Aplicação de Modelos de Aprendizado Profundo na Estimativa de Relações Espaciais dos Objetos para Auxiliar Pessoas com Deficiência Visual

Aline Elí Gassenn[1], Marcelo Chamy Machado[1], Eulanda Miranda dos Santos[2]

[1]Instituto Federal do Amazonas (IFAM)
CEP 69086-475 – Manaus – AM – Brasil

[2]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
CEP 69077-000 – Manaus – AM – Brasil

aline.gassenn@gmail.com
marcelo.chamy@ifam.edu.br
emsantos@icomp.ufam.edu.br

Abstract. *In this paper, we explore computer vision and machine learning to develop an assistive algorithm for visually impaired people. Despite progress in assistive technologies, the literature revelas significant gaps in integrating real time object detection and depth estimation. The proposed methodology uses two pre-trained models, one for object detection (YOLO) and the other for depth estimation (MiDaS). The presented algorithm is able to interpret monocular images, informing the spatial relationship between detected objects and outputting text by sound. The analysis of performance considers the combination of different architectures in CPU and GPU, demonstratrating potential to improve the quality of life for visually impaired people.*

Resumo. *Neste artigo explora-se o uso de visão computacional e aprendizado de máquina no desenvolvimento de um algoritmo assistivo para pessoas com deficiência visual. Apesar dos progressos recentes em tecnologias assistivas, a literatura revela lacunas significativas na integração de detecção de objetos e estimativa de profundidade em tempo real. A metodologia empregada neste estudo utiliza dois modelos pré-treinados: um para detecção de objetos (YOLO) e outro para estimativa de profundidade (MiDaS). O algoritmo desenvolvido é capaz de processar imagens monoculares e de fornecer informações sobre as relações espaciais entre os objetos detectados, além de integrar a saída de texto a alertas sonoros. A avaliação de desempenho aborda a eficiência da combinação dessas arquiteturas em ambientes que operam tanto em CPU quanto em GPU, demonstrando o potencial desta abordagem para melhorar a qualidade de vida de indivíduos com deficiência visual.*

1. Introdução

A visão computacional, aliada ao aprendizado de máquina, tem viabilizado uma vasta gama de aplicações práticas, especialmente no desenvolvimento de tecnologias assistivas para pessoas com deficiência visual [Ming et al. 2021]. A implementação dessas tecnologias é de suma importância, uma vez que a ausência de assistência visual adequada os limita ao uso de sentidos complementares, como o tato, para a navegação e identificação

de objetos [Zafar et al. 2022]. Essa dependência pode resultar em situações de risco, especialmente ao manusear objetos cortantes ou mal posicionados [Vijetha 2024].

Neste estudo, explora-se a integração de técnicas avançadas de detecção de objetos e estimativa de profundidade, utilizando imagens monoculares processadas em tempo real e sem a dependência de internet e câmeras robustas. Foram utilizados os modelos YOLOv8 (*You Only Look Once version 8*) e MiDaS, os quais permitem a identificação de objetos e a avaliação de suas relações espaciais em uma cena. Esta metodologia demonstra aplicabilidade tanto em ambientes internos quanto externos, mediante condições de iluminação adequadas, sejam naturais ou artificiais.

Portanto, o propósito central desta pesquisa consiste na exploração da integração de modelos de redes neurais profundas no processamento de imagens. Esta integração visa estabelecer uma compreensão das relações espaciais entre objetos identificados e determinar as suas posições relativas. De maneira mais específica, este trabalho propõe um mecanismo de suporte destinado a indivíduos com limitações visuais, que poderia melhorar a autonomia e a percepção espacial de pessoas com deficiência visual, através do uso do método proposto para interpretar o ambiente circundante.

No decorrer deste trabalho, serão descritas as metodologias adotadas, os desafios encontrados e as estratégias utilizadas, culminando em uma visão sobre as futuras direções e melhorias que podem ser incorporadas à solução proposta.

2. Trabalhos Relacionados

O uso de técnicas de aprendizado profundo para detecção de objetos e estimativa de profundidade tem sido amplamente reconhecido, especialmente pelo seu potencial de assistência a pessoas com deficiência visual.

[Jadon et al. 2023] desenvolveram um algoritmo para auxiliar pessoas com deficiência visual, integrando tecnologias de Internet das Coisas, *Blockchain* e Aprendizado Profundo. Foi utilizada uma câmera acoplada a um Raspberry Pi, com algoritmo processando 30 quadros por segundo e respondendo em dois segundos mediante condições ideais, identificando obstáculos e emitindo notificações de áudio para o usuário. Foram empregados os modelos pré-treinados YOLOv7 e MiDaS, que atingiram uma precisão média de 0,69 e um Erro Absoluto Relativo de 0,0732. Os autores destacam várias limitações, incluindo a escalabilidade em diferentes ambientes, a necessidade de *hardware* específico como Raspberry Pi e GPUs, e a exigência de uma conexão estável à internet para o processamento em nuvem.

Em um estudo recente, [Vijetha 2024] desenvolveram um aplicativo móvel para auxiliar na navegação de pessoas com deficiência visual, utilizando modelos de inteligência artificial pré-treinados: TopFormer para segmentação semântica e MiDaS para estimativa de profundidade. O aplicativo detecta obstáculos e sugere rotas livres. Os resultados mostraram acurácia de 0,83 a 0,86 e taxas de falsos positivos de 0,04 a 0,11, com base nos conjuntos de dados DIODE e NYU v2, respectivamente. Uma limitação observada foi a dificuldade do modelo de segmentação em identificar superfícies brilhantes, tais como pisos, causando alertas falsos. O estudo também destacou a necessidade de melhorar o *feedback* do sistema, especialmente em ambientes com alta densidade de obstáculos.

[Bauer et al. 2020] implementaram um algoritmo que combina aprendizado profundo com sensores vestíveis de baixo custo, visando auxiliar pessoas com deficiência visual em ambientes urbanos. O algoritmo é responsável por identificar objetos em imagens, e identificar a localização do mesmo em três áreas: direita, centro e esquerda. A informação é transmitida ao usuário por meio de alertas auditivos e táteis por meio de fones de ouvido e dois *smartwatches*. O algoritmo, que processa 3,8 quadros por segundo, utiliza os modelos YOLOv2 e uma ResNet50, alcançando uma precisão média de 0,742 no conjunto de dados PASCAL VOC e um erro médio quadrático de 1,344 no dataset NYU para estimativas de profundidade. Os pesquisadores identificaram limitações significativas, incluindo imprecisões na estimativa de profundidade e localização de objetos, confusão entre pessoas reais e em imagens de publicidade, e desafios causados pelo uso de equipamentos pesados e intrusivos em ambientes externos.

[Masoumian et al. 2021] desenvolveram um algoritmo que opera com duas redes neurais em paralelo para determinar a distância absoluta dos objetos a partir de uma única imagem. Este sistema combina o modelo YOLOv5, utilizado para a detecção de objetos, com a DepthNet, uma rede autocodificadora profunda, para a estimativa de profundidade. A calibração, crucial para a obtenção das medidas de distância absoluta e avaliação do desempenho, representa um ponto crítico do algoritmo. Os autores avaliaram a precisão do sistema comparando as distâncias calculadas pelo modelo com medições manuais em um conjunto de dados privado que inclui 100 imagens. O algoritmo demonstrou uma taxa de acerto de 96% e um erro médio quadrático de 0,406.

[Wang et al. 2021] desenvolveram um algoritmo de detecção de objetos e estimativa de profundidade em tempo real, utilizando duas redes neurais convolucionais ajustadas: RefineDet para detecção de objetos e MonoDepth para estimativa de profundidade. O algoritmo opera em paralelo e utiliza uma câmera estéreo para identificar objetos em imagens, seguido pela geração de imagens estéreo e mapas de profundidade mediante cálculos de geometria epipolar. Nos testes realizados com o conjunto de dados PASCAL VOC, o algoritmo atingiu uma precisão média de 0,7975 e um Erro Absoluto Relativo de 0,08, mantendo uma eficiência de 25 quadros por segundo. Entre as limitações identificadas estão a dificuldade em detectar objetos pequenos, o alto custo computacional, desafios na estimativa de profundidade usando visão estéreo e limitações na precisão das estimativas de profundidade em abordagens de treinamento não supervisionado.

[Won et al. 2021] exploraram a transferência de aprendizado para ajudar pessoas com deficiência visual na detecção de objetos. Os autores utilizaram os modelos SSD MobileNet e *Faster* R-CNN e uma câmera Logitech C920. O algoritmo fornece um sinal auditivo ao usuário sobre os objetos identificados. Avaliado no conjunto de dados COCO, o *Faster* R-CNN obteve uma precisão média de 0,8961 e um tempo de inferência de 7,2 segundos, enquanto o modelo SSD MobileNet alcançou uma precisão de 0,5708 e um tempo de inferência de 1,8 segundos.

No Quadro 1, podemos observar um resumo comparativo dos trabalhos relacionados, abordando aspectos como autor, modelos utilizados para a detecção de objetos e estimativa de profundidade, métricas adotadas e tempo de inferência do algoritmo.

Nesta pesquisa, optou-se por uma abordagem semelhante à proposta por Jadon et al. (2023), porém restringindo o escopo para uma implementação local, essa delimitação

Quadro 1. Relação entre os trabalhos de detecção e profundidade.

Artigo	Modelo detecção	Modelo profundidade	Métrica detecção	Métrica profundidade	Tempo de inferência
[Jadon et al. 2023]	YOLOv7	MiDaS	mAP 0,69	AbsRel 0,0732	2s
[Vijetha 2024]	TopFormer	MiDaS	Acurácia 0,83	N/D	0,2s
[Bauer et al. 2020]	YOLOv2	ResNet50	mAP 0,87	RMSE 0,672	0,26s
[Masoumian et al. 2021]	YOLOv5	DepthNet	Acurácia 0,96	RMSE 0,203	N/D
[Wang et al. 2021]	RefineDet	MonoDepth	mAP 0,79	AbsRel 0,08	0,04s
[Won et al. 2021]	SSD	N/D	mAP 0,57	N/D	1,8s
Este trabalho	**YOLOv8**	**MiDaS**	**mAP 0,52**	**AbsRel 0,116**	**0,029s**

viabiliza o funcionamento da aplicação em áreas com limitações de conectividade à internet, facilitando a mobilidade dos usuários em espaços sem acesso à rede. Assim como os autores, neste trabalho também será utilizado o modelo MiDaS [Ranftl et al. 2022], que tem sido frequentemente citado em publicações acadêmicas para a criação de mapas de profundidade, com resultados satisfatórios [Birkl et al. 2023] [Izadmehr et al. 2022]. Também foi adotado o modelo YOLO, na versão 8, devido à sua implementação direta e por ser a versão mais recente disponível no decorrer da realização dos experimentos. Semelhantemente ao proposto por Bauer et al. (2019), foram implementados alertas auditivos espaciais, informando ao usuário a posição dos objetos em relação à câmera – direita, centro, esquerda, bem como sua posição relativa.

3. Metodologia

Nesta seção, será descrita a metodologia para a construção da solução, com o detalhamento da implementação de modelos pré-treinados, seguida pela estratégia definida para a conversão da saída do modelo em áudio e o método de avaliação dos modelos.

3.1. Implementação de Modelos Pré-Treinados

O uso de modelos pré-treinados é uma estratégia adotada que se refere às arquiteturas de redes neurais já treinadas em um ou mais conjuntos de dados. Esses modelos são comumente disponibilizados ao público sob determinadas licenças, como a MIT *License*, que permite a reutilização do código para qualquer finalidade com a inclusão da licença e dos direitos autorais, e a GPU *General Public License* (GPL), que permite a redistribuição e modificação, exigindo que as alterações sejam abertas e distribuídas sob a mesma licença.

Especificamente, os modelos YOLO disponíveis foram treinados usando o conjunto de dados COCO e estão acessíveis ao público. Conhecidos como modelos pré-treinados, eles possibilitam a realização de inferência em imagens, limitados apenas à detecção de objetos que foram incluídos no conjunto de dados usado para o treinamento. Além disso, esses modelos podem servir como base para treinamentos em conjuntos de dados alternativos. Os modelos pré-treinados já incorporam conhecimentos essenciais sobre imagens, como contornos, cores e texturas. Portanto, a utilização desses modelos

em um novo conjunto de dados pode ser mais eficiente, dada a base de conhecimento pré-existente sobre a extração de características visuais.

No contexto da criação de algoritmos de detecção de objetos para auxiliar pessoas com deficiência visual, a utilização de modelos pré-treinados é vantajosa devido, em parte, à possibilidade de mitigar o oneroso processo de coleta e anotação de dados, bem como evitar o treinamento intensivo que frequentemente requer recursos computacionais substanciais.

Para a detecção de objetos, foram utilizadas as versões pré-treinadas do modelo YOLOv8. Os limiares de confiança e de Intersecção sobre a União (IoU) foram ajustados para 0,4 e 0,45, respectivamente, com base nos testes realizados. Para determinar a localização espacial dos objetos, à esquerda, ao centro ou à direita, dividiu-se a dimensão horizontal da imagem em três segmentos de igual tamanho e se utilizou como base a coordenada central da caixa delimitadora.

Para a tarefa de estimativa de profundidade, optou-se pelo modelo MiDaS, dada a sua eficácia reconhecida e a disponibilidade de arquiteturas pré-treinadas. A visualização do mapa de profundidade é determinada através da normalização e ajustes dos valores para uma escala que varia de 0 a 1. O limite superior dessa escala é estabelecido com base no valor mais alto encontrado no mapa original. Estabeleceu-se um limiar de confiança de 0,43 para distinguir objetos próximos dos distantes, com base em observações empíricas.

No algoritmo proposto, enfatiza-se o processamento de imagens monoculares, sejam provenientes de uma câmera, fotografia ou vídeo, para a realização simultânea da detecção de objetos e da estimativa de profundidade. Esse procedimento é executado sobre a mesma imagem, resultando em dois conjuntos de dados complementares. Posteriormente, esses resultados são unificados, de forma que a lista de objetos identificados também inclui informações relativas à sua profundidade na cena, destacando a capacidade do algoritmo de identificar até 80 classes de objetos.

3.2. Conversão da Saída em Áudio

Ao identificar um objeto em uma imagem, a caixa delimitadora correspondente é alinhada ao mapa de profundidade, o que permite determinar a profundidade com base no valor atribuído ao pixel central da caixa delimitadora. Com base nessas informações, um texto sintetizado é elaborado para tornar os resultados mais humanizados e de fácil interpretação para os usuários.

A configuração do texto é flexível e pode ser personalizada conforme as preferências e requisitos do usuário. Nesse estudo, propõe-se que o algoritmo elabore um texto com a seguinte configuração: “Mesa está longe à esquerda, cadeira está perto à direita”. Esta configuração foi escolhida pelos autores pois proporciona ao usuário informações claras sobre a identidade do objeto, sua distância relativa e sua localização espacial. Acredita-se que esta abordagem ofereça uma estrutura concisa e informativa.

O terceiro componente do algoritmo proposto é responsável por transformar o texto gerado em saída de áudio. Esta funcionalidade permite que pessoas com deficiência visual obtenham informações sobre os objetos em seu entorno de forma auditiva. Para essa finalidade, optou-se pela utilização da biblioteca pyttsx3 [PyPI 2021], dada a sua interface intuitiva e simplicidade de implementação. A velocidade de reprodução do áudio

foi ajustada para 80 palavras por minuto, em vez do padrão 100. Essa alteração foi realizada considerando que o sinal sonoro disponibilizado pela biblioteca está em inglês, o que pode exigir um tempo adicional para a compreensão por parte dos usuários que não são fluentes no idioma.

Na Figura 1, é possível observar o fluxo integral da abordagem sugerida. Uma imagem monocular é obtida (I) e, paralelamente, a informação visual é processada pelos modelos YOLOv8 (II) e MiDas (III). A partir desse processamento, é extraído o mapa de profundidade e a detecção de objetos, juntamente com suas respectivas taxas de confiança. Posteriormente, um limiar é utilizado para estimar se o objeto está próximo ou distante. A imagem é segmentada no eixo translacional em três partes iguais, com o objetivo de determinar a posição do objeto, seja à direita, ao centro ou à esquerda, tomando como referência o centroide do objeto (IV). Em seguida, é gerada uma saída textual que indica o objeto detectado e sua relação espacial (V). Por fim, essa saída textual é convertida em voz (VI) indicando o objeto detectado, sua posição e proximidade.

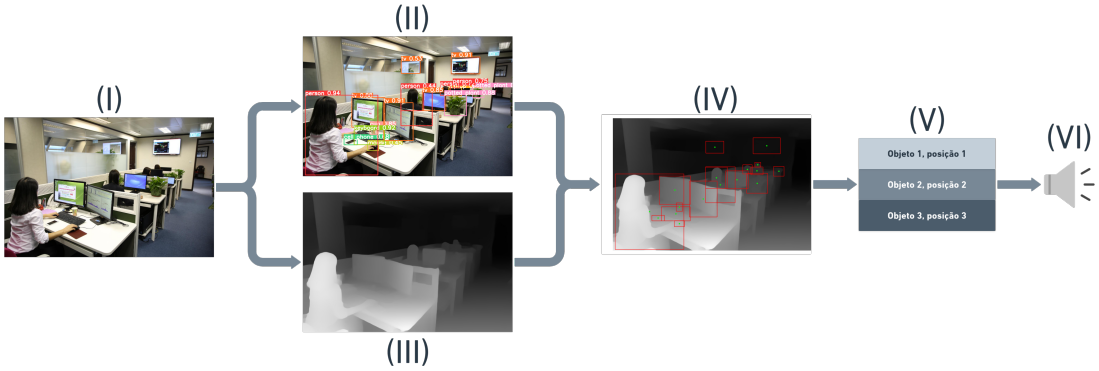

Figura 1. Fluxograma da abordagem proposta.

3.3. Avaliação dos Modelos

Para executar o processamento do algoritmo em tempo real, é de suma importância considerar não apenas a eficácia dos modelos selecionados, mas também o seu tempo de execução. Esta dualidade é essencial para conferir ao usuário um algoritmo simultaneamente ágil e eficiente, além de minimizar o tempo de reação requerido para que indivíduos com deficiência visual possam evitar acidentes ao deslocar-se ou ao interagir com objetos mal posicionados.

Nesse contexto, foram analisadas todas as arquiteturas disponíveis para cada modelo selecionado. O modelo YOLOv8 oferece as arquiteturas: *nano*, *small*, *medium*, *large* e *extra large*, já o modelo MiDaS possui as arquiteturas: *small*, *hybrid* e *large*. Para avaliar o desempenho desses modelos, foi considerada a taxa de quadros por segundo (FPS), que indica a agilidade do processamento das imagens e as métricas específicas relacionadas às tarefas de cada modelo.

Para a detecção de objetos, optou-se por usar a métrica de Precisão Média (mAP, do inglês *mean Average Precision*) em detrimento da acurácia. A escolha se justifica pelo fato que o modelo YOLO, baseado em regressão, produz múltiplas caixas delimitadoras ao estimar a confiança dos objetos em uma imagem. Devido a essa característica, o modelo em questão produz muitos verdadeiros negativos quando comparado a modelos de classificação. A métrica mAP torna-se mais adequada e confiável para avaliar o desem-

penho, uma vez que ela desconsidera os verdadeiros negativos em sua fórmula.

Quanto à estimativa de profundidade realizada pelo modelo MiDaS, a métrica adotada é o Erro Relativo Absoluto (AbsRel). Essa métrica avalia a discrepância entre as profundidades estimadas pelo modelo e as profundidades reais, estabelecendo uma relação relativa entre elas.

Ao considerar essas métricas em conjunto com a taxa de quadros por segundo, pode-se realizar uma avaliação dos modelos em relação ao seu desempenho nos recursos de *hardwares* disponíveis: GPU Nvidia RTX3060 e a CPU Intel Core i7-11800H, com as versões driver 535.104.05 e o CUDA 12.2. Para garantir a compatibilidade com os modelos selecionados, todas as imagens foram redimensionadas para os tamanhos padrão dos modelo utilizados: resolução 640x480 pixels para detecção de objetos e 256x192 pixels para a estimativa de profundidade. Deste modo, pode-se selecionar a variante de modelo mais adequada para aplicações em tempo real, assegurando que a qualidade dos resultados seja mantida, sem sobrecarregar o algoritmo.

4. Resultados

Para a avaliação dos resultados obtidos, será apresentada nesta seção a comparação das métricas obtidas de cada modelo utilizado, em seguida serão detalhadas a apresentação da saída de dados, as limitações deste trabalho e perspectivas de melhorias futuras.

4.1. Métricas dos Modelos

Neste estudo, visa-se identificar uma composição ótima de arquiteturas dedicadas tanto à detecção de objetos quanto à estimativa de profundidade, com o objetivo de alcançar um equilíbrio entre as variáveis: taxa de quadros por segundo (FPS), precisão média (mAP) do modelo de detecção de objetos e o erro relativo absoluto (AbsRel) na estimativa da profundidade. Evidentemente, a seleção do *hardware* é um fator determinante, influenciando diretamente a eficiência do algoritmo.

Os dados coletados são sintetizados no Quadro 2, e representados graficamente na Figura 2, onde se apresentam as métricas de desempenho correspondentes a todas as combinações entre as arquiteturas dos modelos selecionados. Entre essas métricas estão o tamanho do modelo – quantificado pelo número total de parâmetros, assim como o mAP, o AbsRel e o FPS, os quais foram medidos tanto em ambientes de GPU quanto de CPU.

Para o cálculo do mAP, recorreu-se à documentação oficial do modelo YOLO [Ultralytics 2023], que baseia seus resultados no conjunto de dados COCO e considera a média dos intervalos de confiança que variam de 0,5 a 0,95. O AbsRel, por sua vez, foi derivado do artigo dedicado à implementação do modelo MiDaS [Ranftl et al. 2021], cuja métrica é baseada no conjunto de dados ETH3D.

Em relação ao FPS, as medidas foram realizadas no âmbito deste estudo, utilizando o conjunto de dados PASCAL VOC 2012, um conjunto de 7.282 imagens naturais que abrange 20 categorias de objetos. Essas categorias incluem uma variedade de veículos, animais e itens domésticos, tais como carros, bicicletas, cadeiras, garrafas, cachorro e ovelha, sendo amplamente utilizado em aplicações de detecção e segmentação de objetos em imagens [Kaggle 2021].

Na Figura 2, as diferentes formas geométricas simbolizam as distintas arquitetu-

Quadro 2. Relação entre as arquiteturas dos modelos YOLOv8 e MiDaS.

Modelo de detecção	Modelo de profundidade	Tamanho (x 10^7)	mAP	AbsRel	FPS GPU	FPS CPU
nano	*small*	2,44	0,37	0,116	63,36	22,98
small	*small*	3,24	0,44	0,116	56,21	27,57
medium	*small*	4,72	0,50	0,116	42,38	23,09
large	*small*	6,50	0,52	0,116	33,95	20,03
extra large	*small*	8,95	0,53	0,116	26,49	15,82
nano	*hybrid*	12,6	0,37	0,093	9,22	0,96
small	*hybrid*	13,4	0,44	0,093	9,02	0,96
medium	*hybrid*	14,9	0,50	0,093	8,58	0,96
large	*hybrid*	16,6	0,52	0,093	8,09	0,95
extra large	*hybrid*	19,1	0,53	0,093	7,52	0,97
nano	*large*	34,7	0,37	0,089	4,54	0,41
small	*large*	35,5	0,44	0,089	4,50	0,38
medium	*large*	36,9	0,50	0,089	4,32	0,40
large	*large*	38,7	0,52	0,089	4,21	0,44
extra large	*large*	41,2	0,53	0,089	4,01	0,42

ras do modelo MiDaS: *small* (quadrado), *hybrid* (círculo) e *large* (triângulo). As cores, por sua vez, representam as arquiteturas do modelo YOLOv8: *nano* (verde), *small* (vermelho), *medium* (azul), *large* (roxo) e *extra large* (laranja). A dimensão de cada forma geométrica indica o tamanho dos modelos combinados. O posicionamento vertical (eixo y) informa o nível de precisão média (mAP), enquanto o eixo horizontal (x) demonstra o FPS. Como referência, a visão humana consegue processar de 50 a 500 quadros por segundo [Davis et al. 2015], enquanto produções cinematográficas geralmente possuem uma taxa de 24 quadros por segundo [Pazhoohi and Kingstone 2021].

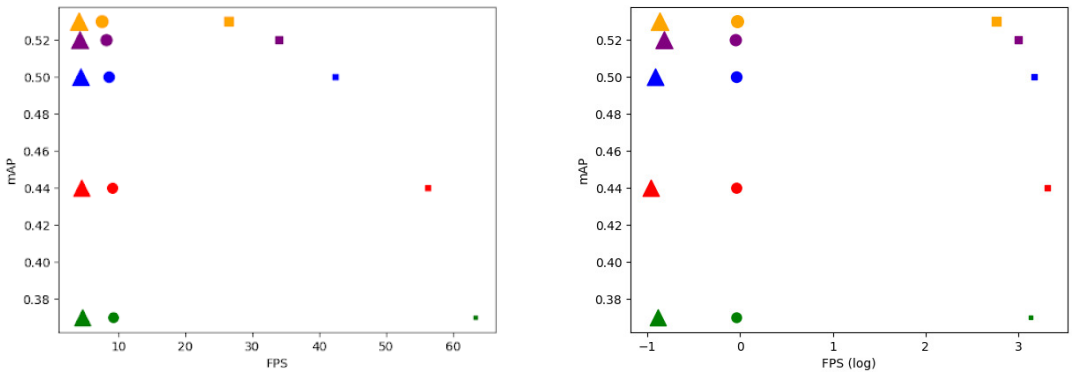

Figura 2. Comparação de desempenho entre as arquiteturas dos modelos YOLOv8 e MiDaS, em ambiente de GPU (à esquerda) e CPU (à direita).

Na análise de desempenho na GPU, os dados indicam que a combinação mais benéfica se dá pela implementação da versão *medium* do modelo YOLOv8 em conjunto com a versão *small* do modelo MiDaS. Essa configuração parece otimizar a relação entre precisão e desempenho, sem comprometer significativamente o FPS. A GPU, com a sua capacidade de paralelismo e múltiplos núcleos, é especialmente adequada para tarefas intensivas como essa, permitindo processar grandes volumes de dados simultaneamente.

Em contrapartida, ao utilizar a CPU como plataforma de processamento, a versão *small* de ambos os modelos, YOLOv8 e MiDaS, parece ser a mais adequada. Isso se deve à natureza da CPU, a qual é mais adaptada para tarefas sequenciais e pode não se beneficiar tanto do paralelismo. Além disso, a combinação da versão *extra large* do YOLOv8 com a versão *small* do MiDaS ainda mantém um bom FPS na GPU, embora haja uma queda nesse aspecto, refletindo a capacidade superior de processamento gráfico da GPU em comparação com a CPU.

Ao compararmos o sistema proposto com os trabalhos existentes na literatura, como o de [Jadon et al. 2023], que empregam os modelos YOLOv7 e MiDaS, identificamos vantagens significativas no presente estudo. Destacam-se, particularmente, a redução no tempo de inferência e uma arquitetura mais simplificada, que elimina a necessidade de processamento em nuvem. Adicionalmente, nosso sistema alcança o menor tempo de inferência quando operado em GPU (0,029 segundos) e apresenta o segundo menor tempo em processamento por CPU (0,43 segundos), utilizando a configuração ideal com o modelo YOLOv8 *large* e a versão *small* do MiDaS.

4.2. Saída de Dados

Na Figura 3, apresentam-se os desempenhos alcançados nas atividades de detecção de objetos e estimativa de profundidade, aplicadas em duas imagens distintas. Na detecção de objetos, cada item identificado é circunscrito por uma caixa delimitadora, com anotações indicando sua classe e sua confiança. Quanto à estimativa de profundidade, a informação é representada por um mapa em escala de cinza: o branco denota objetos mais próximos, enquanto o preto indica aqueles mais distantes, proporcionando uma visualização relativa das variações de profundidade.

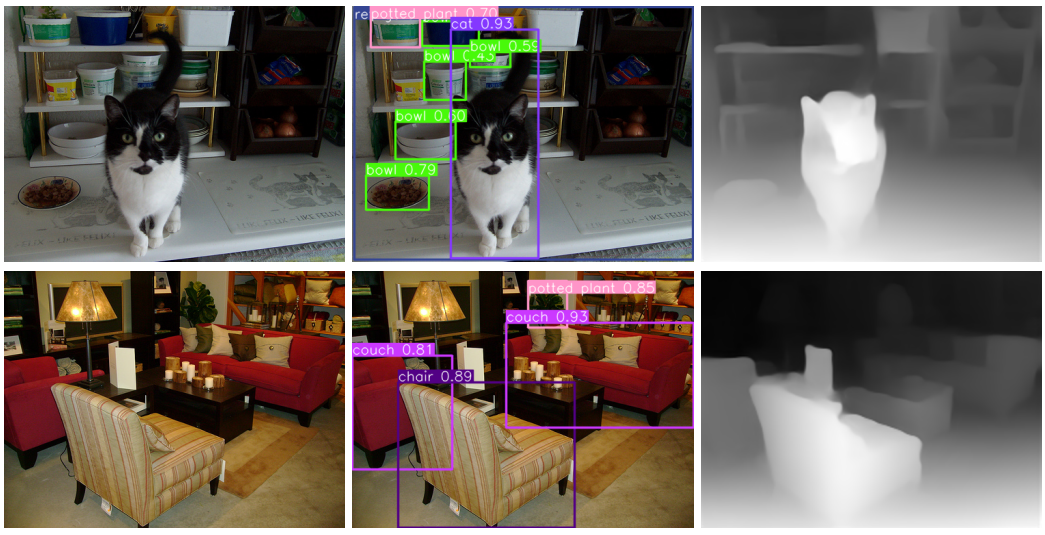

Figura 3. Imagem Original (à esquerda), a detecção de objetos (ao centro) e a estimativa de profundidade (à direita).

Os resultados obtidos são sintetizados e integrados visando estabelecer a posição espacial dos objetos na cena, identificando se estão à esquerda, ao centro ou à direita. Além disso, a análise inclui uma avaliação da profundidade relativa desses objetos em relação à posição da câmera. Utiliza-se um limiar específico para categorizar a proximidade dos objetos: para valores menores que o limiar, os objetos são considerados distantes

da câmera, enquanto valores superiores sugerem proximidade. Esta dimensão de profundidade, bem como as posições dos objetos, está elucidada com mais detalhe na Figura 4.

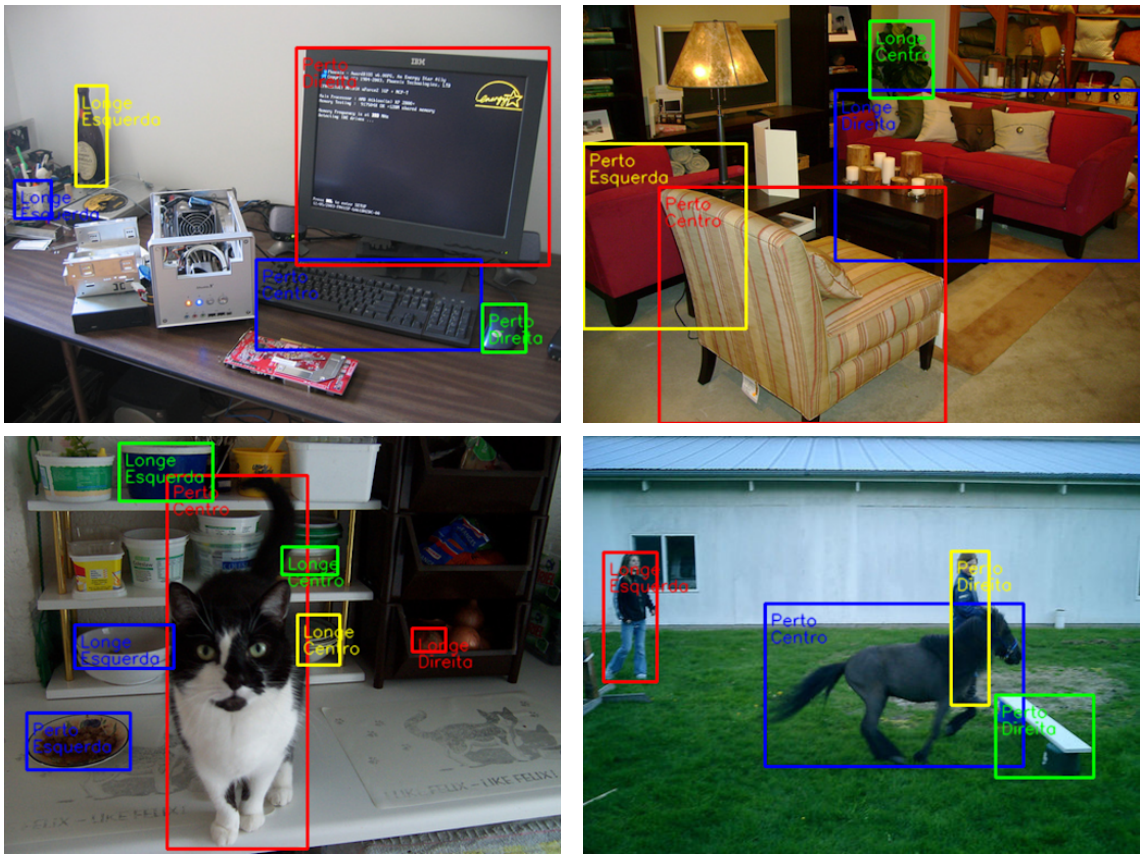

Figura 4. Representação das posições translacionais na dimensão horizontal juntamente com a sua estimativa de profundidade.

4.3. Limitações e Perspectivas Futuras

O desempenho do algoritmo é condicionado pelas limitações intrínsecas dos modelos utilizados. O modelo YOLOv8, por exemplo, apresenta limitações na detecção de objetos de pequenas dimensões, distantes, muito próximos ou em sobreposição. Similarmente, o modelo MiDaS possui dificuldades ao identificar objetos muito distantes e transparentes, tais como itens de vidro.

Adicionalmente, o algoritmo é configurado para emitir sons em intervalos de tempo predefinidos, o que pode comprometer a percepção espacial em cenários onde os objetos se movem rapidamente ou ainda identificar objetos irrelevantes, o que pode distrair o usuário. A eficácia do algoritmo também pode ser reduzida em cenários com alta densidade de objetos, resultando em omissões ou informações desatualizadas.

No entanto, uma das principais vantagens desse código é sua capacidade de operar localmente, eliminando a necessidade de conexão com a internet e preservando a privacidade do usuário, pois os dados não são armazenados em nuvem. Embora os alertas sonoros estejam disponíveis apenas em inglês, futuras implementações poderiam incluir suporte multilíngue e transformar o algoritmo em um assistente interativo, o que permitiria ao usuário fazer perguntas diretamente.

Contudo, há desafios adicionais oriundos da limitação do conjunto de dados utilizado para o treinamento dos modelos. Uma solução seria a integração de novos conjuntos de dados que contemplem obstáculos específicos, como terrenos irregulares e calçadas, visando assim ampliar a segurança dos usuários com deficiência visual.

Embora o foco deste estudo tenha sido a implementação de um algoritmo em *hardware* tradicional, o código possui potencial para otimizações específicas em dispositivos portáteis, tais como uma câmera egocêntrica ou até mesmo um *smartphone*, melhorando assim o entendimento do usuário acerca das relações espaciais entre os objetos ao seu redor.

5. Conclusão

A visão computacional, aliada à inteligência artificial, tem possibilitado avanços significativos na análise e interpretação de imagens. No contexto deste estudo, explorou-se a integração de técnicas de aprendizado de máquina de detecção de objetos e estimativa de profundidade, com foco particular em imagens monoculares. Foram utilizados os modelos YOLOv8 e MiDaS para implementar um algoritmo capaz de reconhecer até 80 classes de objetos diferentes e determinar suas relações espaciais e profundidades relativas.

Contudo, é importante salientar que o algoritmo possui algumas limitações, tais como dificuldades em identificar objetos pequenos, distantes ou sobrepostos. Além disso, o algoritmo emite alerta sonoros em intervalos fixos, o que pode afetar a percepção dos usuários em cenários de movimentação rápida, alta densidade de objetos ou ainda identificar objetos irrelevantes que distraiam o usuário.

Apesar dessas limitações, a combinação desses modelos oferece um impacto significativo, especialmente para indivíduos com deficiência visual. Ao proporcionar uma interpretação mais detalhada e dimensional do ambiente, ao emitir um sinal sonoro informando os objetos e sua distância relativa em relação à câmera, o algoritmo se torna uma ferramenta a fim de aprimorar a percepção desses indivíduos, impactando positivamente em sua qualidade de vida.

Em trabalhos futuros, algumas áreas de melhoria serão exploradas, como a adoção de técnicas de segmentação para uma estimativa de profundidade mais precisa, bem como métodos para determinar a distância absoluta dos objetos em relação à câmera. Outra questão a ser considerada é a avaliação de outras bibliotecas de voz e a utilização de modelos em versões reduzidas e mais otimizadas, implementando o algoritmo em dispositivos móveis.

6. Agradecimentos

Este artigo é o resultado do projeto de PD&I ARANOUÁ, realizado pelo IFAM, em parceria com a Samsung Eletrônica da Amazônia Ltda., usando recursos da Lei Federal nº 8.387/1991, estando sua divulgação e publicidade em conformidade com o previsto no artigo 39.º do Decreto nº 10.521/2020.

Referências

Bauer, Z., Dominguez, A., Cruz, E., Gomez-Donoso, F., Orts-Escolano, S., and Cazorla, M. (2020). Enhancing perception for the visually impaired with deep learning techniques and low-cost wearable sensors. *Pattern Recognition Letters*, 137:27–36.

Birkl, R., Wofk, D., and Müller, M. (2023). MiDaS v3.1 – A Model Zoo for Robust Monocular Relative Depth Estimation. arXiv:2307.14460 [cs].

Davis, J., Hsieh, Y.-H., and Lee, H.-C. (2015). Humans perceive flicker artifacts at 500 Hz. *Scientific Reports*, 5(1):7861.

Izadmehr, Y., Satizábal, H. F., Aminian, K., and Perez-Uribe, A. (2022). Depth Estimation for Egocentric Rehabilitation Monitoring Using Deep Learning Algorithms. *Applied Sciences*, 12(13):6578.

Jadon, S., Taluri, S., Birthi, S., Mahesh, S., Kumar, S., Shashidhar, S. S., and Honnavalli, P. B. (2023). An Assistive Model for the Visually Impaired Integrating the Domains of IoT, Blockchain and Deep Learning. *Symmetry*, 15(9):1627.

Kaggle (2021). PASCAL VOC 2012 Dataset. Disponível em: <https://www.kaggle.com/datasets/gopalbhattrai/pascal-voc-2012-dataset>. Acesso em: 05 de julho de 2023.

Masoumian, A., Marei, D. G. F., Abdulwahab, S., Cristiano, J., Puig, D., and Rashwan, H. A. (2021). Absolute distance prediction based on deep learning object detection and monocular depth estimation models. arXiv:2111.01715 [cs].

Ming, Y., Meng, X., Fan, C., and Yu, H. (2021). Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33.

Pazhoohi, F. and Kingstone, A. (2021). The Effect of Movie Frame Rate on Viewer Preference: An EyeTracking Study. *Augmented Human Research*, 6(1):2.

PyPI (2021). pyttsx3: Text to Speech (TTS) library for Python 2 and 3. Disponível em: <https://github.com/nateshmbhat/pyttsx3>. Acesso em: 28 de julho de 2023.

Ranftl, R., Bochkovskiy, A., and Koltun, V. (2021). Vision Transformers for Dense Prediction. arXiv:2103.13413 [cs].

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. (2022). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3).

Ultralytics (2023). You Only Live Once (YOLO). Disponível em: <https://github.com/ultralytics/ultralytics>. Acesso em: 20 de junho de 2023.

Vijetha, U., G. V. (2024). Obs-tackle: an obstacle detection system to assist navigation of visually impaired using smartphones. *Machine Vision and Applications*, 35(20):1–19.

Wang, H.-M., Lin, H.-Y., and Chang, C.-C. (2021). Object Detection and Depth Estimation Approach Based on Deep Convolutional Neural Networks. *Sensors*, 21(14):4755.

Won, W.-C., Yong, Y.-L., and Khor, K.-C. (2021). Object Detection and Recognition for Visually Impaired Users: A Transfer Learning Approach. In *2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, pages 1–6, IPOH, Malaysia. IEEE.

Zafar, S., Asif, M., Ahmad, M. B., Ghazal, T. M., Faiz, T., Ahmad, M., and Khan, M. A. (2022). Assistive Devices Analysis for Visually Impaired Persons: A Review on Taxonomy. *IEEE Access*, 10:13354–13366.