, de forma temporária, a execução da rede neural diretamente na plataforma embarcada ESP32, devido às suas limitações de processamento.

FIGURA 7. Desempenho com taxa de FPS: (a) Sem classificação e (b) com classificação.



CONCLUSÕES

Este trabalho, ainda em andamento, tem como objetivo principal demonstrar a viabilidade de um sistema de controle aprimorado para robôs seguidores de linha baseado em visão computacional e redes neurais convolucionais. Para atingir esse fim, foi desenvolvida uma arquitetura CNN sequencial, a qual foi treinada com sucesso utilizando um conjunto de dados de imagem reais, coletados diretamente pelo protótipo do robô. O treinamento resultou em uma acurácia satisfatória de aproximadamente 85%, confirmando a eficácia do aprendizado profundo para classificar as ações de seguimento da linha.

Apesar da validação do modelo, a principal limitação observada reside na latência do processamento em tempo real. Essa latência provoca uma redução drástica na taxa de quadros por segundo (FPS), o que, atualmente, torna inviável a execução da CNN na plataforma embarcada ESP32. Para superar este desafio técnico, as próximas etapas do projeto focarão na exploração de diversas estratégias de otimização, incluindo: a utilização de imagens em escala de cinza (melhor pré-processamento) em vez de três canais RGB; a otimização do modelo via TensorFlow Lite; a adoção de arquiteturas CNN mais leves; ou, se necessário, a migração para plataformas de hardware mais robustas, como a Raspberry Pi.

CONTRIBUIÇÕES DOS AUTORES

O presente projeto servirá como base para o Trabalho de Conclusão de Curso, contribuindo de forma significativa para o aprofundamento acadêmico dos autores. Além disso, representa um avanço relevante para as atividades desenvolvidas pelo grupo de robótica do Instituto Federal de São Paulo – Campus Presidente Epitácio, fortalecendo as práticas de pesquisa, desenvolvimento e inovação no âmbito institucional.

REFERÊNCIAS

BROWNE, Matthew; GHIDARY, Saeed Shiry; MAYER, Norbert Michael. Convolutional neural networks for image processing with applications in mobile robotics. In: **Speech, Audio, Image and Biomedical Signal Processing using Neural Networks**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 327-349.

CHANG, Nai-Hsiang et al. A robot obstacle avoidance method using merged CNN framework. In: **2019 International Conference on Machine Learning and Cybernetics (ICMLC)**. IEEE, 2019. p. 1-5.

LAWRENCE, Steve et al. Face recognition: A convolutional neural-network approach. **IEEE transactions on neural networks**, v. 8, n. 1, p. 98-113, 1997.

O'SHEA, Keiron; NASH, Ryan. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015.

RUBIO, Francisco; VALERO, Francisco; LLOPIS-ALBERT, Carlos. A review of mobile robots: Concepts, methods, theoretical framework, and applications. **International Journal of Advanced Robotic Systems**, v. 16, n. 2, p. 1729881419839596, 2019.

UŠINSKIS, Vygantas et al. Sensor-fusion based navigation for autonomous mobile robot. **Sensors**, v. 25, n. 4, p. 1248, 2025.