

MODELO DE APRENDIZADO DE MÁQUINA PARA ANÁLISE CINEMÁTICA DA MARCHA EM PACIENTES HEMIPLÉGICOS

Eduarda Engels, Aurélio Faustino Hoppe – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil
eduarda.engels00@gmail.com, aureliof@furb.br

Resumo: A cinemática da marcha é o estudo dos movimentos relacionados ao deslocamento humano que desempenha um papel crucial na compreensão do impacto da hemiplegia, caracterizada pela paralisia de um dos lados do corpo, influenciando diretamente o padrão da marcha. Portanto, a análise da marcha é uma ferramenta essencial de auxílio a identificação de anormalidades e ao tratamento de indivíduos com dificuldades de locomoção. No entanto, para isso, são necessários equipamentos específicos e métodos de análise sofisticados, como sistemas de captura de movimento, plataformas de força, e análise de imagem. A partir deste contexto, este artigo apresenta o desenvolvimento de dois modelos para detecção e classificação de padrões de marcha. No processo de validação dos modelos YOLOv8 e CNN, foram utilizados dois conjuntos de dados que continham imagens representativas das Silhouettes, Gait Energy Image (GEIs), Skeletons, Skeleton Energy Image (SEIs), Normal Gait e Abnormal Gait. O modelo YOLOv8, pré-treinado na plataforma Roboflow, apresentou 70% de acurácia global, destacando-se com 82% em imagens da classe "Silhouettes". Por outro lado, o modelo CNN obteve acurácias próximas a 100% de precisão, mostrando-se uma alternativa viável para a classificação alguns tipos de cinemática da marcha.

Palavras-chave: Cinemática da marcha. Classificação. Marcha normal. Marcha Hemiplégica. Roboflow. YOLOv8. CNN.

1 INTRODUÇÃO

De acordo com a Organização Mundial da Saúde (2019), cerca de 10% da população mundial vive com algum tipo de deficiência, e grande parte dessas limitações está relacionada à mobilidade. Doenças como a esclerose múltipla, hemiplegia, a poliomielite e lesões na medula espinhal são exemplos que podem comprometer a capacidade de uma pessoa se locomover de maneira eficaz. Além disso, dados do Instituto Brasileiro de Geografia e Estatística (IBGE) indicam que, no Brasil, aproximadamente 6,7% da população possui alguma deficiência motora (MINISTÉRIO DA SAÚDE, 2017).

A mobilidade, um processo intrincado que exige a coordenação precisa de vários segmentos corporais e articulações, torna-se particularmente desafiadora quando afetada por condições de saúde como doenças neuromusculares, lesões traumáticas ou distúrbios ortopédicos (SCHMIDT *et al.*, 2019). A análise da cinemática da marcha, que se refere ao estudo dos movimentos angulares e lineares durante o ciclo locomotor, desempenha um papel crucial na compreensão desses desafios. Em um estado saudável, a mobilidade revela uma sequência harmoniosa de movimentos; entretanto, em casos de condições como a hemiplegia, essa harmonia é significativamente comprometida, impactando substancialmente a capacidade de mobilidade e exigindo abordagens especializadas no diagnóstico e tratamento desses distúrbios motores (PERRY *et al.*, 2010).

A hemiplegia, uma forma de paralisia unilateral, induz a padrões anômalos de movimento durante a marcha. As alterações na cinemática dessa locomoção em pacientes hemiplégicos abrangem assimetria na duração e amplitude das passadas, mudanças na fase de apoio e balanço, além de variações na velocidade angular das articulações (OLNEY *et al.*, 1996). Esta condição, comumente associada à paralisia de um lado do corpo, exerce um impacto substancial na cinemática da marcha. A análise cinemática desse processo em pacientes hemiplégicos oferece informações importantes sobre as modificações nos padrões de movimento, incluindo a redução no comprimento do passo, menor flexão do quadril e diminuição do tempo de apoio do membro afetado (LUO *et al.*, 2019).

De acordo com a Secretaria de Atenção à Saúde (2022), o tratamento da hemiplegia geralmente adota uma abordagem multidisciplinar, incorporando fisioterapia, terapia ocupacional e fonoaudiologia. Essa estratégia tem como foco a maximização da independência funcional do paciente, visando reduzir as limitações impostas pela paralisia. Além disso, as intervenções de reabilitação englobam o treinamento oromotor precoce, direcionado ao desenvolvimento adequado da fala e alimentação, com o objetivo de normalizar as funções respiratória, de deglutição e de mastigação. No contexto da reabilitação, destacam-se também exercícios específicos voltados para a melhoria da cinemática da marcha, como o treinamento de passadas em esteira com suporte de peso corporal, além de terapia manual para potencializar o desempenho do movimento articular (MINISTERIO DA SAÚDE, 2022).

Contudo, apesar da eficácia dessas abordagens, a análise da marcha para iniciar o tratamento muitas vezes demanda ferramentas sofisticadas e equipamentos específicos, como sistemas de captura de movimento, plataformas de força e análise de imagem. Esses recursos, devido ao seu elevado custo, podem não estar disponíveis em todos os centros de reabilitação, limitando o acesso e a avaliação da efetividade das intervenções terapêuticas. Diante disso, a aplicação de técnicas de aprendizado de máquina surge como uma alternativa promissora para o desenvolvimento de modelos preditivos. Esses modelos podem auxiliar na tomada de decisões clínicas, permitindo a identificação precoce de possíveis alterações na cinemática da marcha e o ajuste personalizado do tratamento conforme as necessidades individuais de cada paciente (HARRIS *et al.*, 2022).

Diante deste contexto, observa-se a necessidade de desenvolvimento de uma ferramenta que proporcione uma análise mais objetiva e contínua do progresso do paciente, trazendo uma contribuição significativa para a área da reabilitação, auxiliando na elaboração de terapias mais eficientes e personalizadas para os pacientes, além de poder ser aplicado em outras pessoas com problemas de locomoção. A partir disso, este trabalho propõe o desenvolvimento de um modelo de aprendizado de máquina capaz de identificar anomalias na marcha com ênfase em hemiplegia. Para isso, estabeleceu-se os seguintes objetivos específicos: (i) estruturar um *dataset* com imagens de distúrbios motores; (ii) identificar padrões temporais de cinemática da marcha a partir de imagens utilizando redes neurais convolucionais; (iii) efetuar a classificação de acordo com o tipo da marcha (normal, hemiplégica ou *abnormal*) e; (iv) avaliar o desempenho dos modelos utilizando métricas específicas para identificação de anomalias na marcha, tais como precisão, sensibilidade e confiabilidade.

2 FUNDAMENTAÇÃO TEÓRICA

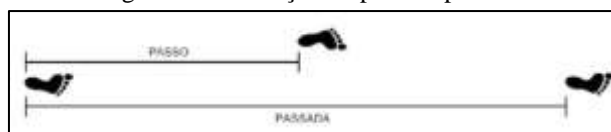
Este capítulo está subdividido em três seções. A seção 2.1 apresenta a análise cinemática da marcha, que consiste na técnica de avaliação da locomoção humana. Já na seção 2.2, conceitua-se aprendizado de máquina e suas principais arquiteturas. Por fim, na seção 2.3 são apresentados os trabalhos correlatos à esta pesquisa.

2.1 CINEMÁTICA DA MARCHA

Segundo Kim *et al.* (2018), identificar anormalidades na marcha é fundamental para o diagnóstico e tratamento de doenças e lesões que afetam o caminhar de uma pessoa, como o Acidente Vascular Cerebral (AVC), lesões medulares, doenças neuromusculares e ortopédicas. Além disso, os autores também afirmam que ao utilizar a análise cinemática da marcha para avaliar os efeitos de um programa de treinamento em pacientes com hemiplegia pode-se reduzir e alcançar resultados melhores na simetria da marcha em menos tempo.

Caparelli (2017) ainda afirma que a marcha é um meio natural de locomoção do corpo, envolvendo o uso alternado das duas pernas para proporcionar suporte e propulsão. O comprimento da passada refere-se à distância entre o contato inicial do calcanhar de um membro no solo e o próximo contato deste no chão, enquanto o passo é a distância longitudinal entre o contato no solo do calcanhar de um membro e o calcanhar contralateral. A frequência dos passos é conhecida como cadência. A Figura 1 apresenta a diferença entre a passada e o passo.

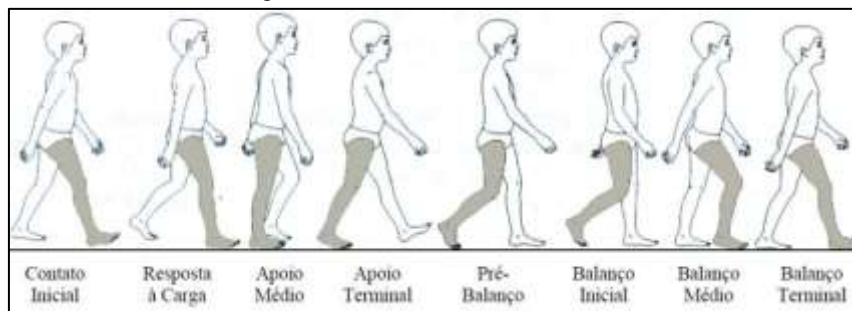
Figura 1 – Definição de passo e passada



Fonte: Caparelli (2017).

Segundo Racic, Pavic e Brownjohn (2009), a partida da marcha normalmente ocorre quando o indivíduo que deseja levantar o pé direito, iniciando o apoio neste mesmo lado e, em seguida, transferindo o peso para o pé esquerdo, permitindo que o pé direito se descole do solo. A eficiência desses movimentos depende da mobilidade articular livre e da atividade muscular adequada. Durante a marcha, o quadril desempenha um papel crucial, com uma amplitude de movimento típica de 10° de extensão e 30° de flexão na articulação. Clinicamente, a posição zero é definida como o tronco e a coxa em postura ereta. Já o joelho desempenha diversas funções durante a marcha, como sustentar o peso corporal, amortecer o impacto, aumentar o comprimento da passada e permitir o movimento do pé durante a fase de balanço. A Figura 2 mostra o caminhar com as fases da marcha.

Figura 2 – Fases da Marcha Humana



Fonte: Caparelli (2017).

Segundo Caparelli (2017), o ciclo da marcha humana normal compreende distintas fases, tarefas e subfases que delineiam o processo complexo de locomoção. Essas incluem as fases de apoio e balanço, tarefas como a aceitação do peso, apoio simples e avanço do membro, e subfases específicas, como contato inicial, resposta à carga, apoio médio, apoio terminal, pré-balanço, balanço inicial, balanço médio e balanço final.

De acordo com Gamble e Rose (2006), a análise cinemática durante um ciclo de marcha detalha os movimentos articulares dos membros inferiores e do tronco. Durante o contato inicial, observa-se o quadril em flexão, o joelho estendido e o tornozelo próximo à posição neutra. No apoio inicial, o quadril inicia o movimento de extensão, o joelho primeiro flexiona para, em seguida, estender, e o tornozelo ganha amplitude de dorsiflexão à medida que o centro de gravidade atravessa o pé. Na fase de pré-balanço, o quadril encontra-se em extensão, o joelho inicia o movimento de flexão, e ocorre a plantiflexão do tornozelo. Durante o período de balanço, ocorre a flexão de quadril, joelho e tornozelo para liberar o pé do chão (GAMBLE; ROSE, 2006).

Segundo Racic, Pavic e Brownjohn (2009), a análise da marcha utiliza parâmetros espaço-temporais para quantificar tanto o espaço quanto o tempo envolvidos nesse processo. Medidas espaciais comuns incluem o comprimento do passo e o comprimento da passada, enquanto parâmetros temporais frequentemente analisados abrangem o tempo da passada, velocidade e cadência. O comprimento do passo é definido como a distância entre o ponto onde o calcanhar de um membro toca o solo e o ponto em que o calcanhar do membro contralateral volta a tocar o solo. Já o comprimento da passada refere-se à distância entre o ponto onde o calcanhar de um membro toca o solo e o ponto em que o mesmo calcanhar toca o solo novamente (PERRY, 2005).

Por fim, Caparelli (2017) ressalta que a análise da marcha contribui para o aprimoramento do tratamento de indivíduos, proporcionando uma compreensão mais aprofundada do processo de locomoção. O autor também destaca a importância de compreender as características espaço-temporais da marcha, como o comprimento da passada, a duração da passada e a velocidade. Esses parâmetros, segundo Caparelli (2017), fornecem informações valiosas sobre a eficiência e o padrão de movimento durante a caminhada.

2.2 APRENDIZADO DE MÁQUINA

De acordo com Alpaydin (2012), *Machine Learning* (ML) é um ramo da inteligência artificial que utiliza técnicas computacionais para permitir que um sistema aprenda e melhore com a experiência. Segundo o autor, ML é uma abordagem de modelagem de dados que automatiza a construção de modelos analíticos. Entre suas principais características, destacam-se (i) o aprendizado automático, que aprendem a partir dos dados e se ajusta para melhoria de desempenho, (ii) a generalização, ou seja, fazer previsões e classificações em novos dados que não foram utilizados no treinamento, e (iii) a adaptabilidade, os modelos podem se adaptar às situações melhorando sua resposta com o tempo.

Neste contexto, nas subseções 2.2.1 e 2.2.2 apresentam, respectivamente, as definições e características dos modelos *Convolutional Neural Network* (CNN) e YOLOv8. As CNNs, destacam-se pela sua eficácia na extração de padrões em dados de grade, sendo especialmente relevantes para tarefas de visão computacional, como detecção e classificação de objetos em imagens. Por outro lado, o YOLOv8, como uma interação avançada da arquitetura YOLO, introduz inovações para aprimorar o desempenho em diversas tarefas, como detecção de objetos.

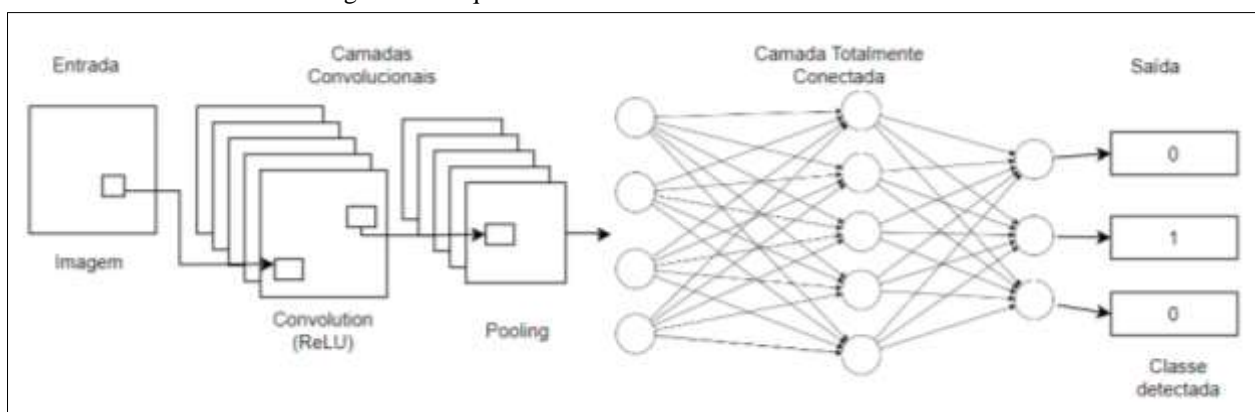
2.2.1 Redes Neurais Convolucionais

Segundo Rosa (2018), a Rede Neural Convolucional (do inglês *Convolutional Neural Network* - CNN) é uma classe da rede neural artificial que começou a ser aplicada no processamento e análise de imagens digitais. A CNN é uma variante da rede *perceptrons* de múltiplas camadas, e foi inspirada no processo biológico de processamentos de dados visuais. As células responsáveis pelo processamento dos dados visuais são sensíveis a pequenas sub-regiões do campo visual, também chamado de campo receptivo. Essas células são apropriadas para explorar a correlação espacial local presente em imagens naturais, funcionando como filtros locais sobre o campo de visão. Esse campo é abrangido pelos campos receptivos de diferentes neurônios, que se sobrepõem parcialmente. Imagens coloridas em treinamentos podem

resultar numa melhora significativa na interpretação dos dados, pois a CNN tira maior proveito nestas imagens do que em imagens em escala de cinza. Isto ocorre porque assim como nos humanos, a cor ajuda na análise e interpretação de objetos, facilitando a memorização e reconhecimento.

Segundo Rosa (2018), uma CNN consiste em múltiplas camadas com funções diferentes. É comum inicialmente aplicar sobre o dado de entradas as camadas que dão o nome à rede neural, chamadas de convolução. Uma camada de convolução é composta por diversos neurônios, cada uma com o trabalho de aplicar um filtro em uma região da imagem. Cada neurônio é conectado a um conjunto de *pixels* da camada anterior e a cada uma dessas conexões é atribuído um peso. A combinação das entradas de um neurônio, utilizando os pesos respectivos de cada uma de suas conexões, produz um resultado que será passado para a camada seguinte. O filtro de convolução é representado por uma matriz que contém os pesos atribuídos as conexões de um neurônio. As camadas são organizadas de forma a detectar padrões mais simples primeiro (linhas, curvas etc.) e padrões mais complexos (faces, objetos etc.) (MISHRA, 2020). A Figura 3 apresenta a arquitetura básica de uma rede neural convolucional.

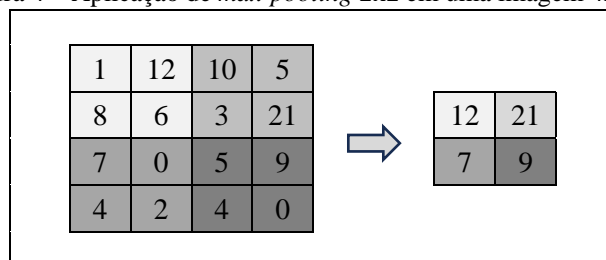
Figura 3 – Arquitetura de uma Rede Neural Convolucional



Fonte: adaptado de Soares e Carmo (2020).

Segundo Ribeiro (2020), geralmente o resultado da convolução é apresentado a função de ativação *Rectified Linear Units* (ReLU), pois é mais indicada para Redes Convolucionais por ser mais eficiente computacionalmente. A ReLU tem por objetivo zerar todos os valores negativos da camada anterior. O autor também ressalta que as camadas de *pooling* têm por função reduzir gradativamente a dimensão das representações da CNN, e como consequência reduzir a resolução do mapa de características e sensibilidade da saída as mudanças e distorções. Com isso, diminui-se ainda mais o número de parâmetros, tornando a complexidade computacional menor. O principal papel da camada de *pooling* é combinar recursos semanticamente semelhantes, fazendo com que as representações variem muito pouco quando os elementos na camada anterior variam em posição e aparência. A *max-pooling* é uma camada da *pooling* que computa o máximo local de uma determinada região do mapa de atributos, eliminando valores não máximos conforme pode-se observar na Figura 4. Ribeiro (2020) destaca que além de reduzir o tamanho da imagem e o processamento para a próxima camada, esta técnica auxilia no tratamento de invariâncias locais.

Figura 4 – Aplicação de *max-pooling* 2x2 em uma imagem 4x4



Fonte: adaptado de Ribeiro (2020).

Segundo Ribeiro (2020), em níveis mais altos a saída das camadas convolucionais e *pooling* representam características da imagem de entrada, e estas são passadas para as camadas totalmente conectadas para a classificação da imagem de acordo com as classes e conjuntos de treinamentos. As camadas totalmente conectadas possuem conexões com todos os neurônios da camada anterior e, estão ligadas a todos os neurônios da camada seguinte.

Soares e Carmo (2020) ressaltam que existem vários filtros convolucionais que extraem características de imagens. A utilização sobre a imagem possibilita o retorno de características desejadas, seus respectivos filtros são aplicados em várias camadas e o valor é incorporado pela própria rede, determinando assim quais são as características relevantes para se obter a solução de saída.

Ainda de acordo com Soares e Carmo (2020), as CNNs são métodos já conhecidos a anos, mas só estão ganhando destaque recentemente devido ao fato de ter grande capacidade computacional jamais vista até então. Tais algoritmos vem se demonstrando muito eficaz no processamento e análise de imagens digitais, por avaliar nativamente as informações espaciais. Segundo Cunha (2020), com o tempo e com base no desenvolvimento da tecnologia foram criando diversos modelos de CNN para resolver problemas mais específicos de acordo com a área, podendo citar: (i) R-CNN, *Fast R-CNN*, *Faster R-CNN* e YOLO que são utilizados principalmente na área de detecção de objetos (MISHRA, 2020).; (ii) *Visual Geometry Group* (VGG) uma rede neural convolucional utilizada principalmente na área de reconhecimento de imagens (BOESCH, 2023); (iii) No grupo dos *Inception-v3*, tem-se a GoogLeNet, como pioneira, pois esta CNN vem a ser um módulo da GoogLeNet. Utilizada principalmente na área de extração de características de imagens para reconhecimento; (iv) ResNet utilizada na área de reconhecimento por imagem (VAIDYA *et al.*, 2022).

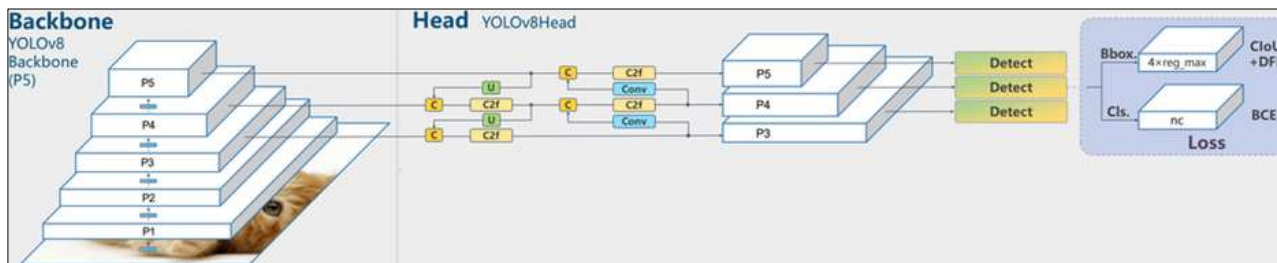
2.2.2 Detector de objetos YOLOv8

O YOLOv8 (*You Only Look Once version 8*), o novo modelo da arquitetura YOLO (*You Only Look Once*), desenvolvido em 2023, consolidando-se como *State-Of-The-Art* (SOTA), baseado nas versões anteriores, este modelo se destaca pela detecção de objetos, classificação de imagens e segmentação de instâncias, introduzindo inovações para impulsionar desempenho e versatilidade em uma ampla gama de tarefas em IA (Inteligência Artificial) de visão. Opera com duas licenças distintas: a AGPL-3.0, oferecendo uma abordagem de código aberto para estudantes e entusiastas, e uma licença empresarial destinada a uso comercial (JOCHER *et al.*, 2023).

Baseado em aprendizado profundo e visão computacional, possui uma precisão considerável. O YOLOv8 é altamente adaptável, permitindo uma implementação eficiente em *hardware* avançado e ambientes de nuvem, mantendo custos sob controle. Sua excepcional precisão é respaldada por avaliações em referências da indústria, como o conjunto de dados *Common Objects in Context* (COCO) e Roboflow, estabelecendo um novo padrão para modelos de detecção de objetos (JOCHER *et al.*, 2023).

O detector de objetos YOLOv8, que faz uso de uma rede neural convolucional de ponta a ponta, ou seja, não existem camadas densas na arquitetura. Isso na prática permite que sejam utilizadas imagens de diferentes resoluções, sem a necessidade de serem feitos redimensionamentos, o que ajuda o modelo a ter uma maior precisão na detecção de objetos menores. A Figura 5 apresenta a arquitetura do YOLOv8.

Figura 5 – Arquitetura YOLOv8



Fonte: Solawetz *et al.* (2023).

A arquitetura YOLO inicialmente recebe uma imagem de entrada, esta é processada através de uma rede neural convolucional chamada de *backbone*. Em casos de classificação de imagens, a saída final da rede é suficiente para fazer uma previsão. Em contrapartida, na detecção de objetos não somente é necessário identificar a classe do objeto, como também a área exata na imagem que ele ocupa, denominada *bouding-box* (BOCHKOVSKIY *et al.* 2020). Em síntese, é necessário sortear cuidadosamente as camadas de recursos do *backbone*, isto acontece na parte do algoritmo denominada *neck*. Além disso, os algoritmos de detecção de objetos podem ser categorizados em dois tipos: *one-stage detectors* e *one-two-stage detectors*. *Two-stage detectors* realizam a localização e classificação dos objetos separadamente para cada caixa delimitadora. Em contrapartida *one-stage detectors* realizam essas duas tarefas simultaneamente. Assim, somente uma execução na rede se demonstra necessária para realizar a detecção dos objetos na imagem. A arquitetura YOLO é um exemplo de *one-stage detectors* (TAN *et al.* 2020).

De acordo com Bochkovskiy *et al.* (2020), a YOLO head é segunda parte da arquitetura, sendo construída a partir das camadas desenvolvidas pelo *backbone*, esta desempenha um papel fundamental para a extração de características juntamente com a previsão realizada na inferência exercida durante as tarefas de detecção de objetos. Esta parte da arquitetura foi estruturada na versão YOLOv3 e então consolidada na versão YOLOv4, e por conseguinte, permanecendo até a versão atual. Esta estrutura é utilizada inicialmente para a detecção dos objetos desejados, incorporando etapas de detecção baseadas em âncoras e três níveis de granularidade. Redmon *et al.* (2018) demonstra que as detecções de objetos baseadas em âncoras se caracterizam por ser um método de detecção de objetos o qual utiliza *deep learning* e *bounding-boxes* pré-definidas (denominadas âncoras), como propostas para realizar a detecção. A ideia por traz de uma detecção de objetos utilizando âncoras consiste em referenciar e prever estas *bounding-boxes*. Portanto, cada objeto pode possuir

um *label* caracterizando a ancora em questão, e cada imagem pode ter ou não a presença destes, sendo que podem existir classes distintas de *labels*/âncoras.

De maneira geral, o modelo YOLO mostra-se um sistema de detecção de objetos altamente eficiente e preciso e que tem sido fortemente aderido na comunidade de visão computacional. Sua capacidade demonstra-se o *state-of-the-art* em realizar a detecção de objetos, principalmente em tempo real. Em contrapartida, mesmo que o modelo apresente grandes índices de performance, ainda é possível perceber *bottlenecks* quanto a questão do *hardware* a ser utilizado. Assim, a disponibilidade de GPUs melhores, possibilitariam resultados mais rápido e eficientes.

2.3 TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos com características semelhantes aos principais objetivos do trabalho desenvolvido. O Quadro 1 detalha o trabalho de Albuquerque *et al.* (2021) ao qual desenvolveram um *dataset* com Inteligência Artificial (IA) para classificar a marcha em diversas patologias, incluindo a hemiplegia. No Quadro 2 é descrito o trabalho de Sheth *et al.* (2023) que utilizou técnicas para identificação biométrica a partir da marcha humana. Por fim, no Quadro 3 apresenta o trabalho de Mukhtar *et al.* (2018) que desenvolveram um modelo para a identificação de deficiência física a partir de vídeos.

Quadro 1 – *Remote Gait Type Classification System Using Markerless 2D Video*

Referência	Albuquerque <i>et al.</i> (2021)
Objetivos	Analisar a marcha em diferentes patologias, incluindo Parkinson, diplegia, hemiplegia e neuropatia.
Principais funcionalidades	Classificação das patologias na marcha.
Ferramentas de desenvolvimento	<i>Convolutional Neural Network</i> (CNN), <i>Linear Discriminant Analysis</i> (LDA), <i>Recurrent Neural Network</i> (RNNs) e <i>Long Short-Term Memory</i> (LSTM)
Resultados e conclusões	O sistema atingiu uma precisão de classificação de 93,4% e 92,6% para as representações GEI e SEI da marcha no conjunto de dados GAIT-IT. Em comparação com os modelos VGG-19 de última geração, o sistema demonstrou uma melhoria média de 3,4% para GEIs e 1,3% para SEIs durante a avaliação cruzada em conjuntos adicionais, incluindo o GAIT-IST. Os resultados indicam uma capacidade aprimorada de generalização e desempenho competitivo, enquanto mantém uma complexidade computacional notavelmente menor, destacando-se como uma solução eficiente para a classificação de tipos de marcha a partir de vídeos 2D.

Fonte: elaborado pela autora.

Quadro 2 – *Gait Recognition Using Convolutional Neural Network*

Referência	Sheth <i>et al.</i> (2023).
Objetivos	Identificação biométrica a partir das características da marcha
Principais funcionalidades	Reconhecimento das passadas
Ferramentas de desenvolvimento	<i>Convolutional Neural Network</i> (CNN)
Resultados e conclusões	O modelo de reconhecimento de marcha por CNN, treinado com o conjunto de dados CASIA-B, alcançou precisões acima de 90% na identificação de pessoas com base na marcha, mostrando potencial para aplicações práticas em reconhecimento remoto e segurança.

Fonte: elaborado pela autora.

Quadro 3 – *Gait Analysis of Pedestrians with the Aim of Detecting Disabled People*

Referência	Mukhtar <i>et al.</i> (2018).
Objetivos	Identificação de características na marcha para a detecção automatizada de pessoas com deficiência física em vídeos.
Principais funcionalidades	Reconhecimento de padrões para analisar assinaturas da marcha e distinguir características de movimento associadas a diferentes condições físicas.
Ferramentas de desenvolvimento	<i>You Only Look Once</i> (YOLO), <i>Convolutional Neural Networks</i> (CNN) e <i>Gaussian Mixture Model</i> (GMM)
Resultados e conclusões	Embora a abordagem automatizada para reconhecimento da marcha não tenha conseguido distinguir pessoas com deficiência, a análise manual sugere que há informações suficientes na marcha para caracterizar movimentos saudáveis, com uma diferença de 47% nos valores de pico entre as pernas de pessoas com deficiências e saudáveis, o erro de localização do YOLO influenciou negativamente nos resultados.

Fonte: elaborado pela autora.

A partir dos quadros acima, pode-se observar que todos os autores utilizam modelos de avaliação dos membros inferiores para a identificação da marcha humana. Destaca-se o trabalho de Albuquerque *et al.* (2021), que para a coleta de dados utilizaram vídeos gravados em *smartphones*. Além disso, os pesquisadores capturaram a simulação do caminhar de indivíduos com as patologias estudadas. Esses vídeos formaram a base de dados, composta por sequências de imagens, que alimentou os algoritmos de aprendizado de máquina, incluindo CNN, LDA, RNNs e LSTM. Para a categorização das patologias estudadas, utilizaram a detecção da silhueta da marcha humana, no qual obtiveram a taxa de acerto superior a 90%.

Sheth *et al.* (2023) demonstraram potencial biométrico eficaz com o *dataset* denominado CASIA-B, utilizando o *Gait Energy Image* (GEI) como entrada para CNN a fim de detectar a silhueta da imagem e classificá-las de acordo com as articulações e ângulos das passadas. Os autores visavam a identificação do indivíduo a partir da marcha humana. Sheth *et al.* (2023) apontam que o modelo de CNN alcançou uma precisão superior a 90% na identificação remota.

Por fim, Mukhtar *et al.* (2018) desenvolveram um modelo que utiliza imagens de câmeras de vigilâncias para identificar se o indivíduo possui ou não uma deficiência física a partir da marcha humana. Os autores destacam que a categorização foi realizada através da detecção e extração da silhueta da imagem. Para isso, Mukhtar *et al.* (2018) utilizaram redes como *Graph Neural Networks* (GMM), CNN e YOLO na implementação do modelo. Com relação aos resultados os autores mencionaram que o YOLO influenciou negativamente nos resultados da predição, porém, destacaram a presença de informações úteis na marcha para caracterização de movimentos saudáveis.

Diante do cenário apresentado, pode-se observar que os trabalhos utilizam diferentes métodos de identificação da marcha humana para se encaixar dentro do contexto estudado. No entanto, também se percebe que há necessidade evidente de desenvolver modelos mais adequados para analisar informações cinemáticas da marcha, considerando os resultados e lacunas dos modelos existentes.

3 DESCRIÇÃO DO MODELO

Este capítulo está dividido em 3 seções. A seção 3.1 trata sobre os requisitos funcionais e não funcionais do modelo de classificação. A seção 3.2 detalha a estruturação dos *datasets*. Por fim, a seção 3.3 discorre sobre a implementação dos modelos YOLOv8 e CNN e os resultados obtidos a partir dessas arquiteturas.

3.1 ESPECIFICAÇÃO

Esta seção apresenta os Requisitos Funcionais (RF) e Não Funcionais (RNF) que foram utilizados como base para o desenvolvimento dos modelos. O Quadro 4 apresenta os Requisitos Funcionais e o Quadro 5 apresenta os Requisitos Não Funcionais.

Quadro 4 – Requisitos Funcionais

RF01 – permitir a entrada de imagens ou vídeos da marcha humana para análise e classificação
RF02 – binarizar as imagens de entrada
RF03 – efetuar a análise cinemática dos membros inferiores utilizando redes neurais convolucionais (CNN e Yolo)
RF04 – classificar e demarcar o tipo da marcha como normal ou hemiplégica ou abnormal
RF05 – ser adaptável a diferentes variações da marcha, considerando distintos biotipos e estilos de movimento

Fonte: elaborado pela autora.

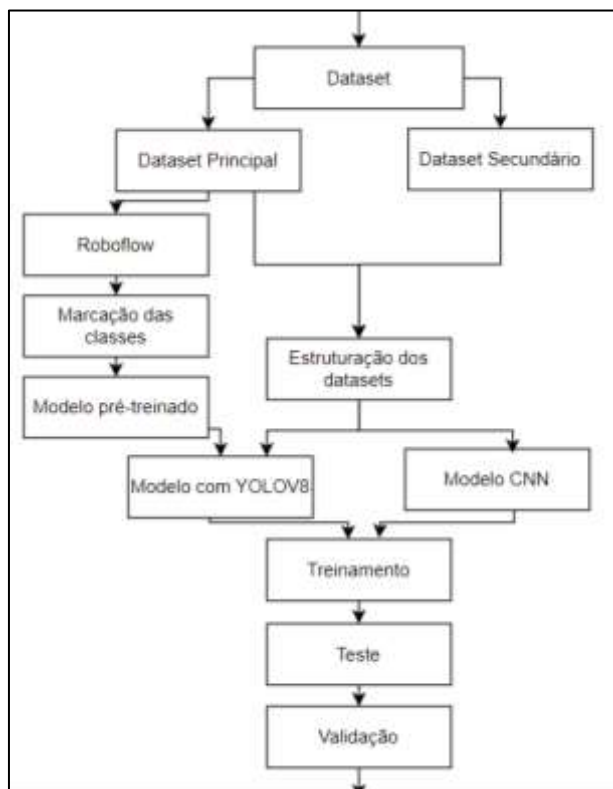
Quadro 5 – Requisitos Não Funcionais

RNF01 – ser capaz de aprimorar o modelo através da inclusão de novas imagens e vídeos de entrada, permitindo uma constante evolução e adaptação.
RNF02 – o modelo deve apresentar alta precisão na classificação da marcha, assegurando resultados confiáveis
RNF03 – ser desenvolvido utilizando as bibliotecas OpenCV para o processamento de imagens e o Keras ou <i>Tensorflow</i> para a construção da rede neural artificial
RNF04 – utilizar a plataforma Roboflow como ferramenta de versionamento de dados

Fonte: elaborado pela autora.

O diagrama de atividades, apresentado na Figura 6, mostra uma visão geral e esquemática do fluxo adotado durante todo o processo de classificação da cinemática da marcha, desde a aquisição e preparação do conjunto de dados até as fases de treinamento, teste e validação.

Figura 6 – Diagrama das etapas para a criação dos modelos de classificação da cinemática da marcha



Fonte: elaborada pela autora

O processo de implementação do modelo teve início com a preparação do conjunto de dados, envolvendo não apenas a organização, mas também o estabelecimento de divisões de pastas de forma estratégica para configurar uma gama diversificada de cenários de teste. Foram utilizados dois conjuntos de dados provenientes de fontes distintas: (i) obtido a partir do trabalho de Albuquerque *et al.* (2021) e (ii) por meio de vídeos existentes na internet.

Em seguida, aplicou-se dois modelos de redes neurais convolucionais: (i) YOLOv8 e (ii) CNN. No contexto do modelo YOLOv8, primeiramente realizou-se a marcação manual das imagens para a distinção das amostras em classes. A partir disso, um modelo pré-treinado fundamentado no YOLOv8, foi gerado utilizando a plataforma Roboflow. Dessa forma, torna-se possível efetuar o treinamento e validação do modelo YOLOv8. Por fim, são efetuados testes manuais para verificar a eficácia do modelo proposto a partir das bases de dados.

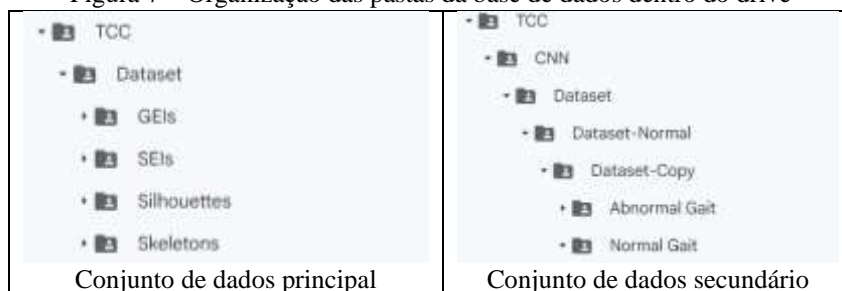
Simultaneamente, um segundo modelo foi implementado, valendo-se de uma *Convolutional Neural Network* (CNN) com classificação binária das classes. A implementação desse modelo abrangeu 10 épocas de treinamento, seguidas por teste e validação. Esses processos foram conduzidos de forma minuciosa, utilizando os *datasets* nos cenários de testes predefinidos, consolidando, assim, uma análise abrangente do desempenho do modelo em diversas situações.

3.2 ESTRUTURAÇÃO DOS DATASETS

O conjunto de dados principal originou-se do artigo de Albuquerque *et al.* (2021). Para acessar esse conjunto de dados, assinou um termo de compromisso para obter a senha de acesso, permitindo o uso por até dois anos. A necessidade de assinar o termo de compromisso enfatiza a importância ética e legal associada ao uso desse *dataset* específico. Este *dataset* foi construído a partir de imagens geradas por meio de simulações de patologias por parte dos indivíduos, utilizando inteligência artificial para criar representações binarizadas do ciclo da marcha.

Os *datasets* foram hospedados no Google Drive, escolhido pela sua fácil integração com o ambiente do Google Colab. Os *datasets* foram estruturados em subpastas contendo diversos tipos de imagens de marcha, como *silhouettes*, *GEIs*, *skeletons* e *SEIs*, abrangendo uma representação completa do comportamento humano durante a locomoção. Estruturou-se dois *datasets* distintos, cada qual passando por uma preparação específica para o modelo utilizado. A organização das pastas no *drive* foi realizada conforme ilustrado na Figura 7. No lado esquerdo, estão armazenadas as imagens do conjunto de dados principal, oriundo do artigo "*Remote Gait Type Classification System Using Markerless 2D Video*" de Albuquerque *et al.* (2021). No lado direito, encontram-se as pastas do conjunto de dados secundário, oriundo de vídeos com gravações das marchas, que possui as pastas "*Abnormal Gait*", para marcha anormal, e "*Normal Gait*" para a marcha normal.

Figura 7 – Organização das pastas da base de dados dentro do drive



Fonte: elaborado pela autora.

O *dataset* principal possui dois diretórios principais, ambos contendo quatro subpastas distintas: "*Silhouettes*" (marcha estática), "*GEIs*" (imagens de marcha em movimento), "*Skeletons*" (marcha estática do esqueleto humano) e "*SEIs*" (imagens do esqueleto humano em movimento). A distribuição, em termos quantitativo, do conjunto de dados principal (*Silhouettes*, *GEIs*, *Skeletons* e *SEIs*) pode ser observada no Quadro 6.





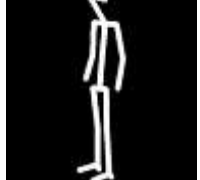


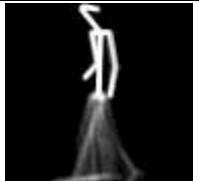
Quadro 6 – Distribuição de amostras por base/pasta e classe

	<i>Silhouettes</i>	<i>GEIs</i>	<i>Silhouettes+GEIs</i>	<i>Skeletons</i>	<i>SEIs</i>	Total
Normal	824	547	684	750	314	1256
Hemiplegica	3138	1027	2054	2364	1015	4060
Total	3962	1574	3148	3115	1329	5316

Fonte: elaborado pela autora.

Cada subpasta do conjunto de dados principal proporciona diferentes perspectivas e representações da marcha humana, conforme demonstra o Quadro 7. A classificação foi conduzida a partir das pastas, separadas em "*Normal*" e "*Hemiplegic*".

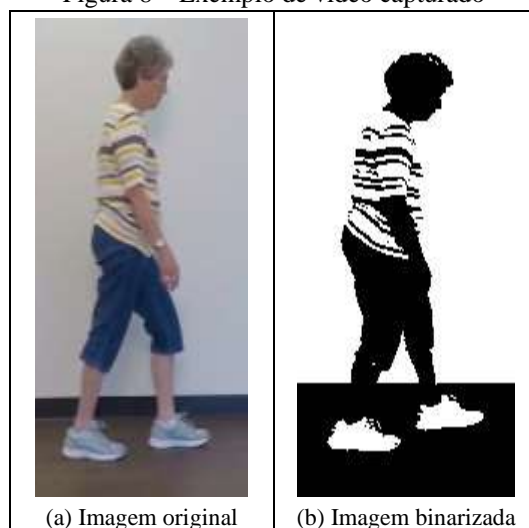
Quadro 7 – Exemplificação das imagens do *dataset principal*

Marcha Normal	Marcha Hemiplégica	Arquivo/pasta	Número de imagens utilizadas
		<i>Silhouettes</i>	3962
		<i>GEIs</i>	1574
		<i>Skeletons</i>	3115
		<i>SEIs</i>	1329

Fonte: elaborado pela autora.

Já o conjunto de dados secundário, também utilizado neste trabalho, foi constituído a partir de vídeos que capturaram a marcha de indivíduos em um ambiente controlado, conforme exhibe a Figura 8.

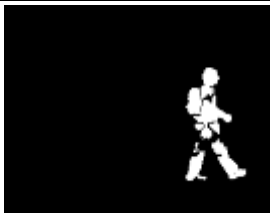

Figura 8 – Exemplo de vídeo capturado



Fonte: elaborado pela autora

Para representar distintamente a marcha normal e a hemiplegia, as imagens/*frames* dos vídeos foram binarizadas (Figura 8 item b) e classificadas como "*Normal*" e "*Abnormal*", respectivamente, conforme exemplificado no Quadro 8. Ressalta-se que este conjunto de dados proporcionou uma abordagem complementar, enfatizando uma representação específica da locomoção humana.

Quadro 8 – Exemplificação das imagens do *dataset* secundário

Marcha	Arquivo	Número de imagens utilizadas
	Normal Gait	1500
	Abnormal Gait	1500

Fonte: elaborado pela autora.

No total, o *dataset* secundário é composto por 15689 imagens de marcha normal e 15891 imagens caracterizadas como marcha abnormal. Optou-se por limitar a amostra a 1500 imagens para cada classe, totalizando 3 mil imagens. Para a seleção das imagens, utilizou-se a função "Random" para escolha aleatória. Essa decisão foi tomada com o intuito de gerenciar a complexidade computacional e otimizar o tempo de processamento durante o treinamento e teste do modelo.

3.3 IMPLEMENTAÇÃO

O desenvolvimento deste trabalho foi realizado no Google Colab, uma plataforma baseada em nuvem que oferece acesso a *Graphics Processing Units* (GPUs) e a linguagem Python. A escolha do *colab* foi motivada pela necessidade de lidar com grandes conjuntos de dados e realizar treinamentos complexos em modelos de aprendizado de máquina. Durante o processo, as opções Colab Pro e Colab Pro+ foram assinadas para superar limitações de recursos, como GPU e armazenamento, agilizando o processo de execução. Em determinados momentos, a exibição de imagens foi desativada para otimizar os recursos, e os modelos foram executados com a opção de "Ram alta" ativada.

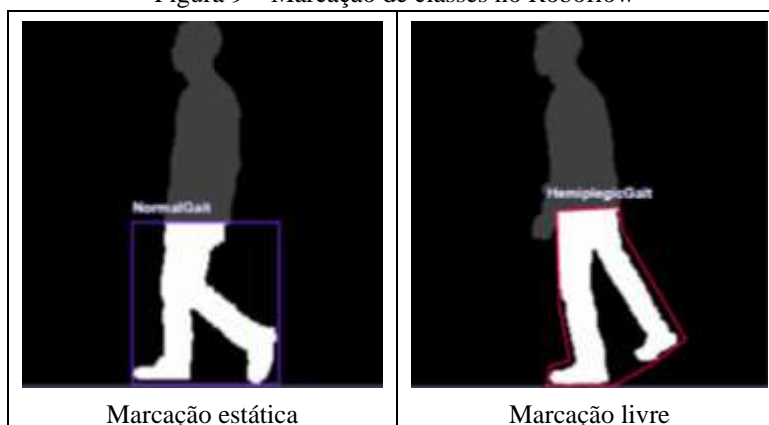
Com a organização dos conjuntos de dados finalizada, o foco do trabalho direcionou-se para a estruturação dos modelos de redes neurais convolucionais. Este processo foi conduzido de maneira sistemática, dividindo a implementação em dois módulos distintos: (i) um baseado no modelo YOLOv8 e (ii) outro na arquitetura de *Convolutional Neural Network* (CNN). Para tornar a exposição mais acessível, optou-se por adotar uma explicação por módulo, delineando as

características e nuances específicas de cada modelo para facilitar a compreensão e análise. Na subseção 3.2.1 detalha-se a implementação e os resultados do modelo YOLOv8. Por fim, a seção 3.2.2 descreve a implantação e os resultados do modelo CNN.

3.3.1 Implementação e resultados do Modelo YOLOv8

O fluxo de atividades realizadas no desenvolvimento do modelo YOLOv8 é composta pelas seguintes etapas: (i) marcação e treinamento das imagens no Roboflow e; (ii) utilização do modelo pré-treinado no *colab*. Inicialmente, realizou-se o *upload* das imagens que irão compor o *dataset*, anotando-as no Roboflow. Foram utilizadas apenas imagens da pasta "*silhouettes*", sendo categorizadas em duas classes: "NormalGait", equivalente ao `class_id` 0, para representar imagens de marcha normal e "HemiplegicGait", equivalente ao `class_id` 1, para imagens de marcha hemiplégica. Para isso, realizou-se a marcação manual nas imagens, utilizando a marcação estática (*bounding boxes*) e a livre para moldar a espessura da imagem, conforme exemplifica a Figura 9.

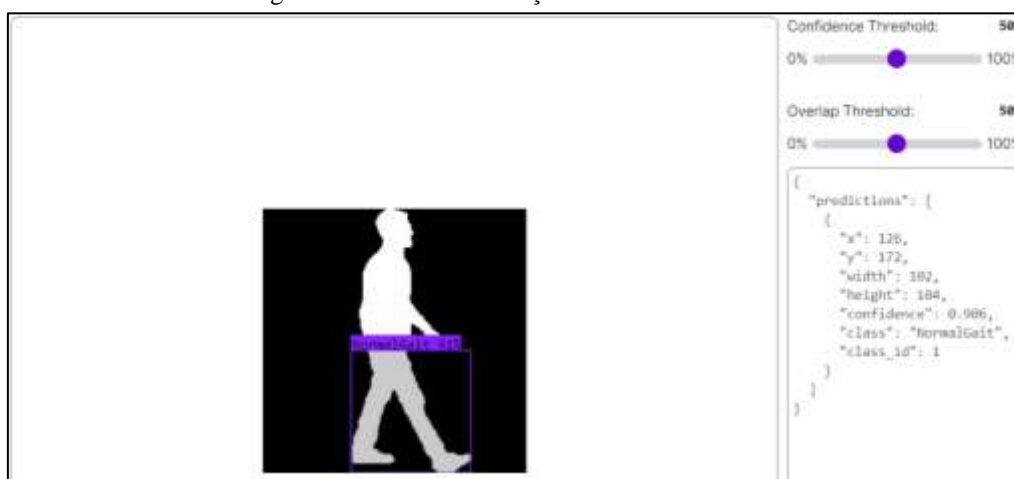
Figura 9 – Marcação de classes no Roboflow



Fonte: elaborado pela autora.

Após o processo de anotação de 7626 imagens, dividiu-se os conjuntos de treinamento (70%), teste (15%) e validação (15%). A análise final da execução do treino do modelo YOLOv8 se alcançou uma taxa de 99,8% de precisão. Por fim, inferiu-se outras imagens, a partir do Roboflow, para testar a detecção do modelo pré-treinado. No teste, o modelo obteve-se um percentual de 91% de confiabilidade de que a imagem pertence a classe Normal. No entanto, vale ressaltar que a plataforma permite a realização de ajustes no *threshold* de confiabilidade, conforme exibe a Figura 10.

Figura 10 – Teste de detecção dentro do Roboflow



Fonte: elaborado pela autora.

A partir do modelo do YOLOv8 pré-treinado, gerado no Roboflow, desenvolveu-se o modelo de classificação na linguagem Python. Inicialmente, aplicou-se ele a cada uma das quatro subpastas dos *datasets* para verificar qual tipo de imagem ele teria a maior precisão (*Silhouettes*, *GEIs*, *Skeletons* e *SEIs*, ver exemplo de imagens no Quadro 7). No *colab*, realizou-se a instalação do pacote de dados do Roboflow, inserindo a chave da API, conforme mostra o pseudocódigo para conexão com a API do Roboflow represente no Quadro 9.

Quadro 9 – Código fonte da conexão da API do Roboflow

```
01 #Instala a API do roboflow
02 !pip install roboflow
03
04 from roboflow import Roboflow
05 rf = Roboflow(api_key="M*****")
06 project = rf.workspace().project("gaite_v2")
07 model = project.version(2).model
```

Fonte: elaborado pela autora

Utilizando a API do Roboflow, foram inseridas imagens originadas do *dataset* principal para realizar as predições. O Quadro 10 demonstra a chamada da função `predict`, passando como parâmetro uma imagem, a taxa de confiabilidade e o nome da imagem resultante que irá conter a demarcação da classe encontrada.

Quadro 10 – Código fonte de teste do modelo pré-treinado

```
01 teste = model.predict("frame200.jpg", confidence=40, overlap=30).json()
02 print(teste)
03 model.predict("frame200.jpg", confidence=40, overlap=30).save("prediction.jpg")
04 results = io.imread("prediction.jpg")
05 io.imshow(results)
```

Fonte: elaborado pela autora

Ao finalizar os testes em cada subpasta, agrupou-se todas as quatro subpastas, escolhendo aleatoriamente as imagens de forma a equilibrar o número de amostras de cada classe. Realizou-se a randomização com o auxílio das bibliotecas `Random` e `glob` do Python, utilizando o Google Drive para importar os diretórios/arquivos, conforme mostra no Quadro 11.

Quadro 11 – Normalização das imagens do *dataset*

```
01 import glob
02 import random
03 from google.colab import drive
04 drive.mount('/content/drive')
05
06 min_images = min(
07     len(image_paths_Skeletons), len(image_paths_SEIs), len(image_paths_GEIs),
08     len(image_paths_Silhouettes)
09 )
10 image_paths_Skeletons = random.sample(image_paths_Skeletons, min_images)
11 image_paths_SEIs = random.sample(image_paths_SEIs, min_images)
12 image_paths_GEIs = random.sample(image_paths_GEIs, min_images)
13 image_paths_Silhouettes = random.sample(image_paths_Silhouettes, min_images)
14
15 all_image_paths = image_paths_GEIs + image_paths_SEIs + image_paths_Skeletons +
16 image_paths_Silhouettes
```

Fonte: elaborado pela autora.

Para a implementação do modelo no *colab* criou-se, para cada conjunto de dados, um código específico observando principalmente a taxa de precisão. Devido ao grande número de imagens em cada *dataset*, tornou-se necessária a normalização e redução para otimizar o desempenho do programa.

O treinamento ocorreu separadamente para cada um dos 2 conjuntos de dados iniciais, totalizando 3963 imagens do *dataset* “*Silhouettes*” e 1574 imagens do “*GEIs*”. Adicionalmente, um terceiro conjunto de teste foi criado combinando imagens dos dois *datasets*, porém realizando a normalização randômica para obter o mesmo número de imagem em ambos os *datasets*. Para o *dataset* de “*Skeletons*”, foi necessário reduzir para 3115 imagens, enquanto o *dataset* de “*SEIs*” todas as 1329 imagens foram utilizadas devido ao baixo número de amostras.

A implementação do modelo de treinamento (YOLOv8 pré-treinado no Roboflow) foi a mesma para todos os conjuntos de dados, tendo apenas a variação do conjunto de entrada. Conforme mostra o Quadro 12, para a implementação do modelo os conjuntos de treinamento, teste e validação foram divididos em 70%, 15% e 15%, respectivamente.

Quadro 12 – Código fonte do treinamento do modelo

```

01 from sklearn.model_selection import train_test_split
02 train_dir = "train_images"
03 test_dir = "test_images"
04 validation_dir = "validation_images"
05 os.makedirs(train_dir, exist_ok=True)
06 os.makedirs(test_dir, exist_ok=True)
07 os.makedirs(validation_dir, exist_ok=True)
08
09 count_modelo = 0
10 #treinamento: 70%
11 for image_path in train_images:
12     if image_path is not None:
13         result = model.predict(image_path, confidence=40, overlap=30)
14         print(result)
15         result.save(os.path.join(train_dir, os.path.basename(image_path)))
16         count_modelo+=1
17         print("Treinamento: ", count_modelo)
18
19 #teste: 15%
20 for image_path in test_images:
21     if image_path is not None:
22         result = model.predict(image_path, confidence=40, overlap=30)
23         print(result)
24         result.save(os.path.join(test_dir, os.path.basename(image_path)))
25         count_modelo+=1
26         print("Teste: ", count_modelo)
27
28 #Validação: 15%
29 for image_path in validation_images:
30     if image_path is not None:
31         result = model.predict(image_path, confidence=40, overlap=30)
32         print(result)
33         result.save(os.path.join(validation_dir, os.path.basename(image_path)))
34         count_modelo+=1
35         print("Validação: ", count_modelo)

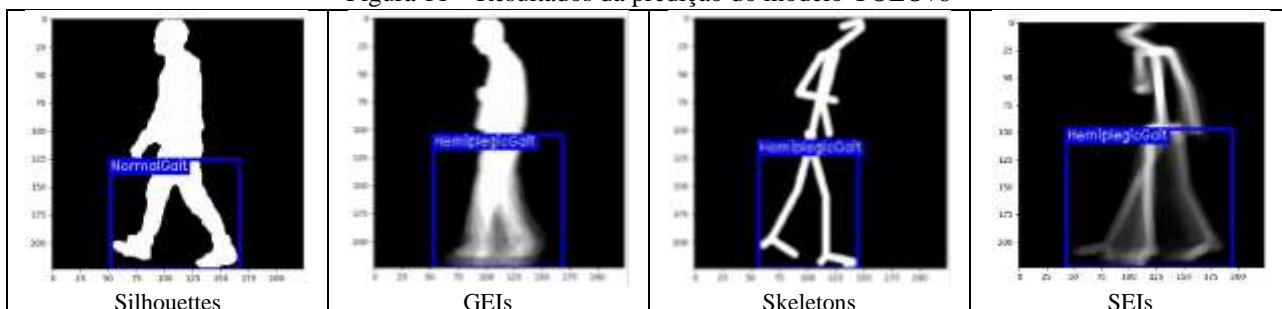
```

Fonte: elaborado pela autora.

Inicialmente, a variável `count_modelo` é inicializada em zero para rastrear o número total de imagens processadas pelo modelo. O processo inicia-se com a etapa de treinamento, abrangendo 70% do conjunto de dados. Em um *loop* iterativo, cada imagem em `train_images` é submetida ao modelo para predição. Os resultados são prontamente impressos e as imagens resultantes são arquivadas no diretório de treinamento (`train_dir`). A variável `count_modelo` é incrementada a cada interação para acompanhar o progresso, sendo acompanhada por uma mensagem que indica claramente a etapa de treinamento. Este procedimento é replicado para os conjuntos de teste (15%) e validação (15%), onde diretórios distintos (`test_dir` e `validation_dir`) são empregados. O controle sistemático de `count_modelo` em cada interação fornece uma visão detalhada do desempenho do modelo em cada fase do processo, contribuindo para uma compreensão abrangente do seu comportamento em diferentes conjuntos de dados.

Ao final de cada treinamento, um teste randômico com 10 imagens, do *dataset* utilizado, busca reafirmar a acurácia da predição. Para uma segunda validação, o caminho da imagem no Google Drive foi verificado para confirmar a correta classificação. Por fim, os quatro *datasets* foram integrados em um único programa, normalizando as imagens e selecionando aleatoriamente para igualar o tamanho do menor conjunto. Um total de 5316 imagens foi utilizado, sendo 1329 de cada *dataset*. Na Figura 11 exemplifica o resultado desta predição.

Figura 11 – Resultados da predição do modelo YOLOv8



Fonte: elaborado pela autora.

A partir das validações, um segundo conjunto de teste distinto foi avaliado utilizando o mesmo modelo YOLOv8 pré-treinado. Este conjunto foi gerado a partir de vídeos que capturam a marcha de indivíduos em um ambiente controlado, sendo dividido em duas categorias: "*Normal*" e "*Abnormal*", sendo composto por 3 mil imagens. Ao final deste processo, calculou-se a média de confiança das predições nas fases de treino e validação.

Os resultados obtidos com o modelo YOLOv8 proporcionaram uma análise profunda da sua capacidade de classificação em diferentes conjuntos de dados, cada um representando aspectos específicos da marcha humana. A avaliação do desempenho abrangeu diversas categorias, proporcionando uma compreensão abrangente da eficácia do modelo. O Quadro 13 apresenta os resultados obtidos pelo modelo YOLOv8 nos conjuntos de dados preestabelecidos como cenário de teste.

Quadro 13 – Resultados do modelo YOLOv8

Datasets	Acurácia		
	Treinamento (%)	Validação (%)	Validação com 10 imagens não incluídas nos datasets (%)
<i>Silhouettes</i>	82,00	80,00	82,00
GEIs	63,00	61,00	67,00
<i>Silhouettes</i> +GEIs	74,00	76,00	73,00
SEIs	55,00	53,00	55,00
<i>Skeletons</i>	67,00	61,00	62,00
ALL (<i>Silhouettes</i> +GEIs+SEIs+ <i>Skeletons</i>)	70,00	76,00	70,00
Dataset Secundário (<i>Normal</i> + <i>Abnormal</i>)	54,00	41,00	54,00

Fonte: elaborado pela autora.

A partir do Quadro 13 é possível observar que no *dataset* de silhuetas (representando a marcha estática), o modelo alcançou uma acurácia de 80%, destacando sua robustez ao distinguir entre marcha normal e hemiplegia. Para as imagens com GEIs (representando a marcha em movimento), a acurácia foi de 61%, revelando desafios adicionais na identificação de padrões nesse conjunto em específico. A combinação de silhuetas estáticas e da marcha em movimento (*Silhouettes*+GEIs) resultou em uma acurácia de 76%, representando uma melhora significativa em relação às classificações individuais.

No conjunto de dados de "*Skeletons*" (esqueletos), o modelo alcançou uma acurácia de 67%, indicando uma certa capacidade na interpretação de características baseadas na estrutura corporal. A classificação do conjunto SEIs (esqueletos em movimento) apresentou uma acurácia de 55%, apresentando dificuldades na identificação de padrões dinâmicos.

Contudo, em relação aos testes realizados a partir do conjunto de dados principais, obteve-se uma acurácia geral de 76%, evidenciando a habilidade do modelo em generalizar para uma ampla variedade de condições de marcha. Além disso, na utilização do modelo do YOLOv8 com o *dataset* secundário que abrange as classes "*Normal*" e "*Abnormal*," observou-se uma acurácia de 41%, indicando uma consistência ruim no desempenho do modelo, mesmo quando confrontado com amostras inesperadas desse conjunto de dados.

A análise dos resultados obtidos para 10 imagens não incluídas no *dataset* principal proporciona uma visão abrangente do desempenho dos modelos em relação a análise de marcha. A representação baseada em silhuetas demonstrou uma precisão sólida de 82% na validação, indicando uma capacidade robusta de reconhecer e analisar padrões de marcha em cenários não familiares. Por outro lado, as representações utilizando GEIs e esqueletos apresentaram uma precisão de 62% e 67%, respectivamente, indicando um desempenho moderado nessa condição específica.

A métrica global "*ALL*" considerando todas as representações atingiu uma precisão de 70% na validação, tendo um desempenho razoável no conjunto das 10 amostras não familiares. A combinação de Silhueta e GEIs obteve uma precisão de 73%, destacando-se como uma estratégia eficaz, possivelmente capitalizando as vantagens de ambas as representações.

Em relação a imagens adicionais ao *dataset* secundário, o modelo registrou uma precisão de 54%, indicando uma performance ruim na identificação de padrões de marcha. Portanto, mostrando-se ineficaz neste tipo de anomalias ou estilo de imagens capturadas.

3.3.2 Implementação e resultados do Modelo CNN

Além da arquitetura YOLOv8 pré-treinado, desenvolveu-se uma segunda abordagem de classificação sem pré-treinamento. Os testes realizados anteriormente utilizando o YOLOv8 foram replicados no novo modelo de CNN, ao qual se utilizou as bibliotecas *TensorFlow*, *Keras*, *Scikit-learn* e *Matplotlib* no ambiente Google Colab. A Quadro 14 apresenta a arquitetura da CNN.

Quadro 14 – Arquitetura da CNN

```

01 model = Sequential([
02     Conv2D(filters=32, kernel_size=(3, 3), padding='same',
03         activation='relu', input_shape=(altura, largura, canais)),
04     MaxPool2D(pool_size=(2, 2), padding='same'),
05
06     Conv2D(filters=64, kernel_size=(3, 3), padding='same', activation='relu'),
07     MaxPool2D(pool_size=(2, 2), padding='same'),
08
09     Conv2D(filters=128, kernel_size=(3, 3), padding='same', activation='relu'),
10     MaxPool2D(pool_size=(2, 2), padding='same'),
11
12     Conv2D(filters=256, kernel_size=(3, 3), padding='same', activation='relu'),
13     MaxPool2D(pool_size=(2, 2), padding='same'),
14
15     Flatten(),
16     Dense(1024, activation='relu'),
17     Dropout(0.2),
18     Dense(2, activation='sigmoid')
19 ])

```

Fonte: elaborado pela autora.

A arquitetura do Quadro 14 representa um modelo de rede neural convolucional desenvolvida com a biblioteca Keras, utilizando a interface *TensorFlow*. O modelo é composto por diversas camadas, começando com uma *Conv2D* inicial que possui 32 filtros, cada um com um *kernel* de tamanho (3, 3), aplicado à entrada com uma ativação *ReLU*. Após cada camada convolucional, uma camada *MaxPool2D* é inserida para reduzir a dimensionalidade espacial da representação, minimizando o risco de *overfitting*.

Esse padrão de convolução, seguido por *maxpooling*, é repetido em camadas subsequentes, aumentando gradualmente o número de filtros para 64, 128 e 256. Essa progressão permite que a rede aprenda características hierárquicas cada vez mais complexas. Após a última camada de *maxpooling*, a camada *Flatten* é empregada para transformar a saída tridimensional em um vetor unidimensional, preparando os dados para as camadas densas.

Em seguida, uma camada densa com 1024 neurônios e ativação *ReLU* é incorporada para aprender relações mais complexas entre as características extraídas. Para evitar *overfitting*, uma camada *Dropout* é aplicada, descartando aleatoriamente 20% dos neurônios durante o treinamento. Finalmente, a última camada densa possui 2 neurônios com uma função de ativação *sigmoid*, indicando que o modelo está sendo treinado para uma tarefa de classificação de 2 classes: *Normal* e *Abnormal Gait*.

A partir da definição da CNN, realizou-se a separação da base de dados em duas classes binárias: "*Normal Gait*" (0) e "*Abnormal Gait*" (1), correspondentes às imagens de marcha normal e hemiplégica, respectivamente. A divisão dos dados foi realizada em 80% para treinamento e 20% para teste, utilizando a função *train_test_split* da biblioteca do *Scikit-learn*, conforme mostra o Quadro 15.

Quadro 15 – Divisão de dados de treinamento e testes

```

01 from sklearn.model_selection import train_test_split
02 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
03     random_state = 0)
04 print(len(X_train), len(X_test), len(y_train), len(y_test))

```

Fonte: elaborado pela autora.

O treinamento foi realizado com 10 épocas visando evitar o *overfitting* e com uma baixa função de perda, que indica o quanto o modelo está se ajustando aos dados. A Figura 12 apresenta os resultados do treinamento da CNN.

Figura 12 – Resultados do treinamento da CNN conjunto de dados principal

Epoch 1/10	100/100	[=====]	- 68s	669ms/step	- loss: 3.6458	- accuracy: 0.8713
Epoch 2/10	100/100	[=====]	- 68s	675ms/step	- loss: 0.0503	- accuracy: 0.9811
Epoch 3/10	100/100	[=====]	- 69s	694ms/step	- loss: 0.0145	- accuracy: 0.9965
Epoch 4/10	100/100	[=====]	- 71s	713ms/step	- loss: 0.0057	- accuracy: 0.9994
Epoch 5/10	100/100	[=====]	- 72s	721ms/step	- loss: 0.0028	- accuracy: 0.9997
Epoch 6/10	100/100	[=====]	- 72s	721ms/step	- loss: 0.0019	- accuracy: 1.0000
Epoch 7/10	100/100	[=====]	- 72s	719ms/step	- loss: 8.3380e-04	- accuracy: 1.0000
Epoch 8/10	100/100	[=====]	- 72s	718ms/step	- loss: 0.0019	- accuracy: 0.9994
Epoch 9/10	100/100	[=====]	- 71s	713ms/step	- loss: 6.9408e-04	- accuracy: 1.0000
Epoch 10/10	100/100	[=====]	- 71s	714ms/step	- loss: 3.6936e-04	- accuracy: 1.0000
25/25		[=====]	- 5s	203ms/step	- loss: 0.0040	- accuracy: 0.9975

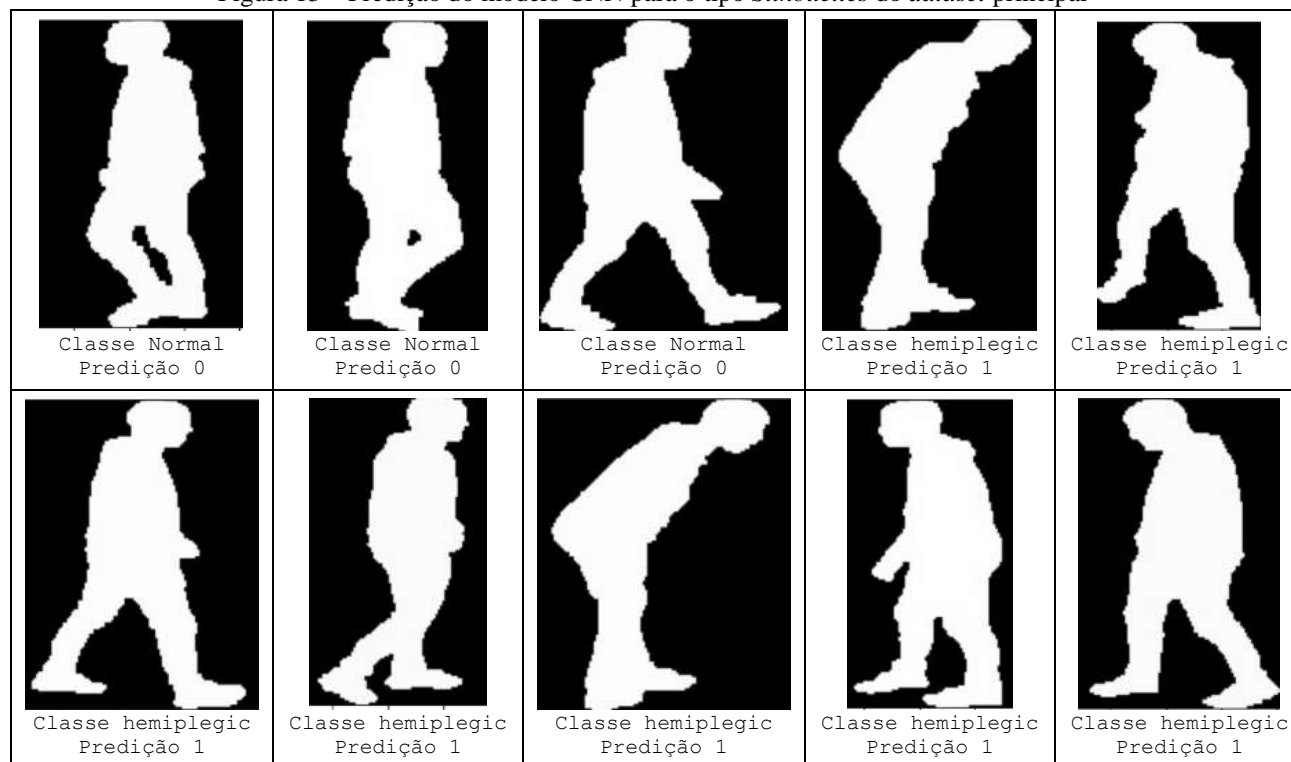
Fonte: elaborado pela autora.

Os resultados do treinamento do modelo, ao longo de 10 épocas, revelam uma evolução significativa nas métricas de perda e precisão. Inicialmente, a perda apresentou um valor relativamente alto na primeira época, indicando que o modelo não estava completamente ajustado aos dados. Contudo, nas épocas seguintes, observou-se uma rápida redução na perda, atingindo valores baixos. Esse padrão sugere uma eficiente capacidade do modelo em aprender e se adaptar aos dados de treinamento.

Quanto à precisão, a primeira época apresentou um desempenho moderado de 87,13%. Nas épocas subsequentes, entretanto, a precisão aumentou substancialmente, chegando a valores próximos ou atingindo 100%. Isso sugere que o modelo está eficazmente aprendendo os padrões presentes nos dados de treinamento e conseguindo prever corretamente as classes associadas. Além disso, a avaliação nos dados de teste, após as 10 épocas, demonstrou uma perda de 0,0040 e uma precisão de 100%. Isso sugere uma boa capacidade de generalização do modelo para dados não vistos, embora seja necessário cautela na interpretação desses resultados, garantindo que o conjunto de testes seja representativo.

A validação adicional foi realizada por meio da seleção randomizada de 10 imagens do conjunto de teste. A Figura 13 mostra os resultados obtidos após a validação de 10 imagens da pasta *Silhouettes* do conjunto de dados principal.

Figura 13 – Predição do modelo CNN para o tipo *Silhouettes* do *dataset* principal



Fonte: elaborado pela autora.

Para uma comparação direta com o modelo YOLOv8, o mesmo conjunto de dados foi utilizado (*Silhouettes*). Além disso, o *dataset* secundário, caracterizado anteriormente, foi testado no novo modelo para análise comparativa. O Quadro 16 apresenta os resultados obtidos ao longo de 10 épocas de execução.

Quadro 16 – Resultados do treinamento da CNN conjunto de dados secundário

Epoch	1/10	75/75	[=====]	-	47s	605ms/step	-	loss: 3.5357	-	accuracy: 0.6446
Epoch	2/10	75/75	[=====]	-	45s	605ms/step	-	loss: 0.2069	-	accuracy: 0.9362
Epoch	3/10	75/75	[=====]	-	45s	601ms/step	-	loss: 0.0828	-	accuracy: 0.9750
Epoch	4/10	75/75	[=====]	-	45s	600ms/step	-	loss: 0.0379	-	accuracy: 0.9892
Epoch	5/10	75/75	[=====]	-	45s	600ms/step	-	loss: 0.0250	-	accuracy: 0.9933
Epoch	6/10	75/75	[=====]	-	46s	609ms/step	-	loss: 0.0142	-	accuracy: 0.9979
Epoch	7/10	75/75	[=====]	-	45s	598ms/step	-	loss: 0.0076	-	accuracy: 0.9979
Epoch	8/10	75/75	[=====]	-	45s	601ms/step	-	loss: 0.0057	-	accuracy: 0.9992
Epoch	9/10	75/75	[=====]	-	45s	593ms/step	-	loss: 0.0084	-	accuracy: 0.9979
Epoch	10/10	75/75	[=====]	-	45s	597ms/step	-	loss: 0.0065	-	accuracy: 0.9987
19/19	[=====]	-	3s	149ms/step	-	loss: 0.2042	-	accuracy: 0.9417		

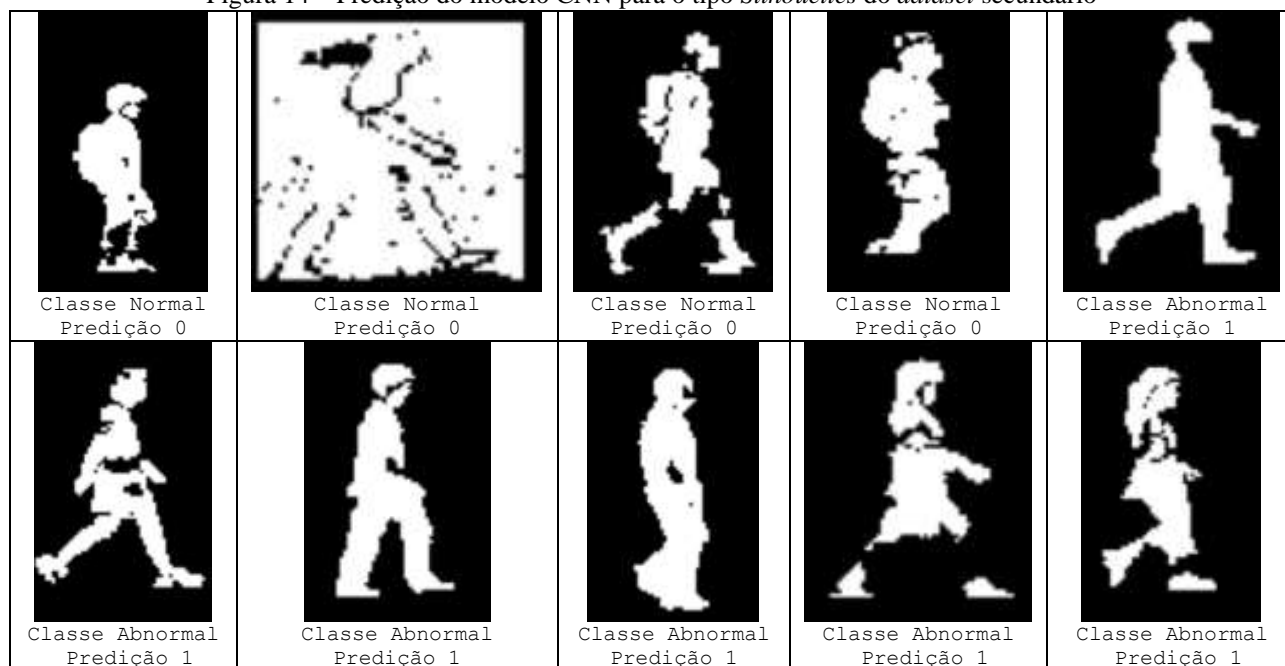
Fonte: elaborado pela autora.

Os resultados do treinamento do modelo, apresentados no Quadro 16, evidenciam um padrão significativo de aprendizado e adaptação, assemelhando-se ao comportamento observado no treinamento do conjunto de dados principal. Inicialmente, a perda apresentou um valor consideravelmente alto na primeira época, indicando uma fase inicial de ajuste do modelo. No entanto, ao longo das épocas subsequentes, observa-se uma redução acentuada da perda, atingindo valores relativamente baixos.

No que diz respeito à precisão, a primeira época registrou uma taxa de 64,46%, indicando um desempenho moderado antes do ajuste completo. Contudo, a precisão aumentou significativamente nas épocas seguintes, atingindo valores próximos ou iguais a 100%. Isso destaca que o modelo está aprendendo de maneira eficaz os padrões presentes nos dados de treinamento, sendo capaz de prever corretamente as classes associadas.

Ao ser avaliado em um conjunto de dados de teste independente após as 10 épocas, o modelo apresentou uma perda de 0.2042 e uma precisão de 94,17%. Esses resultados sugerem uma boa capacidade de generalização para dados não vistos, mantendo um desempenho elevado. A Figura 14 mostra as predições obtidas pelo modelo no qual as classes 0 e 1 representam a marcha normal e anormal respectivamente.

Figura 14 – Predição do modelo CNN para o tipo *Silhouettes* do *dataset* secundário



Fonte: elaborado pela autora.

Os resultados obtidos com o modelo CNN revelaram uma performance impressionante na classificação dos diferentes conjuntos de dados relacionados à marcha humana. Cada categoria específica foi submetida à avaliação, proporcionando uma visão abrangente da eficácia do modelo. No *dataset* de silhuetas, que representa a marcha estática, o modelo alcançou uma acurácia de 99%. Essa precisão indica a robustez do modelo na distinção entre marcha normal e hemiplégica em imagens estáticas. O Quadro 17 apresenta os resultados obtidos pelo modelo CNN nos conjuntos de dados preestabelecidos como cenário de teste.

Quadro 17 – Resultados do modelo CNN

Datasets	Treinamento (Menor valor de perda e sua acurácia)			Validação com 10 imagens não incluídas no datasets
	Valor de Perda	Acurácia (%)	Épocas	Acurácia (%)
<i>Silhouettes</i>	0.0087	99,00	10	90,00
GEIs	0.0030	100,00	10	80,00
SEIs	0.0300	98,87	10	50,00
<i>Skeletons</i>	0.0001	100,00	10	90,00
ALL (<i>Silhouettes</i> +GEIs+SEIs+ <i>Skeletons</i>)	0.0100	99,00	10	90,00
Dataset Secundário (<i>Normal</i> + <i>Abnormal</i>)	0.2070	94,00	10	100,00

Fonte: elaborado pela autora.

Para o conjunto de dados de GEIs (silhueta em movimento), a CNN atingiu uma acurácia perfeita de 100%, demonstrando uma capacidade excepcional de identificar padrões nesse contexto específico. O mesmo desempenho foi alcançado na classificação de "Skeletons" (esqueletos), indicando uma interpretação altamente eficiente das características com base na estrutura corporal. O conjunto de dados SEIs (esqueletos em movimento) apresentou uma acurácia de 98.87%, evidenciando a habilidade da CNN em lidar com padrões dinâmicos relacionados à locomoção humana.

Ao reunir esses conjuntos de dados diversos, a acurácia geral do modelo CNN foi de 99%, destacando sua capacidade de generalização para várias condições de marcha. Além da acurácia, a função de perda foi avaliada como um indicativo da eficiência do modelo. Os valores de função de perda obtidos foram notavelmente baixos em todos os conjuntos de dados, indicando uma convergência eficaz durante o treinamento.

Utilizando o *dataset* secundário no modelo CNN, que inclui as classes "*Normal*" e "*Abnormal*," a rede neural alcançou uma acurácia de 94%. Esse resultado destaca a eficiência do modelo em lidar com um conjunto de dados menor e mais específico, mantendo um desempenho notável na classificação de diferentes padrões de marcha. Além disso, a função de perda foi avaliada em 0.207, indicando uma convergência eficaz durante o treinamento. Essa medida complementa a acurácia, demonstrando que o modelo CNN não apenas classifica com precisão, mas também aprende de forma eficiente durante o processo de treinamento.

No contexto de validação, considerando 10 imagens que não incluídas ao conjunto de dados de teste, os *datasets* de *Silhouettes*, GEIs e *Skeletons* apresentaram uma alta precisão, alcançando 90%, 80% e 80%, respectivamente. Esses resultados sugerem que o modelo tem capacidade de reconhecer padrões de marcha em cenários não incluídos ao domínio do conjunto de dados. Em contraste, o *dataset* de SEIs apresentou um desempenho inferior, atingindo uma precisão de 50%, indicando uma possível limitação na generalização para dados não vistos para esta classe.

Por fim, tais resultados demonstram que o modelo CNN foi altamente bem-sucedido na classificação de diferentes tipos de marcha, apresentando uma performance excepcional em comparação com o modelo YOLOv8 pré-treinado no Roboflow.

4 CONCLUSÕES

A cinemática da marcha, que se dedica ao estudo dos movimentos específicos durante a caminhada, emerge como uma ferramenta essencial para desvendar os padrões anormais que podem surgir. Essa investigação torna-se fundamental para uma compreensão abrangente dos desafios locomotores enfrentados por indivíduos em condições adversas. Neste estudo, foi implementado dois modelos, YOLOv8 pré-treinado no Roboflow e CNN, para detectar anomalias na marcha, com ênfase na marcha hemiplégica. No qual, ambos os modelos proporcionaram uma análise profunda, revelando nuances importantes nos movimentos durante a caminhada, especialmente em cenários de hemiplegia.

Os resultados do YOLOv8 oferecem uma análise abrangente de sua capacidade de classificação em diversos conjuntos de dados. Destaca-se a distinção entre marcha normal e hemiplegia, alcançando 82% de acurácia no conjunto de *Silhouettes*. Entretanto, ao lidar com a complexidade da marcha em movimento, apresentou uma acurácia de 63% no conjunto de GEIs. A combinação desses conjuntos de dados resultou em uma acurácia global de 70%, sinalizando a habilidade do YOLOv8 em generalizar para uma variedade de condições de marcha, apesar da diminuição de desempenho observada no *dataset* secundário "*Normal*" e "*Abnormal*," no qual a acurácia foi de 54%. Embora o YOLOv8 tenha demonstrado uma sólida performance, a sensibilidade do modelo a conjuntos de dados específicos, como evidenciado no *dataset* principal, ressalta a importância de considerar cuidadosamente as características do conjunto de dados ao escolher esta abordagem.

O modelo CNN apresentou precisão notável em todos os conjuntos de dados, especialmente com o conjunto secundário "Normal" e "Abnormal", atingindo 94%. Apesar de demandar maior tempo de execução e do investimento em recursos computacionais mais elevados, a CNN revelou uma convergência eficiente com acurácias próximas a 100% e a função de perda obteve valores excepcionalmente baixos em todos os conjuntos de dados. A escolha da CNN como modelo preferencial em situações que demandam precisão extrema é respaldada não apenas pelos altos índices de acurácia, mas também pela sua habilidade de generalização para diferentes condições de marcha.

Ao contrastar os modelos desenvolvidos, YOLOv8 e CNN, com os trabalhos correlatos, percebe-se distintas abordagens e aplicações na análise da marcha humana. Com relação ao estudo de Albuquerque *et al.* (2021) que buscou classificar diferentes patologias utilizando CNN, LDA, RNNs e LSTM no qual alcançaram uma precisão de 93,4%, o YOLOv8 destacou-se na marcha estática com 82% de acurácia, enquanto a CNN superou, especialmente com 100% nos *datasets* "GEIs" e "SEIs". Sheth *et al.* (2023) focaram na identificação biométrica através da marcha, utilizando exclusivamente CNN. Os autores relatam uma acurácia acima de 90%, destacando o potencial para aplicações em reconhecimento remoto e segurança. Mukhtar *et al.* (2018) direcionaram seus esforços à detecção automatizada de pessoas com deficiência física, utilizando YOLO, CNN e GNN. A análise manual sugere informações suficientes na marcha, embora a precisão tenha sido impactada negativamente pelo erro de localização do YOLO. Comparativamente, os modelos desenvolvidos neste estudo ofereceram desempenho melhor, especialmente a CNN com acurácias próximas a 100%.

Em relação aos objetivos específicos inicialmente estabelecidos, foi possível alcançar cada um deles de maneira satisfatória. Primeiramente, a estruturação de dois *datasets* com imagens de distúrbios motores proporcionou uma base robusta para o treinamento dos modelos. Em seguida, a identificação de padrões temporais de cinemática da marcha a partir de imagens foi realizada com sucesso, utilizando eficazmente redes neurais convolucionais e o modelo YOLOv8. A etapa subsequente envolveu a classificação da marcha conforme seu tipo (normal, hemiplégica ou abnormal), no qual os modelos alcançaram uma alta precisão dentro dos cenários de testes estabelecidos. Quanto a avaliação da eficácia dos modelos em comparação com métodos tradicionais de análise da cinemática da marcha revelou resultados promissores. Embora o YOLOv8 tenha apresentado sensibilidade a conjuntos de dados específicos, especialmente no cenário "Normal" e "Abnormal," sua capacidade de generalização para diferentes condições de marcha pode ser considerada razoável. A CNN, por sua vez, destacou-se pela precisão em vários conjuntos de dados, indicando sua eficácia na classificação de padrões de marcha.

Em síntese, ambos os modelos desenvolvidos, o YOLOv8 e a CNN, tiveram um desempenho superior na classificação da marcha, um adendo para o modelo de CNN que superou as expectativas e obteve uma acurácia maior que os trabalhos correlatos. No entanto, para trabalhos futuros, sugere-se estender a aplicabilidade dos modelos desenvolvidos em cenários práticos. Uma direção promissora seria a implementação dos modelos para detecção em tempo real, permitindo sua utilização em situações do mundo real. Essa abordagem tornaria os modelos mais viáveis para aplicações práticas, especialmente na detecção de padrões de marcha hemiplégica em ambientes dinâmicos.

REFERÊNCIAS

- ALBUQUERQUE, Pedro; MACHADO, João Pedro; VERLEKAR, Tanmay Tulsidas; CORREIA, Paulo Lobato; SOARES, Luís Ducla. Remote Gait Type Classification System Using Markerless 2D Video. **Diagnosics**, [S.l.], v. 11, n. 10, p. 1824, 2 out. 2021. MDPI AG. <http://dx.doi.org/10.3390/diagnostics11101824>.
- ALPAYDIN, Ethem. **Introduction to Machine Learning**. Cambridge: Cambridge University Press (CUP), 2012.
- CAPARELLI, Thiago Bruno et al. **Reconstrução de Parâmetros Biomecânicos da Marcha por meio de Ciclogramas e Redes Neurais Artificiais**. 2017. Tese de doutorado na área de concentração Processamento da Informação, Universidade Federal de Uberlândia, Uberlândia. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/20879/3/Reconstru%C3%A7%C3%A3oPar%C3%A2metrosBiomec%C3%A2nicos.pdf>. Acesso em: 11 nov. 2023.
- CUNHA, Leonardo C. **Redes neurais convolucionais e segmentação de imagens: uma revisão bibliográfica**. 2020. 55 f. Monografia (Grau de Engenheiro de Controle de Automação) - Universidade Federal de Ouro Preto, Ouro Preto, 2020.
- GAMBLE, J. G.; ROSE, J. In: ROSE, J.; GAMBLE, J. G. **Human Walking**. 3. ed. Philadelphia: Lippincott William e Wilkins, 2006
- JOCHER, Glenn. **Ultralytics YOLOv8 Docs**. 2023. Disponível em: <https://docs.ultralytics.com/>. Acesso em: 03 dez. 2023.
- KIM, J. H. et al. Effect of gait training with functional electrical stimulation on gait capacity in stroke survivors with chronic hemiplegia: a randomized controlled trial. **Annals of Rehabilitation Medicine**, v. 42, n. 4, p. 560-571, 2018.
- LUO, Guoliang. et al. Random forest-based classification and analysis of hemiplegia gait using low-cost depth cameras. **Medical & Biological Engineering & Computing**, [S.l.], v. 58, n. 2, p. 373-382, 18 dez. 2019. Springer Science and Business Media LLC.
- MUKHTAR, Amir; CREE, Michael J.; SCOTT, Jonathan B.; STREETER, Lee. Gait Analysis of Pedestrians with the Aim of Detecting Disabled People. **Applied Mechanics And Materials**, [S.L.], v. 884, p. 105-112, ago. 2018. Trans Tech Publications, Lt. <http://dx.doi.org/10.4028/www.scientific.net/amm.884.105>.

- OLAH, Christopher. Understanding LSTM Networks. **Colah's Blog**, San Francisco, 27 ago. 2015. Disponível em: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 19 nov. 2023.
- OLNEY, Sandra J.; RICHARDS, Carol. Hemiparetic gait following stroke. Part I: characteristics. **Gait & Posture**, [S.L.], v. 4, n. 2, p. 136-148, 9 dez. 1999. Elsevier BV. [http://dx.doi.org/10.1016/0966-6362\(96\)01063-6](http://dx.doi.org/10.1016/0966-6362(96)01063-6).
- PERRY J. **Análise de Marcha: Marcha normal**. São Paulo: Manole, 2005.
- PERRY, Jacquelin; BURNFIELD, Judith M.; CABICO, Lydia M. **Gait Analysis: normal and pathological function**. 2. ed. Nebraska: Slack, 2010. 551 p.
- RACIC, V.; PAVIC, A.; BROWNJOHN, J.M.W. **Experimental identification and analytical modeling of human walking forces: Literature review**. Journal of Sound and Vibration, v,326,p.1–49, 2009.
- RIBEIRO, Edivaine G. **Rede neural convolucional aplicada ao reconhecimento de passagens de nível clandestinas em Ferrovias**. 2020. 59 f. Monografia (Especialista em Sistemas Inteligentes Aplicados a Automação) - Instituto Federal do Espírito Santo, Vitória, 2020.
- SECRETARIA DE ATENÇÃO À SAÚDE. **Diretrizes de Atenção à Reabilitação de Pessoa com Acidente Vascular Cerebral**. Ministério da Saúde, Brasília - DF, p. 3-74, 18 nov. 2022. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/s/saude-da-pessoa-com-deficiencia/publicacoes/diretrizes-de-atencao-a-reabilitacao-da-pessoa-com-acidente-vascular-cerebral.pdf/view>. Acesso em: 07 dez. 2023.
- MINISTÉRIO DA SAÚDE. **POLÍTICA NACIONAL DE SAÚDE DA PESSOA COM DEFICIÊNCIA**. Ministério da Saúde, Brasília - DF, 2017. Disponível em: <https://bvsms.saude.gov.br/bvs/saudelegis/gm/2017/MatrizesConsolidacao/comum/37518.html>. Acesso em: 07 dez. 2023.
- MISHRA, M. **Convolutional Neural Networks, Explained**, [s.l], 2020. Disponível em: <<http://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>>. Acesso em: 08 dez. 2023.
- SHETH, Abhishek Arun; SHARATH, Meghana; REDDY, Avula Sai Charan; K, Sindhu. Gait Recognition Using Convolutional Neural Network. **International Journal Of Online And Biomedical Engineering (Ijoe)**, [S.L.], v. 19, n. 01, p. 107-118, 17 jan. 2023. International Association of Online Engineering (IAOE). <http://dx.doi.org/10.3991/ijoe.v19i01.33823>.
- SOARES, Gustavo R.; CARMO, Alisson F. C. Viabilidade de detecção de câncer de mama através de rede neural convolucional em mamografias. **ETIC-ENCONTRO DE INICIAÇÃO CIENTÍFICA-ISSN 21-76-8498**, [S. l.], v. 16, n. 16, p. 1 - 15, set. 2020.
- SOLAWETZ, Jacob; ZUPPICHINI, Francesco Saverio. **What is YOLOv8?: The Ultimate Guide**. 2023. Disponível em: <https://blog.roboflow.com/whats-new-in-yolov8/>. Acesso em: 03 dez. 2023.
- VAIDYA, P. et al. A Novel Dog Breed Identification using Convolutional Neural Network. **PriMera Scientific Engineering**, [s.l.], v. 2, n. 1, p. 16-21, dez. 2022.