UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Desenvolvimento de um sistema de navegação e orientação para pessoas com baixa visão utilizando inteligência artificial

Antonio de Sousa Leitão Filho

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Antonio de Sousa Leitão Filho

Desenvolvimento de um sistema de navegação e orientação para pessoas com baixa visão utilizando inteligência artificial

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial e Big Data

Orientador: Prof. Dr. Valdir Grassi Junior

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Leitão Filho, Antonio de Sousa
L533d Desenvolvimento de um sistema de navegação e orientação para pessoas com baixa visão utilizando inteligência artificial / Antonio de Sousa Leitão Filho; orientador Valdir Grassi Junior. -- São Carlos, 2023.
64 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Visão computacional. 2. Síntese de fala. 3. Tecnologia assistiva. I. Grassi Junior, Valdir, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Antonio de Sousa Leitão Filho

Navigation and orientation system based on artificial inteligence developed to aid persons with low vision

Monograph presented to the Department of Computer Sciences of the Institute of Mathematics and Computer Sciences, University of São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Inteligência Artificial

Original version

São Carlos

2023

Antonio de Sousa Leitão Filho

Desenvolvimento de um sistema de navegação e orientação para pessoas com baixa visão utilizando inteligência artificial

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Data de defesa: 09 de dezembro de 2023

Comissão Julgadora:

Prof. Dr. Valdir Grassi Junior
Orientador

Nícolas Roque dos Santos
Convidado

São Carlos
2023

Este trabalho é dedicado à minha família, especialmente aos meus filhos e à minha irmã, que mesmo sendo deficiente visual nunca para de sonhar ver nitidamente.

AGRADECIMENTOS

Agradeço ao Eterno Jeová, Todo-Poderoso Deus por sua graça infinita e doses sabedoria que partilhou com a humanidade desde o Éden.

Agradeço à minha família pelo amor incondicional.

Ao Prof. Dr. Valdir Grassi Junior pelas orientações e pela paciência.

À USP e à Coordenação do Curso de Pós-Graduação em Inteligência Artificial e Big Data pela oportunidade.

"O objetivo final da pesquisa em IA não é criar máquinas que nos superem em inteligência, mas criar máquinas que possam colaborar e coexistir conosco."

Stuart Russell

RESUMO

LEITAO FILHO, A. de S. **Desenvolvimento de um sistema de navegação e orientação para pessoas com baixa visão utilizando inteligência artificial**. 2023. 62p. Monografia - MBA em Inteligência Artificial e Big Data - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

Este trabalho teve como objetivo desenvolver uma prova de conceito de um sistema inteligente de navegação e orientação para pessoas com baixa visão. O método envolveu a integração de modelos de aprendizado profundo para visão computacional e processamento de linguagem natural. Os modelos visuais baseados em YOLOv8 alcançaram alto desempenho na detecção e segmentação de elementos urbanos cruciais. Contudo, a síntese de fala apresentou latência excessiva. Ainda assim, o sistema representou uma contribuição relevante e com potencial para aprimoramentos. Os resultados serviram para avaliar métricas-chave e obter indícios sobre o potencial de soluções multimodais para aplicações assistivas. Como trabalhos futuros, propõe-se reduzir a latência da fala e realizar mais testes com usuários reais.

Palavras-chave: Visão computacional. Síntese de fala. Tecnologia assistiva.

ABSTRACT

LEITAO FILHO, A. de S. **Navigation and orientation system based on artificial inteligence developed to aid persons with low vision**. 2023. 62p.
Monografia - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

This work aimed to develop a proof of concept of an intelligent navigation and guidance system for people with low vision. The method involved integrating deep learning models for computer vision and natural language processing. YOLOv8-based visual models achieved high performance in detecting and segmenting crucial urban elements. However, speech synthesis showed excessive latency. Still, the system represented a relevant contribution with potential for improvements. The results were used to evaluate key metrics and obtain clues about the potential of multimodal solutions for assistive applications. As future work, it is proposed to reduce speech latency and carry out more tests with real users.

Keywords: Computer vision. Speech Synthesis. Assistive Technology.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Deficiência visual e cegueira são desafios de saúde significativos que afetam pelo menos 2,2 bilhões de pessoas em todo o mundo, sendo que quase metade desses casos poderia ter sido prevenida ou ainda não foi abordada (WHO, 2022). A Organização Mundial da Saúde (OMS) classifica a deficiência visual em duas categorias, a deficiência visual próxima e a deficiência visual à distância, cada uma com diferentes níveis de gravidade, desde leve até cegueira conforme a acuidade (WHO, 2022). Tais deficiências podem dificultar substancialmente a capacidade de uma pessoa de navegar e interagir com seu ambiente, impactando sua independência, produtividade e qualidade de vida no geral.

Contudo, técnicas de Inteligência Artificial (IA) e Aprendizado de Máquina (ML) mostram-se como promessa considerável no apoio a indivíduos com deficiência visual. Pesquisas recentes desenvolveram sistemas baseados em aprendizado profundo para identificação em tempo real do ambiente, detecção e classificação de obstáculos, e sistemas de orientação baseados em imagens. Esses sistemas usam uma combinação de Redes Neurais Convolucionais (CNNs), abordagens de Long Short-Term Memory (LSTM) e tecnologias da Internet das Coisas (IoT), com precisão variando de 80% a mais de 95% (DHOU *et al.*, 2022). Embora esses sistemas tenham mostrado resultados promissores, ainda há potencial para melhoria e inovação na área.

Como exemplo, smartphones e tecnologias vestíveis oferecem uma plataforma prática e acessível para tecnologias assistivas baseadas em IA e IoT para indivíduos com deficiência visual. Aplicativos como, o Seeing AI da Microsoft, fornecem descrições sonoras do ambiente do usuário, lendo anotações escritas à mão, identificando produtos por meio de digitalização de códigos de barras e descrevendo a posição das pessoas em uma imagem. Da mesma forma, o Microsoft Soundscape cria um mapa sonoro 3D do ambiente do usuário usando dados de localização, balizas sonoras e som estéreo 3D sintetizado (MICROSOFT, 2022).

Apesar desses avanços, ainda há uma necessidade significativa de soluções mais abrangentes, intuitivas e acessíveis que aproveitem os últimos avanços em IA e IoT. O desenvolvimento de um sistema inteligente de navegação e orientação para pessoas com baixa visão poderia representar um avanço significativo a este respeito. Ao aproveitar o poder da IA para análise ambiental em tempo real e a conectividade da IoT para feedback e orientação oportunos, tal sistema poderia oferecer maior independência e qualidade de vida a essas pessoas (DHOU *et al.*, 2022) e (MICROSOFT, 2022).

1.1 Justificativa e Motivação

De acordo com a OMS, a deficiência visual representa um enorme ônus financeiro global, com os custos anuais globais de perdas de produtividade associadas à deficiência visual estimados em US$ 411 bilhões. As principais causas de deficiência visual e cegueira são erros de refração não corrigidos e cataratas. Globalmente, pelo menos 2,2 bilhões de pessoas têm deficiência visual próxima ou à distância. Em pelo menos 1 bilhão - ou quase metade - desses casos, a deficiência visual poderia ter sido prevenida ou ainda não foi tratada (IJPH, 2022).

Nesse contexto, a inteligência artificial e a internet das coisas emergem como ferramentas promissoras para auxiliar pessoas com deficiência visual, com o potencial de melhorar a qualidade de vida e a autonomia destes indivíduos. Isso corrobora com o fato de que a tecnologia tem desempenhado um papel fundamental na superação de barreiras, e a inteligência artificial tem progredido significativamente para melhorar a acessibilidade.

1.2 Questões de Pesquisa e Objetivos

Neste trabalho, tem-se como expectativa desenvolver uma prova de conceito (PoC) de um sistema inteligente de navegação e orientação para pessoas com baixa visão, que reconheça semáforos, faixas de pedestres e passeios públicos em uma paisagem urbana com capacidade de fornecer *feedback* de áudio conforme o cenário. Diante dos desafios e problemas atualmente enfrentados por sistemas assistivos inteligentes para pessoas com baixa visão, foi elaborada a seguinte questão de pesquisa que norteará este projeto:

Q1 "Como um sistema que utiliza visão computacional e feedback audível pode auxiliar a mobilidade e autonomia de pessoas com baixa visão em ambientes urbanos?"

Diante desta questão de pesquisa, são definidos os seguintes objetivos para o desenvolvimento deste trabalho:

- Empregar modelos de visão computacional para reconhecer semáforos, faixas de pedestres e passeios públicos.
- Utilizar um modelo *text-to-speech* para fornecer feedback de áudio para o usuário.
- Integrar, através de uma lógica, os resultados dos modelos de visão ao modelo *text-to-speech*.

1.3 Organização do Trabalho

Este trabalho está dividido em 5 capítulos: O Capítulo 1 trata da introdução ao assunto abordado neste trabalho, incluindo a sua motivação, justificativa e objetivos; o Capítulo 2 apresenta a revisão bibliográfica necessária para o entendimento do trabalho; o

Capítulo 3 contém o método proposto para o desenvolvimento de uma prova de conceito (PoC) de um sistema inteligente de navegação e orientação; o Capítulo 4 apresenta os resultados obtidos e suas respectivas discussões; e por fim, o Capítulo 5 capítulo apresenta a conclusão do trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 Modelos de Visão Computacional para Reconhecimento em Tempo Real

A visão computacional tornou-se um componente crítico em inúmeras aplicações devido à sua capacidade de automatizar tarefas complexas que requerem capacidades de visão semelhantes às humanas. Estas incluem a detecção de objetos, o reconhecimento de padrões, a análise de imagens, entre outras. O reconhecimento em tempo real, que é um subconjunto da visão computacional, permite aos sistemas identificar e classificar objetos ou padrões instantaneamente. Esta capacidade é crucial em setores como segurança pública, veículos autônomos e tecnologias assistivas para deficientes visuais, onde o principal requisito do sistema é a de fornecer resposta imediata ante a um cenário (VISIO.AI, 2023).

Muito em razão do largo poder de processamento computacional dos *smartphones* modernos, modelos de visão computacional processados na borda podem ser úteis para aplicações de tempo real. Pois no tocante a isso, Véstias *et al.* (2020) expõe que a tendência atual é mover o processamento de dados da nuvem para a borda, visto que em particular, os algoritmos de *machine learning* estão sendo cada vez mais implantados em dispositivos móveis. Um exemplo disso, é a utilização do *framework* de aprendizado de máquina como TensorFlow Lite ou PyTorch Mobile, que tem a capacidade de delegar parte ou toda a execução de um algoritmo para a GPU (Graphics Processing Unit) móvel, reduzindo a latência e aumentando a eficiência da execução do modelo, como em sistemas assistivos (MARTINEZ-ALPISTE *et al.*, 2022).

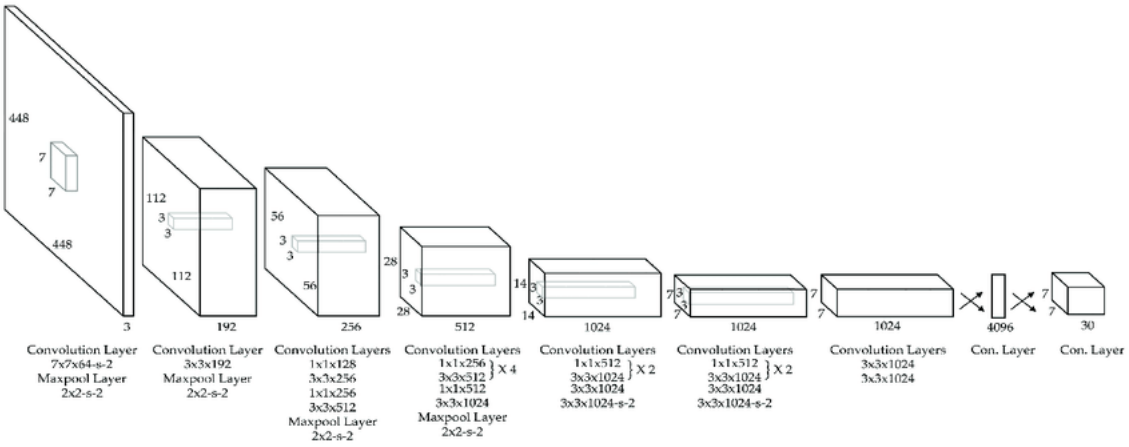
Nesse cenário, dos modelos de visão computacional, úteis para aplicações móveis de tempo real, que incorporam os benefícios do TensorFlow Lite, destacam-se o YOLOv8 (DAVANTHAPURAM; YU; SANIIE, 2021), SSD (Single Shot MultiBox Detector) (CHAI *et al.*, 2021) e MobileNetV3 + SSD (GUPTA *et al.*, 2023).

2.1.1 YOLOv8

O YOLO (You Only Look Once) foi concebido em 2015 por Redmon *et al.* (2016), na Universidade de Washington. Gautam *et al.* (2023) destaca que o YOLO é um modelo de visão notável para reconhecimento em tempo real. A precisão e velocidade excepcionais do YOLO contribuíram para sua rápida ascensão em popularidade. Como exemplo, a versão YOLOv2, divulgada em 2016, que incorporou funcionalidades como normalização de lote, caixas âncora e aglomerados de dimensões, aperfeiçoou o modelo original. A Figura 1 demonstra a arquitetura típica do YOLO.

Com o auxílio de uma *backbone network* mais eficiente, múltiplas âncoras e agrupamento piramidal espacial, o YOLOv3, apresentado em 2018, ampliou significativamente a

Figura 1 – Arquitetura típica da rede YOLO.



Fonte: Redmon *et al.* (2016)

capacidade do modelo. Inovações consideráveis, como a ampliação de dados em mosaico, uma nova cabeça de detecção livre de âncora e uma função de perda renovada, foram incorporadas no YOLOv4, lançado em 2020. O YOLOv5, com a adição de novos recursos, entre eles a otimização de hiperparâmetros, o rastreamento integrado de experimentos e a exportação automática para formatos de uso amplo, incrementou significativamente a funcionalidade do modelo (GAUTAM *et al.*, 2023).

Mais recentemente o modelo Ultralytics YOLOv8, para detecção de objetos e segmentação de imagens, aprimora os feitos de seus predecessores. O modelo YOLOv8, de vanguarda e contemporâneo, foi projetado para oferecer melhor desempenho, flexibilidade e eficiência. Dada a ênfase significativa em velocidade, tamanho e precisão de sua construção, o modelo se apresenta como uma opção atrativa para uma variedade de aplicações em inteligência artificial visual (GAUTAM *et al.*, 2023); (TERVEN; CORDOVA-ESPARZA, 2023); (Ultralytics, 2023).

2.1.2 SSD (Single Shot MultiBox Detector)

Consoante Liu *et al.* (2016), a abordagem SSD utiliza uma rede convolucional que, de maneira direta, cria caixas específicas na imagem para identificar e avaliar se um objeto pertence a determinada categoria. Para eliminar detecções redundantes ou sobrepostas, é utilizada uma técnica chamada supressão não máxima, que mantém a caixa de detecção e descarta as outras com sobreposições significativas. As camadas iniciais da rede derivam de uma estrutura padrão voltada para a classificação de imagens de alto padrão, interrompida antes de qualquer camada classificatória, a qual se denomina rede base. A essa estrutura, adiciona-se uma composição auxiliar para aprimorar a detecção, com as seguintes características principais:

1. **Mapas de recursos em multiescala para detecção** que incorpora camadas de características convolucionais ao término da rede base truncada, que diminuem gradualmente em tamanho, possibilitando a determinação de detecções em múltiplos tamanhos. O modelo convolucional para detecção varia conforme a camada de recursos, diferentemente de outras abordagens que utilizam um mapa de características único.

2. **Preditores convolucionais para detecção** onde cada camada de recursos inserida (ou potencialmente já existente na rede base) pode estabelecer um conjunto definido de previsões de detecção por meio de filtros convolucionais. O elemento fundamental para prever os parâmetros de uma possível detecção é um pequeno núcleo $3 \times 3 \times p$ que fornece uma avaliação para uma categoria ou uma modificação em relação à caixa padrão. Em cada localização onde o núcleo é utilizado, entrega-se um valor de saída.

3. **Caixas padrão e proporções de aspecto** são designadas como um conjunto de caixas delimitadoras padrão a cada seção do mapa de características, englobando vários mapas no topo da rede. Essas caixas organizam o mapa de maneira convolucional, mantendo a posição de cada caixa em relação à sua respectiva seção fixa. Para cada localização no mapa de recursos, prevê-se desvios em relação às caixas padrão, assim como as pontuações para cada classe que indicam a presença de uma categoria naquelas caixas.

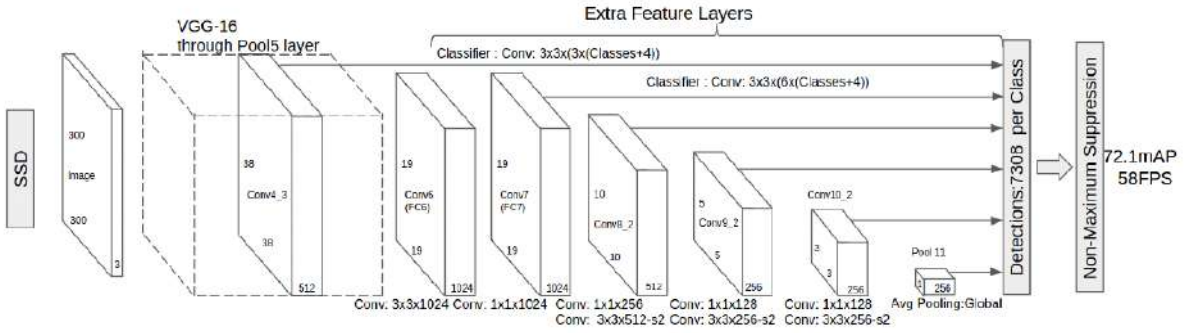
Em suma, Hossain and Momtaz (2023) expõe que a abordagem SSD fundamenta-se na estrutura de rede convolucional VGG-16, que historicamente tem sido considerada uma das arquiteturas de vanguarda em tarefas de reconhecimento de imagem. Este alicerce proporciona à SSD uma base robusta, permitindo que ela se beneficie da capacidade da VGG-16 de extrair características detalhadas e discriminativas de imagens, como é revelada na arquitetura do modelo na Figura 2.

Ademais, ao excluir as camadas totalmente conectadas e incorporar estratégias como mapas de recursos em multiescala e preditores convolucionais, o SSD otimiza a eficiência e a precisão em tempo real. Tal inovação traduz a contínua evolução na intersecção da visão computacional e das redes neurais profundas, onde o objetivo não é apenas melhorar a performance, mas também adaptar-se às demandas de aplicações em tempo real e às limitações dos dispositivos de borda, por exemplo.

2.1.3 MobileNetV3 + SSD

No contexto da visão computacional, ao passo que o modelo SSD emergiu como um divisor de águas, estabelecendo-se como o pioneiro entre os detectores de um único estágio, cuja notoriedade reside na capacidade de rivalizar em desempenho com detectores

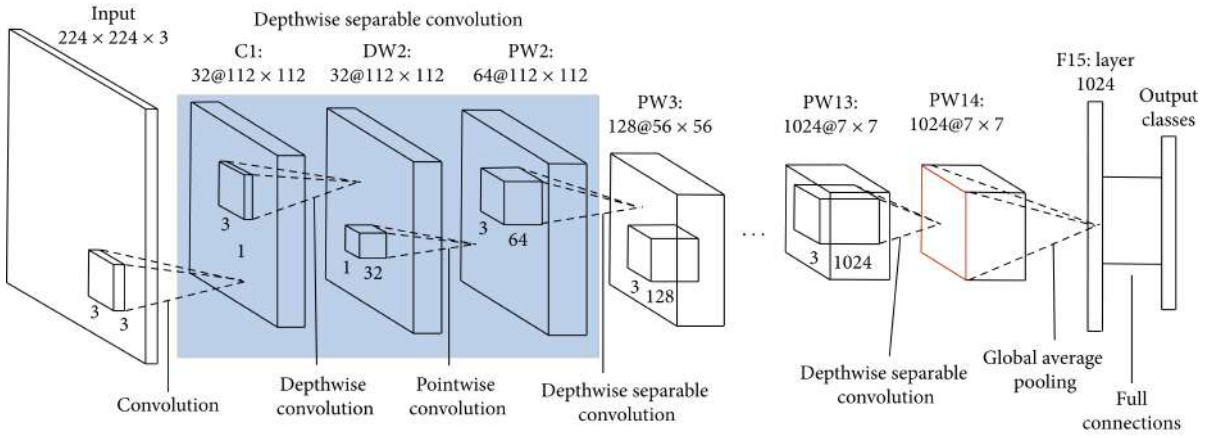
Figura 2 – Arquitetura do modelo SSD.



Fonte: Liu *et al.* (2016)

de dois estágios, o que representou uma conquista significativa na aceleração de detecções em tempo real. Em paralelo a isso, o avanço das aplicações móveis exigia modelos mais leves, eficientes e ainda assim robustos. Foi nesse cenário que a arquitetura MobileNet foi concebida, direcionada especialmente para otimizar o processamento em dispositivos móveis. Ao invés de se ater às camadas convolucionais tradicionais, a primeira versão do MobileNet introduziu convoluções separáveis em profundidade, marcando uma revolução na eficiência computacional (HOWARD *et al.*, 2017), como ilustra a Figura 3.

Figura 3 – Arquitetura da rede MobileNet.



Fonte: Wang *et al.* (2020)

O MobileNetV2, por sua vez, inovou ainda mais ao incorporar o conceito de *linear bottleneck*. Esta técnica de otimização, que minimiza distorções causadas por não linearidades, possibilita um processamento mais eficiente, sem comprometer o desempenho (SANDLER *et al.*, 2018). MobileNetV3, a evolução subsequente, refina ainda mais a

arquitetura, trazendo inovações e ajustes que amplificam sua aplicabilidade e eficácia em ambientes *mobile* (HOWARD *et al.*, 2019).

Notadamente a combinação do MobileNetV3 com SSD potencializa as capacidades de detecção em tempo real, aproveitando a leveza e eficiência do MobileNetV3, ao mesmo tempo que tira proveito da robustez do SSD. Essa fusão se revela particularmente poderosa em circunstâncias onde a eficiência no processamento e a precisão são igualmente cruciais, como em dispositivos móveis e sistemas embarcados, estabelecendo um novo padrão para aplicações de detecção em tempo real (ZHANG; LIU, 2023).

2.2 Modelos Text-to-Speech

Os sistemas de síntese de fala, também conhecidos como text-to-speech (TTS), tornaram-se indispensáveis para interfaces mais naturais entre humanos e máquinas, com capacidade de gerar fala artificial inteligível a partir de texto permitindo maior acessibilidade e conveniência em assistentes virtuais, sistemas de navegação e outros aplicativos falados (WANG *et al.*, 2017). De modo especial, modelos TTS baseados em deep learning têm alcançado uma qualidade de áudio próxima à fala humana. Isso os torna adequados para uso em massa em dispositivos móveis e IoT, conforme expõem Sotelo *et al.* (2017).

Arquiteturas de ponta a ponta para TTS podem alcancar o estado da arte em sintese neural (SHEN *et al.*, 2018). Nesse sentido, o Tacotron 2 + WaveNet do Google, por exemplo, merece destaque, visto que inclui o Tacotron 2 para prever características espectrais a partir do texto, e o WaveNet como *vocoder* neural para gerar as amostras de áudio. Outros modelos notáveis incluem o EfficientSpeech, uma abordagem capaz de sintetizar voz em tempo real em dispositivos com *Central Processing Unit tipo Advanced RISC Machine* (CPU-ARM) (ATIENZA, 2023), e o FastSpeech 2 da NVIDIA, que é focado em velocidade de inferência (ŁAńCUCKI, 2021).

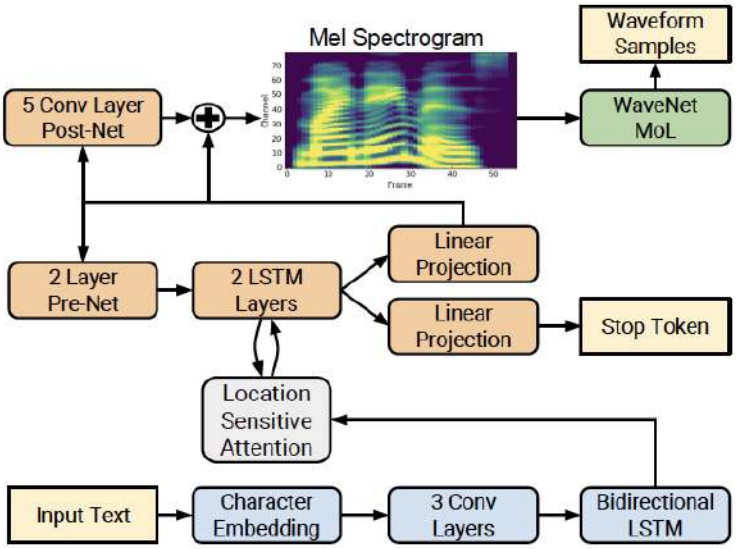
2.2.1 Tacotron 2 + WaveNet (Google)

A combinação Tacotron 2 + WaveNet (Google) possui uma arquitetura de ponta a ponta para a síntese de fala artificial de alta naturalidade. Enquanto o Tacotron 2 emprega um mecanismo de atenção, que possibilita o preciso alinhamento entre o texto de entrada e os mel-espectrogramas previstos (SHEN *et al.*, 2018), o WaveNet, por sua vez, incorpora o conceito de dilatações causais, que permite modelar com impressionante eficiência as complexas dependências temporais no sinal de áudio (OORD *et al.*, 2016).

A Figura 4 ilustra como essa arquitetura híbrida aproveita o melhor dos dois modelos. O Tacotron 2 gera mel-espectrogramas alinhados ao texto usando seu codificador-decodificador com atenção, ao passo que o WaveNet então sintetiza as amostras de áudio

condicionado nesses mel-espectrogramas, por meio de uma rede neural convolucional profunda.

Figura 4 – Diagrama de blocos da arquitetura da combinação entre Tacotron2 + Wavenet.



Fonte: Shen *et al.* (2018)

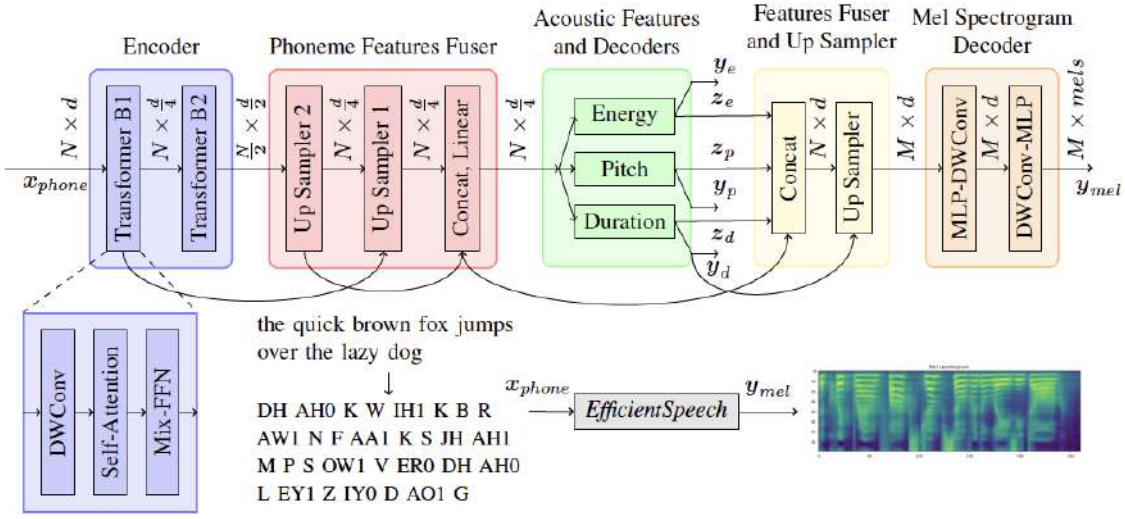
Essa combinação de capacidades complementares possibilita alcançar impressionante naturalidade na fala sintética, antes inimaginável em sistemas baseados em deep learning (SHEN *et al.*, 2018). Todavia, o alto custo computacional dessa combinação complexa ainda representa um obstáculo para aplicações em tempo real (TACHIBANA; UENOYAMA; AIHARA, 2018). Pesquisas em técnicas de otimização como, poda de parâmetros e destilação de conhecimento buscam tornar essa arquitetura acessível na prática (KURATA *et al.*, 2021).

2.2.2 EfficientSpeech

O EfficientSpeech é uma proposta de modelo de síntese de fala (TTS) voltado para uso em dispositivos embarcados, os quais tipicamente apresentam restrições de memória, capacidade de processamento e acesso à rede (ATIENZA, 2023). Seu propósito é viabilizar a geração de áudio sintetizado em tempo real nessas plataformas, mitigando problemas de privacidade e dependência de serviços em nuvem.

Para tanto, o EfficientSpeech emprega uma arquitetura baseada em Transformers não-autoregressiva compacta, estruturada como U-Net e contendo apenas 266k parâmetros, conforme ilustrado na Figura 5. Essa topologia combina módulos de codificação, fusão de recursos e decodificação para mapear os fonemas em espectrogramas acústicos de maneira eficiente (ATIENZA, 2023).

Figura 5 – Arquitetura do modelo EfficientSpeech.



Fonte: Atienza (2023)

Experimentos evidenciam que essa abordagem propicia síntese de fala em tempo real em dispositivos com arquitetura ARM do tipo RPi4, com fator de tempo real médio de 104,3 e 90 MFLOPS de consumo (ATIENZA, 2023). O EfficientSpeech representa um avanço em TTS para aplicações móveis em tempo real.

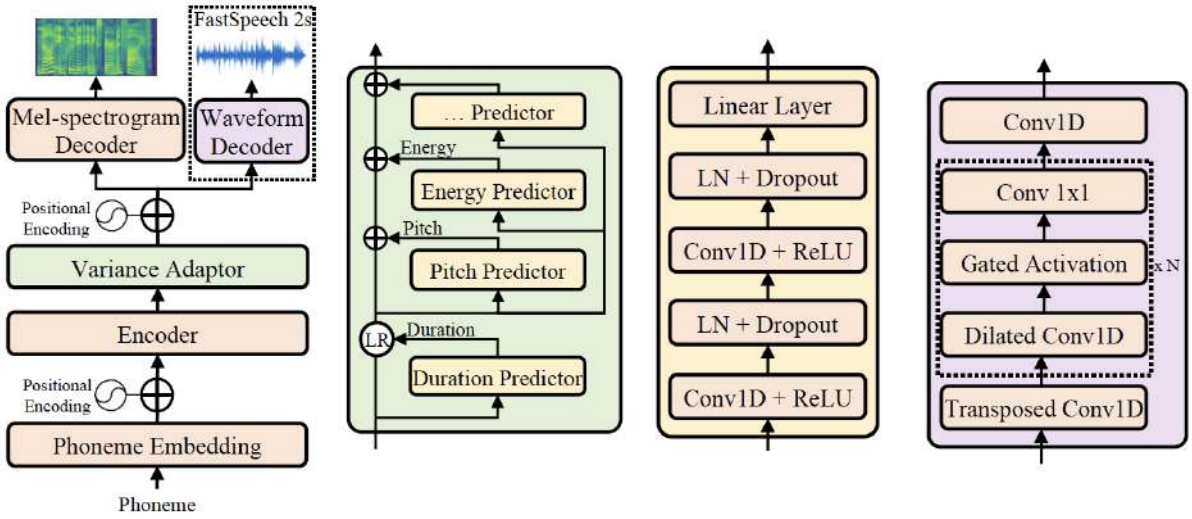
2.2.3 FastSpeech 2

O FastSpeech 2, proposto por Ren *et al.* (2022), aborda limitações de modelos antecessores como o FastSpeech, visando melhorar o mapeamento um-para-muitos na síntese de fala. Ao invés de treinar com saídas simplificadas, utiliza-se o áudio real como alvo. Além disso, informações de prosódia como duração, pitch e energia são extraídas da fala natural e incorporadas durante treinamento e inferência.

Conforme ilustrado na Figura 6, sua arquitetura conta com um codificador Transformer que mapeia os fonemas em representações latentes, um adaptador de variância prosódica e um decodificador que gera os mel-espectrogramas em paralelo (REN *et al.*, 2022). O FastSpeech 2s estende o modelo para síntese diretamente da forma de onda.

Resultados experimentais evidenciam ganhos de 3x na velocidade de treinamento e aprimoramentos na qualidade do áudio em relação ao FastSpeech original. O FastSpeech 2 até supera modelos autoregressivos em naturalidade, mantendo inferência rápida (REN *et al.*, 2022).

Figura 6 – Arquitetura do modelo FastSpeech 2.



Fonte: Ren *et al.* (2022)

2.3 Trabalhos Relacionados

Nesta seção são apresentados trabalhos que envolvem a utilização de modelos de visão e de fala sintética em aplicações integradas de tecnologia assistiva em tarefas de navegação e orientação.

2.3.1 Sistemas assistivos multimodais integrados (visão + fala)

Islam *et al.* (2023) propôs um sistema de baixo custo para detecção de obstáculos e descrição do ambiente. A abordagem combina o modelo SSDLite MobileNetV2 de detecção de objetos com o módulo de síntese de fala Google Text-to-Speech para prover feedback sonoro em tempo real. O sistema foi validado por meio de implementação em ambiente de desktop e plataforma embarcada Raspberry Pi.

Para isso, o modelo SSDLite MobileNetV2 foi treinado no dataset COCO de detecção de objetos, composto por 328.000 imagens categorizadas em 90 classes. Para o modo de descrição de ambiência, foi utilizado um dataset climático com 500 imagens de quatro classes (ensolarado, nublado, chuvoso e neblina). O modelo obteve precisão média (mAP) de 0,791 e taxa de quadros por segundo de 2,15 quando executado no Raspberry Pi 4. Já o modo ambiência, alcançou 95% de acurácia após apenas 15 épocas de treinamento. Quando integrado em protótipo de hardware, o sistema apresentou 88,89% de acurácia na detecção de objetos e aproximadamente 90% de acurácia de validação na classificação das cenas.

No trabalho proposto por Guravaiah *et al.* (2023), chamado de Third Eye, o sistema emprega o modelo YOLOv5 para detecção de objetos em imagens capturadas em tempo real. As saídas de detecção são convertidas em fala sintetizada pelas bibliotecas gTTS e pyttsx3. Sendo que o modelo YOLOv5 foi treinado em um dataset customizado contendo 95 classes.

O *Third Eye* contou com o modelo YOLOv5 para detecção, sendo treinado em um COCO (Common Objects in Context) *dataset* personalizado composto por 5000 imagens rotuladas. Este *dataset* incluiu 95 classes no total, sendo 15, contextuais à realidade indiana. Guravaiah *et al.* (2023) expõe que as imagens apresentavam variadas condições de luminosidade e orientações dos alvos. Para a síntese de fala, contrastaram-se duas bibliotecas em Python: gTTS e pyttsx3. Após 50 épocas de treinamento, o modelo alcançou precisão de 0,45, revocação de 0,35 e mAP de 0,3.

Já o sistema proposto por Chen *et al.* (2023), funciona como um guia móvel para ajudar a mobilidade em pisos táteis de pedestres deficientes visuais. O sistema proposto foi implementado integralmente em *smartphone* com sistema operacional Android. A abordagem emprega o *framework* MobileNetV2 pré-treinado na base de dados ImageNet para extração de características visuais dos pisos táteis no solo. Sabe-se que o MobileNetV2 utiliza convoluções separáveis para construir uma rede neural leve e eficiente em termos de processamento. O algoritmo SSD (Single Shot MultiBox Detector) foi então aplicado nas saídas do MobileNetV2 para detectar e delimitar precisamente a localização dos pisos táteis nas imagens. Após análise, as coordenadas e classes preditas pelo modelo são convertidas em comandos verbais pelos módulos de síntese de fala do Android.

Para este *app*, o modelo MobileNetV2-SSD foi treinado em um conjunto de dados personalizado composto por 1323 imagens rotuladas. Essas imagens retratavam pisos táteis do tipo direcional, inseridos em ambientes internos e externos diversificados. O modelo neural alcançou 93,76% de acurácia na detecção dos pisos táteis, com 0,94 de precisão e 0,93 de revocação. Após a quantização, a taxa de processamento do *framework* foi de 14 quadros por segundo em um *smartphone* com processador Snapdragon 660.

2.4 Considerações Finais

Considerando os trabalhos relacionados analisados, modelos de aprendizado profundo têm demonstrado grande potencial para compor sistemas de assistência visual integrados com síntese de fala. Em especial, a família YOLO se destaca para a tarefa de detecção em tempo real devido à sua arquitetura leve e eficiente, como evidencia Reis *et al.* (2023).

O YOLOv8, especificamente, traz aprimoramentos no *backbone* CSP (*Cross Stage Partial*) que resultam em aumento de 5% na precisão média (mAP) e 30% na velocidade

em relação ao YOLOv5, como evidenciado por Gašparović *et al.* (2023). Isso viabiliza o processamento fluido de imagens para navegação assistiva em cenários urbanos contendo semáforos, faixa de pedestre e passeio público, por exemplo.

Já para síntese de fala multidialetal, o modelo Massively Multilingual Speech (MMS) proposto por Pratap *et al.* (2023) representa o estado da arte, também sendo de código aberto. Seu treinamento com mais de 50 mil horas de fala em 1.128 línguas permite conversão textual-fala (TTS) inteligível para centenas de idiomas, incluindo o português, com uma única rede. A fim de avaliar métricas-chave como precisão, acurácia, F1 *score*, *recall* e latência, a integração do YOLOv8 com o MMS pode resultar em uma solução assistiva e personalizável, podendo gerar descrições verbais a partir da interpretação visual de cenários urbanos, contendo semáforos, faixa de pedestre e passeio público.

3 MATERIAIS E MÉTODOS

3.1 Projeto de Estudo

Para investigar o potencial de modelos de visão computacional e processamento de linguagem natural baseados em aprendizado profundo na assistência a pessoas com deficiência visual, este estudo propõe o desenvolvimento de uma prova de conceito (PoC) de um sistema inteligente de navegação e orientação.

O sistema visa reconhecer semáforos, faixas de pedestres e passeios públicos em imagens, fornecendo resposta em áudio ao usuário por meio da integração de modelos de aprendizado profundo, pretendendo responder à seguinte pergunta: "Como um sistema que utiliza visão computacional e *feedback* audível pode auxiliar a mobilidade e autonomia de pessoas com baixa visão em ambientes urbanos?"

Para tanto, foram empregados modelos de ponta como o YOLOv8 (Ultralytics, 2023) e o sintetizador de fala MMS (PRATAP *et al.*, 2023). Experimentos serviram para avaliar métricas-chave como precisão, F1-*score*, *recall* e latência. Através do desenvolvimento desta prova de conceito e dos experimentos realizados foi possível obter indícios sobre o potencial de soluções multimodais com visão computacional e processamento de linguagem natural para aplicações de tecnologia assistiva.

3.2 Base de Dados

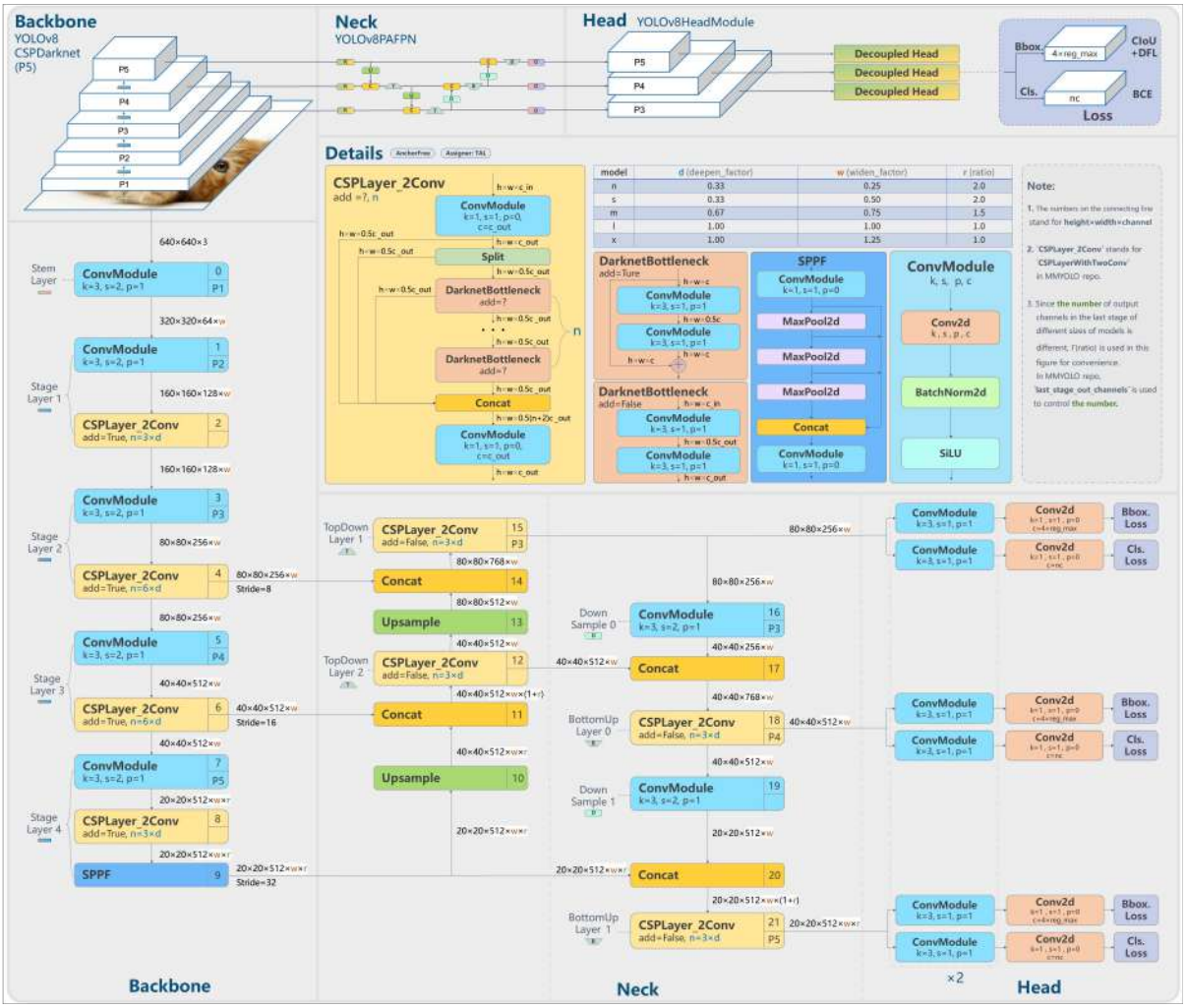
Para os modelos de visão, as bases de dados utilizadas foram construídas com imagens de cenários urbanos disponíveis publicamente na plataforma Roboflow. Para detecção, o primeiro *dataset* contém 2307 imagens, com resolução de 640x640 pixels, dividido em 70% para treinamento, 20% para validação e 10% para teste (YRIS, 2023a). Já para segmentação, o segundo *dataset* contém originalmente 220 imagens, que com as técnicas de aumentação de dados *exposure* entre -42% e +42% e *blur* de até de 5.5 pixels, perfazendo um total de 526 imagens, também de 640x640 pixels de resolução, dividido em 87% para treinamento, 9% para validação e 4% para teste (YRIS, 2023b). O conjunto de dados do modelo de fala consistiu em mais 44,7 mil horas de fala anotada, com corpus derivado da leitura do Novo Testamento em 1.107 línguas (PRATAP *et al.*, 2023).

3.3 Modelo de Rede Neural Profunda para Segmentação e Detecção Visual

O modelo Ultralytics YOLOv8 (Ultralytics, 2023) destaca-se como uma arquitetura de ponta para detecção e segmentação visual baseada em redes neurais profundas. Sua engenharia inovadora integra os recentes avanços em *backbone networks*, tais quais as camadas CSP *(Cross Stage Partial)* que otimizam o fluxo direcional de informações na

rede. Além disso, apresenta um *design* de pescoço *(neck)* único com a estrutura PAN *(Path Aggregation Network)* para aprimorar a extração e combinação de características visuais. O cabeçote *(head)* utiliza uma abordagem livre de âncoras *(anchor-free)*, com o *design decoupled* da Ultralytics, simplificando e robustecendo o processo de reconhecimento de objetos, como demonstra a Figura 7. O modelo equilibra precisão, velocidade e consumo computacional, tornando-o apropriado para uso em aplicações de tempo-real. Há diversas versões pré-treinadas disponíveis, customizadas para casos de uso distintos. Assim, o YOLOv8 provê alto desempenho aliado à flexibilidade de configuração para integração em diversos sistemas que demandam capacidades avançadas de visão computacional.

Figura 7 – Arquitetura do YOLOv8



Fonte: RangeKing (2023)

A Ultralytics utilizou o conjunto de dados COCO (COCO Consortium, 2023) como base para o pré-treinamento inicial dos modelos YOLOv8, visando potencializar

suas capacidades de detecção e segmentação visual. O COCO constitui um repositório de imagens amplamente empregado para o desenvolvimento de algoritmos de ponta em visão computacional, reunindo cerca de 200 mil imagens anotadas em 80 classes de objetos cotidianos. Ao disponibilizar métricas padronizadas como mAP e mAR, o conjunto COCO viabiliza a avaliação precisa e comparável do desempenho entre modelos. Assim, submeter o YOLOv8 a este processo de treinamento prévio mostrou-se essencial para solidificar sua precisão e confiabilidade na interpretação visual de elementos comuns em imagens reais, estabelecendo bases sólidas para sua aplicação subsequente em projetos específicos.

Para o refinamento dos modelos YOLOv8m e YOLOv8m-seg, foi utilizado o ambiente Google Colaboratory com a GPU NVIDIA Tesla T4 de 16GB. O processo de *fine-tuning* para a detecção foi conduzido ao longo de 30 épocas, processando *mini-batches* de 16 imagens de resolução 640x640 pixels. Utilizou-se o otimizador AdamW, ajustado com uma taxa de aprendizagem inicial de 0.001111 e *momentum* de 0.9. Paralelamente, para a segmentação semântica, aplicou-se uma taxa de aprendizagem de 0.001429, mantendo-se a mesma resolução e número de épocas. Técnicas de regularização como *weight decay* foram empregadas para evitar o *overfitting*. Além disso, estratégias de *learning rate scheduling* e técnicas de *data augmentation* foram implementadas para aprimorar a generalização do modelo. Esses procedimentos visaram calibrar finamente os modelos às características específicas do *dataset* em uso.

3.4 Modelo de Fala

O modelo de síntese de fala utilizado é o mms-tts-por, integrante da família Massively Multilingual Text-to-Speech (MMS TTS) da Meta AI. Trata-se de um modelo pré-treinado baseado na arquitetura Transformer, implementado no framework PyTorch, para converter texto em fala de forma escalável para milhares de idiomas.

Especificamente, emprega-se a abordagem *Variational Inference with adversarial learning for end-to-end Text-to-Speech* (VITS) com um codificador de texto via múltiplas camadas de atenção e um decodificador para geração das *features* acústicas e forma de onda. O pré-treinamento envolveu o *corpus* MMS-lab, derivado da leitura do Novo Testamento em 1.107 línguas, totalizando 44,7 mil horas de fala anotada. Utilizou-se o otimizador Adam com 1 milhão de atualizações por idioma, em *batches* efetivos de 2,3 a 3,5 horas distribuídos em 64 GPUs NVIDIA A100. A taxa de aprendizado foi aquecida nas primeiras 32.000 etapas e então decaiu polinomialmente.

Análises automatizadas e humanas indicaram alto nível de inteligibilidade e naturalidade para a grande maioria dos 1.107 modelos VITS gerados. Conclui-se que essa arquitetura, devidamente pré-treinada, mostrou-se apropriada para sintetizar fala expressiva e compreensível a partir de qualquer entrada de texto nos diversos idiomas contemplados. Portanto, o *mms-tts-por* viabiliza soluções de conversão texto-fala robustas

e escaláveis para o sistema proposto.

3.5 Sistema Integrado de Visão e Fala

O Sistema Integrado de Visão e Fala foi desenvolvido inteiramente em Python, utilizando o ambiente PyCharm, implementado em uma estação de trabalho portátil com um Apple MacBook Pro M2 equipado com 16 GB de memória RAM e 512 GB de armazenamento em estado sólido. Essa configuração de hardware de alto desempenho foi a aposta para viabilizar o processamento das funcionalidades complexas de visão computacional e processamento de linguagem natural envolvidas no sistema.

O núcleo do sistema consiste em dois modelos de aprendizado profundo baseados na arquitetura YOLOv8. O modelo *yris_detect.pt* é especializado na detecção de cinco classes de objetos urbanos: semáforos abertos e fechados para pedestres e veículos, e semáforos específicos para pedestres. O modelo *yris_segm.pt* foca na segmentação semântica de faixas de pedestre e calçadas. Ambos são integrados ao módulo *vision.py*, que processa imagens para identificar esses elementos em cenários urbanos. A saída deste módulo é um conjunto de dados que inclui coordenadas de caixas delimitadoras, índices de classes e máscaras de segmentação, representando visualmente os objetos e áreas de interesse identificados, como é detalhado pelo algoritmo da Figura 8.

Figura 8 – Algoritmo do módulo *vision.py*

```
1  Início
2      Definir a classe Vision com dois atributos: detect_model
   e segment_model (ambos modelos YOLO)
3
4      Função detect_objects:
5          Entrada: caminho da imagem, limiar de confiança
6          Utilizar detect_model para processar a imagem com o
   limiar de confiança especificado
7          Retornar os resultados da detecção, incluindo caixas
   delimitadoras e classes dos objetos detectados
8
9      Função segment_objects:
10         Entrada: caminho da imagem, limiar de confiança
11         Utilizar segment_model para processar a imagem com o
   limiar de confiança especificado
12         Retornar os resultados da segmentação, incluindo
   máscaras de segmentação e classes das áreas segmentadas
13 Fim
```

Fonte: Elaborada pelo autor.

Posteriormente, o módulo *processing.py* recebe esses dados e realiza um pós-processamento. Ele analisa e interpreta as coordenadas e índices das caixas delimitadoras,

bem como as máscaras de segmentação, para extrair informações contextuais. Por exemplo, ele determina a presença e o estado dos semáforos (abertos ou fechados) e a localização das faixas de pedestre e calçadas. A saída desse módulo é um conjunto de atributos booleanos, como *'is_pedestrian_light_red'*, *'has_crosswalk'*, pormenorizado na Figura 9.

Em seguida, as informações processadas são passadas para o módulo *context_analyzer.py*. Este módulo interpreta os dados processados e gera mensagens contextualizadas para o usuário. As mensagens incluem frases como "Neste cenário, existe faixa de pedestre e calçada", "Pare! O semáforo de pedestre está fechado. Aguarde o sinal abrir", entre outras, dependendo do contexto detectado e processado. A Figura 10 detalha algotmicamente o módulo *context_analyzer.py*.

Finalmente, o módulo *speech.py* converte essas mensagens textuais em áudio. Utilizando o modelo de síntese de fala *mms-tts-por*, este módulo transforma as frases em sinais acústicos claros e audíveis, salvando cada amostra de áudio como *voice_yris_#.wav* no diretório *outputs*. Os detalhes do algoritmo do módulo *speech.py* é detalhado na Figura 10.

A Figura 12 ilustra de maneira esquemática o fluxo de dados e a interação entre os módulos do Sistema Integrado de Visão e Fala.

Figura 9 – Algoritmo do módulo *processing.py*

```
Início
    Função process_results:
        Entrada: resultados de segmentação
(segmentation_results), resultados de detecção
(detection_results)
        Inicializar um dicionário (processed_info) para
armazenar informações processadas com chaves específicas
para cada categoria relevante

        Processamento dos Resultados de Segmentação:
            Para cada resultado em segmentation_results:
                Se existirem caixas delimitadoras
(result.boxes não é nulo e tem itens):
                    Para cada caixa delimitadora (box) em
result.boxes.xyxy:
                        Obter o índice da classe (class_id)
da caixa atual
                        Se class_id é 0 (faixa de pedestre):
                            Definir
processed_info['has_crosswalk'] como True
                        Se class_id é 2 (passeio):
                            Definir
processed_info['has_sidewalk'] como True

        Processamento dos Resultados de Detecção:
            Para cada resultado em detection_results:
                Se existirem caixas delimitadoras:
                    Para cada caixa delimitadora:
                        Obter o índice da classe (class_id)
                        Avaliar class_id e atualizar
processed_info de acordo:
                            Se class_id é 3 (vermelho
pedestre):
                                Definir
processed_info['is_pedestrian_light_red'] como True
                            Se class_id é 4 (vermelho
veículos):
                                Definir
processed_info['is_vehicle_light_red'] como True
                            Se class_id é 1 (verde
pedestre):
                                Definir
processed_info['is_pedestrian_light_green'] como True
                            Se class_id é 0 (semáforo
pedestre):
                                Definir
processed_info['is_pedestrian_traffic_light'] como True
                            Se class_id é 2 (verde
veículos):
                                Definir
processed_info['is_vehicle_light_green'] como True

        Retornar o dicionário processed_info com as
informações processadas
Fim
```

Fonte: Elaborada pelo autor.

Figura 10 – Algoritmo do módulo *context_analyzer.py*

```
Início
    Função analyze_context:
        Entrada: informações processadas (processed_info), sistema de fala (speech)
        Inicializar uma lista vazia para armazenar mensagens contextuais (context_messages)

        Baseado nas informações processadas:
            Se processed_info['has_crosswalk'] é True e processed_info['has_sidewalk'] é True:
                Adicionar 'Neste cenário, existe faixa de pedestre e calçada.' à lista context_messages
            Senão, se processed_info['has_crosswalk'] é True:
                Adicionar 'Existe faixa de pedestre neste cenário.' à lista context_messages
            Senão, se processed_info['has_sidewalk'] é True:
                Adicionar 'Transite pelo passeio público.' à lista context_messages
            Senão:
                Adicionar 'Faixas de pedestre ou passeio público não foram detectados. Se possível, peça ajuda.' à lista context_messages

            Se processed_info['is_vehicle_light_red'] é True:
                Adicionar 'Sinal fechado para veículos.' à lista context_messages
            Se processed_info['is_pedestrian_light_red'] é True:
                Adicionar 'Pare! O semáforo de pedestre está fechado. Aguarde o sinal abrir.' à lista context_messages
            Se processed_info['is_vehicle_light_green'] é True:
                Adicionar 'Cuidado! Semáforo aberto para veículos. Aguarde sua vez.' à lista context_messages
            Se processed_info['is_pedestrian_light_green'] é True:
                Adicionar 'Agora você pode transitar em segurança pela faixa de pedestre.' à lista context_messages

        Para cada mensagem na lista context_messages:
            Utilizar o sistema de fala (speech) para converter a mensagem em áudio
            Reproduzir o áudio gerado
            Aguardar um intervalo de tempo entre as mensagens, se houver mais de uma mensagem na lista
Fim
```

Fonte: Elaborada pelo autor.

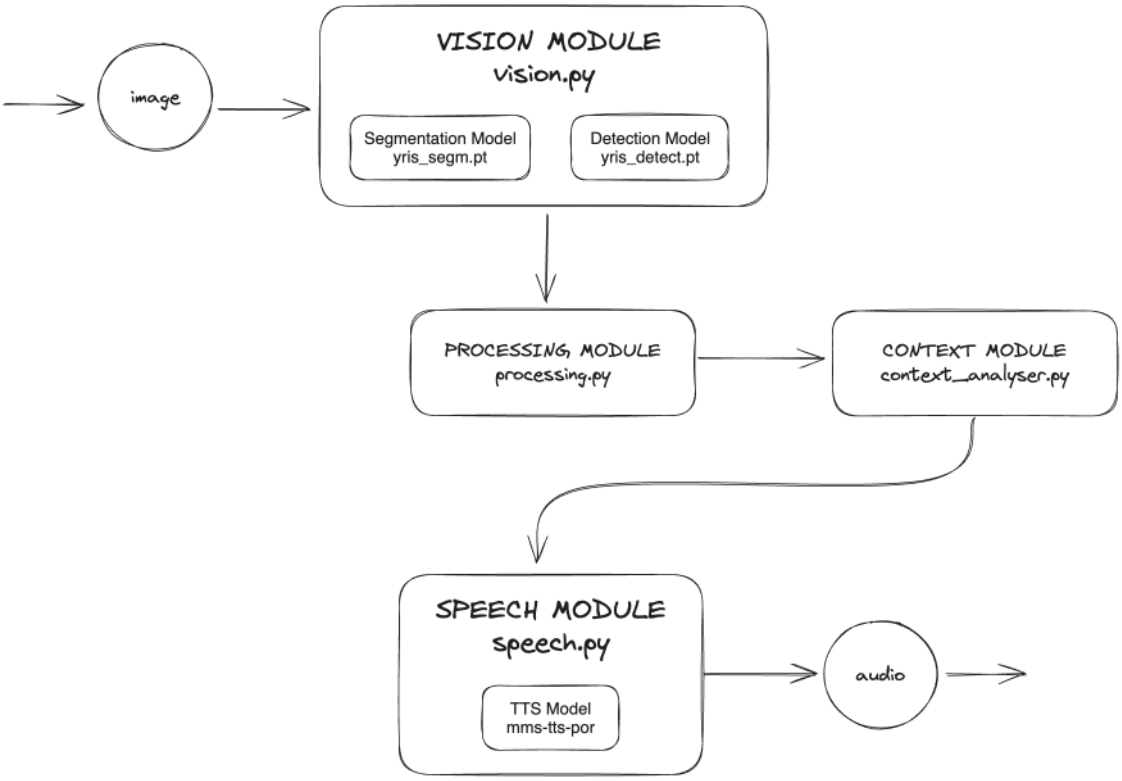
Figura 11 – Algoritmo do módulo *speech.py*

```
Início
    Definir a classe Speech com inicialização do modelo VitsModel e tokenizer
    Inicializar contador e diretório de saída

    Função text_to_speech:
        Entrada: texto
        Utilizar tokenizer para converter o texto em formato tensor
        Gerar waveform de áudio a partir do modelo com o tensor
        Normalizar e converter o áudio para o formato PCM 16-bit
        Gerar nome de arquivo de áudio único no diretório 'outputs'
        Salvar arquivo de áudio
        Reproduzir áudio usando comando do sistema operacional
        Incrementar contador de arquivos
Fim
```

Fonte: Elaborada pelo autor.

Figura 12 – Diagrama de Blocos do Sistema Integrado de Visão e Fala Proposto



Fonte: Elaborada pelo autor.

4 RESULTADOS E DISCUSSÕES

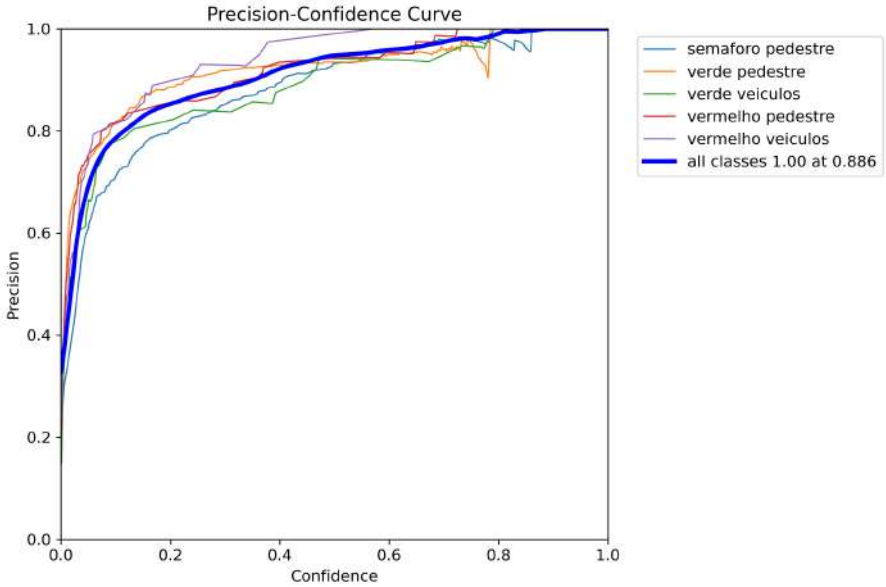
Os experimentos conduzidos (ajuste fino e 50 medições do tempo de processamento das partes do sistema) permitiram uma avaliação ampla das capacidades do sistema integrado de visão computacional e síntese de fala proposto. Foram coletadas métricas quantitativas como precisão, F1-*score* e *recall* do processo de ajuste fino dos modelos de visão (detecção e segmentação) e desempenho do sistema em termos de latência, bem como uma análise qualitativa por meio de casos de uso.

4.1 Métricas Quantitativas de Detecção Visual

Empregando a arquitetura YOLOv8 da Ultralytics, o modelo de detecção visual *yris_detect.pt* foi avaliado quanto a seu desempenho. Através de conjuntos de validação e teste, foram coletadas métricas como precisão, revocação e pontuação F1 para cada classe de objetos de interesse, incluindo os estados dos semáforos para pedestres e veículos. Simultaneamente, acompanhou-se a progressão dessas métricas durante as 30 épocas de treinamento supervisionado.

4.1.1 Avaliação do Modelo *yris_detect.pt*

Figura 13 – Curva Precisão-Confiança para Detecção

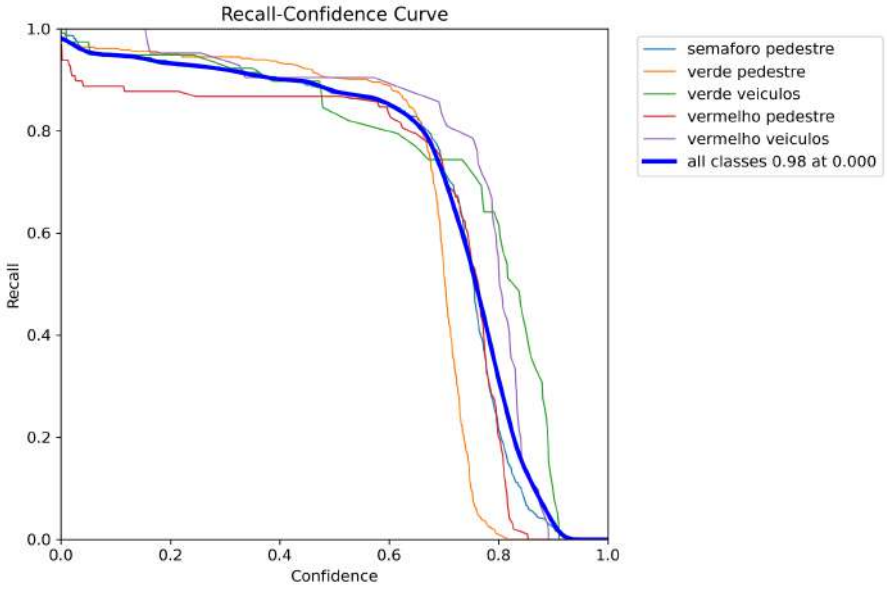


Fonte: Elaborada pelo autor.

O gráfico da Figura 13 apresenta a curva de precisão-confiança para a detecção de diferentes classes de objetos-alvo de interesse. As curvas representam a precisão do modelo

em relação ao seu nível de confiança na detecção de semáforos para pedestres (azul-escuro), luz verde para pedestres (laranja), luz verde para veículos (verde), luz vermelha para pedestres (vermelho) e luz vermelha para veículos (roxo). Observa-se que o sistema alcança uma precisão ótima (1.00) a um nível de confiança de 0.886 para todas as classes (azul), indicando um desempenho confiável.

Figura 14 – Curva *Recall*-Confiança para Detecção

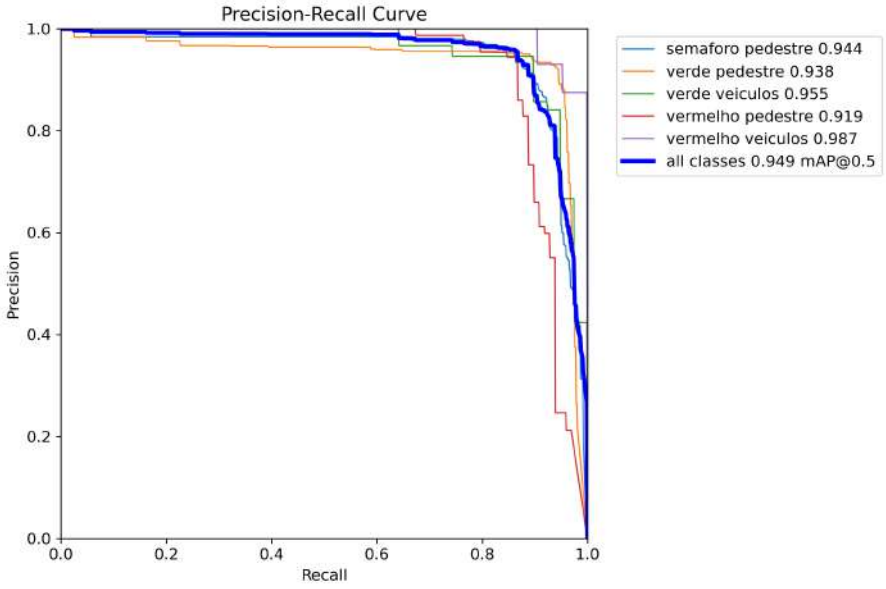


Fonte: Elaborada pelo autor.

Na Figura 14, é apresentada a curva de recall-confiança para a detecção das diversas classes de objetos relevantes pelo modelo *yris_detect.pt*. As curvas ilustram o *recall* do modelo em função do nível de confiança para a detecção de semáforos para pedestres (azul-escuro), luz verde para pedestres (laranja), luz verde para veículos (verde), luz vermelha para pedestres (vermelho) e luz vermelha para veículos (roxo). Nota-se que, para todas as classes, o modelo apresenta um *recall* de 0.98 mesmo em um nível de confiança zero (azul). Esta característica revela uma alta sensibilidade do modelo em identificar os objetos-alvo corretamente, reduzindo o risco de falsos negativos em um contexto de aplicação real.

A curva precisão-*recall* apresentada na Figura 15 reflete o desempenho do modelo de detecção visual. As classes de semáforos para pedestres, luz verde pedestres, luz verde veículos, luz vermelha pedestres e luz vermelha veículos obtiveram valores de precisão entre 0,919 e 0,987. O modelo alcançou uma média de precisão mAP *(mean Average Precision)* de 0,949 considerando um limiar de confiança de 0,5 para todas as classes. Esses resultados demonstram capacidade do sistema em equilibrar precisão e *recall*, garantindo baixas taxas de falsos positivos e falsos negativos. Essa confiabilidade é essencial em soluções de

Figura 15 – Curva Precisão-*Recall* para Detecção



Fonte: Elaborada pelo autor.

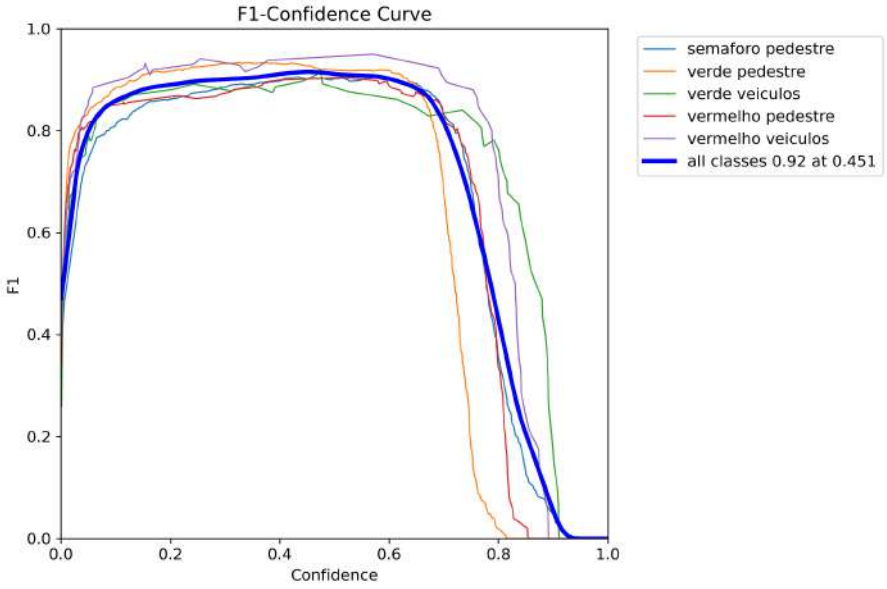
assistência para deficientes visuais, onde erros de detecção podem comprometer segurança e autonomia.

A Figura 16 ilustra a curva F1-*score*-Confiança para a detecção de sinais de trânsito pelo modelo *yris_detect.pt*. As curvas representam o equilíbrio entre precisão e *recall* (valor F1) para a detecção de semáforos para pedestres (azul-escuro), luz verde para pedestres (laranja), luz verde para veículos (verde), luz vermelha para pedestres (vermelho) e luz vermelha para veículos (roxo). Com um *score* F1 que atinge um pico de 0.92 a um nível de confiança de 0.451 para todas as classes (azul), o gráfico demonstra um desempenho excepcionalmente alto do modelo em todos os limiares de confiança.

A matriz de confusão normalizada na Figura 17 proporciona uma visualização quantitativa do desempenho do modelo de detecção. Os valores ao longo da diagonal principal mostram taxas de classificação corretas elevadas para semáforos de pedestres (0.93), luz verde para pedestres (0.96), luz verde para veículos (0.85), e luz vermelha para veículos e pedestres (0.88). Observam-se também valores não negligenciáveis na classificação errônea de objetos como *background*, indicando espaço para melhoria na distinção entre objetos-alvo e o ambiente.

A Figura 18 exibe uma série de gráficos que detalham a progressão do treinamento do modelo *yris_detect.pt*. Os gráficos de perda *(loss)* para caixa *(box)*, classe *(cls)* e dimensão da caixa *(dfI)* nos conjuntos de treino e validação revelam uma tendência decrescente ao longo das 30 épocas, indicando melhoria contínua na capacidade do modelo

Figura 16 – Curva F1-*Score*-Confiança para Detecção



Fonte: Elaborada pelo autor.

de identificar e classificar objetos corretamente. Nota-se também a suavização das curvas (linha pontilhada laranja), sugerindo a efetiva aprendizagem e generalização do modelo. Os gráficos de precisão e *recall* mostram um aumento consistente, refletindo melhorias na identificação de objetos relevantes. Os valores de métrica de área sob a curva *(mAP)* tanto para um único limiar de *Intersection over Union* IoU (mAP@0.5) quanto para múltiplos limiares (mAP@0.5-0.95) crescem substancialmente, confirmando o aprimoramento geral do modelo ao longo do treinamento. Esses resultados quantificam o sucesso do processo de treinamento em aprimorar a precisão do sistema de detecção.

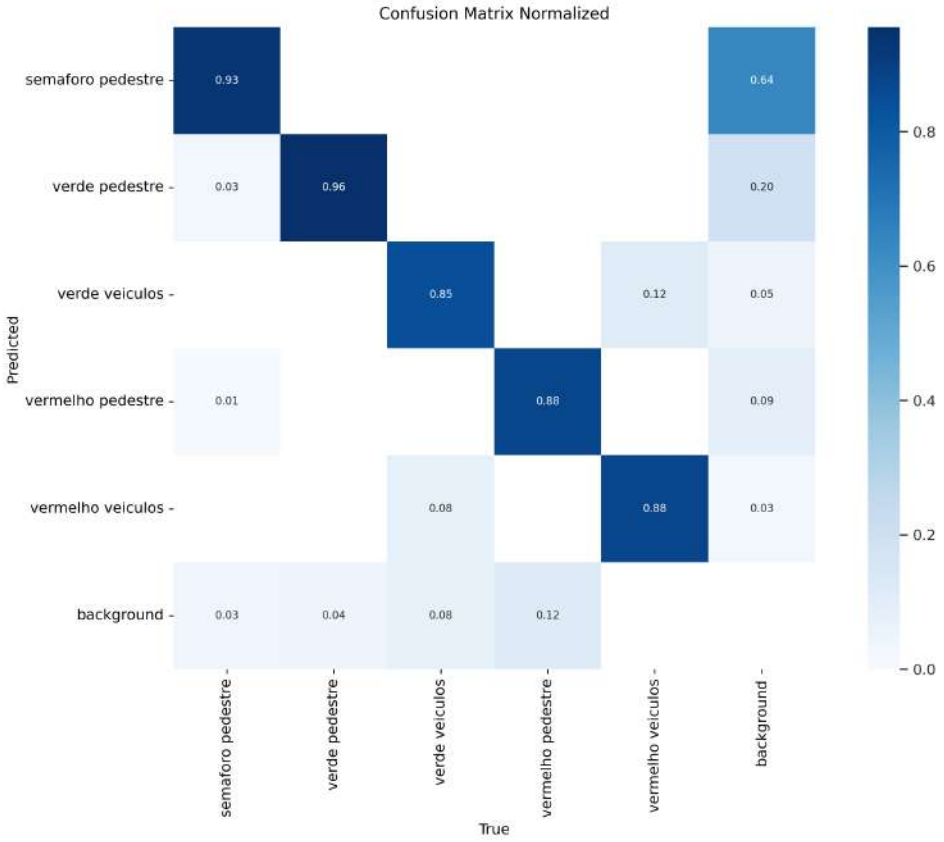
4.2 Métricas Quantitativas de Segmentação Semântica

Valendo-se da arquitetura YOLOv8 da Ultralytics, o modelo *yris_segm.pt* passou por uma avaliação detalhada de seu desempenho nos dados de validação e teste. Foram coletadas métricas centrais, como precisão, revocação e pontuação F1 durante as predições nesses conjuntos, bem como acompanhou-se a evolução desses indicadores no decorrer do treinamento do sistema de segmentação semântica. Essas informações quantitativas permitiram uma análise abrangente das capacidades e limitações do modelo proposto para a tarefa específica de identificação visual de faixas de pedestres e calçadas em imagens.

4.2.1 Avaliação do Modelo *yris_segm.pt*

A Figura 19 exibe a curva de precisão-confiança para o modelo *yris_segm.pt*. As curvas destacam a precisão do modelo na identificação de faixas de pedestres (azul-escuro)

Figura 17 – Matriz de Confusão para Detecção



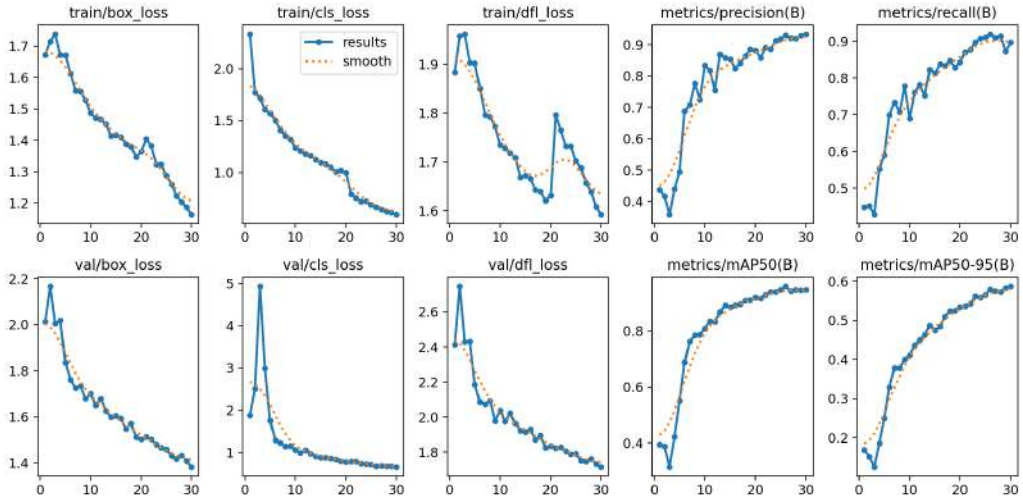
Fonte: Elaborada pelo autor.

e passeios (laranja). Nota-se que a precisão para todas as classes combinadas (linha azul mais espessa) atinge o valor máximo (1.00) a um nível de confiança de 0.920, sugerindo que o modelo é confiável na segmentação precisa destes elementos. Tal desempenho indica uma forte correlação entre a confiança do modelo e a precisão de suas previsões.

Na Figura 20, observa-se a curva *recall*-confiança para o modelo *yris_segm.pt*. A curva para todas as classes (em azul mais espesso) atinge um recall quase perfeito (0.97) mesmo com um nível de confiança de 0.000, ressaltando a capacidade do modelo de identificar corretamente os objetos de interesse independente do grau de confiança. Isso sugere que o modelo é sensível na segmentação dos elementos cruciais para a navegação de pedestres.

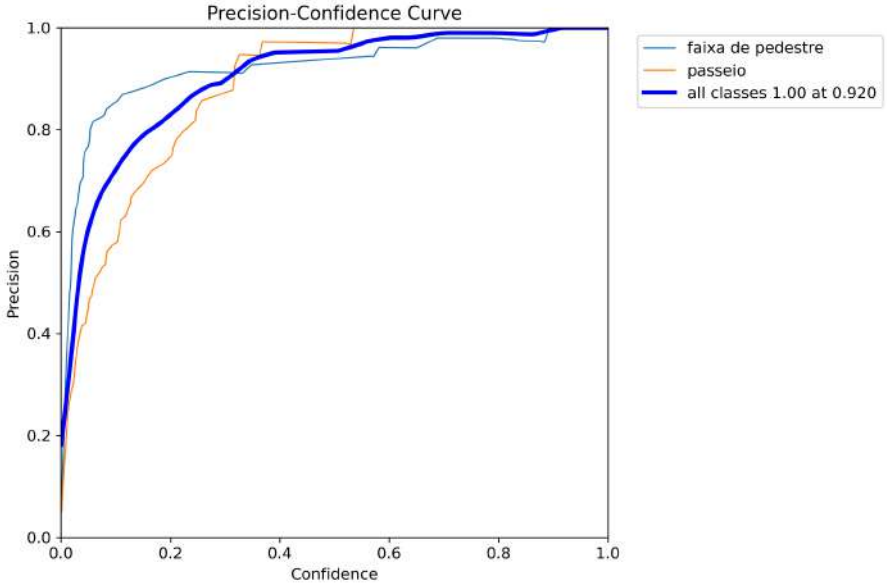
A Figura 21 exibe a curva de precisão-recall para o modelo *yris_segm.pt*. As curvas evidenciam a capacidade do modelo de identificar com precisão faixas de pedestres (em azul-escuro) e passeios (em laranja), com métricas de precisão muito altas (0.956 e 0.958, respectivamente). Notavelmente, o modelo mantém uma média de precisão (mAP) de 0.957 a um limiar de *Intersection over Union* (IoU) de 0.5 para todas as classes. Este

Figura 18 – Evolução do Treinamento para Detecção



Fonte: Elaborada pelo autor.

Figura 19 – Curva Precisão-Confiança para Segmentação

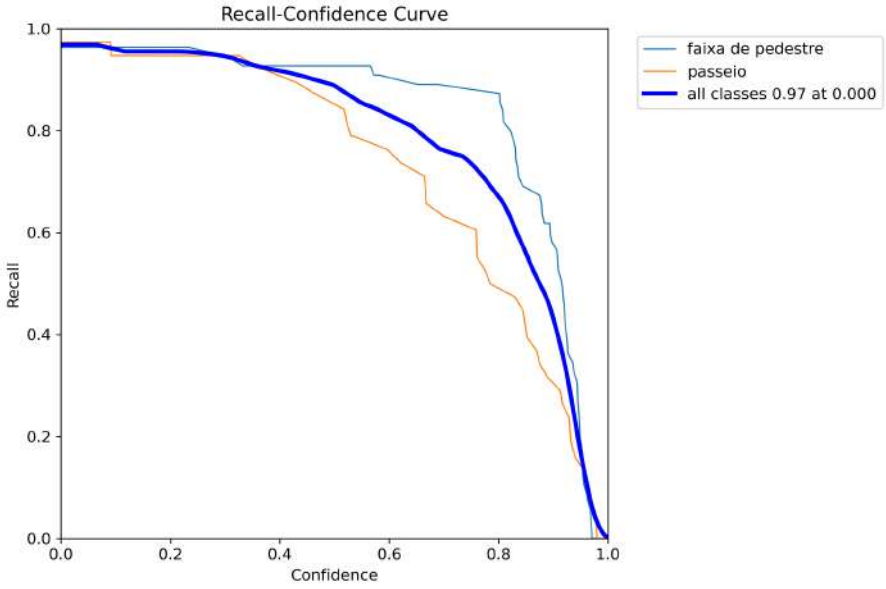


Fonte: Elaborada pelo autor.

desempenho notável ilustra a competência do modelo em fornecer segmentações de alta fidelidade, essenciais para a navegação segura e independente.

A Figura 22 mostra a curva F1-*Confidence* para o modelo *yris_segm.pt*. As curvas representam o *score* F1, que é a média harmônica entre precisão e *recall*, para a detecção de faixas de pedestres (azul-escuro) e passeios (laranja). O modelo alcança um *score* F1

Figura 20 – Curva *Recall*-Confiança para Segmentação



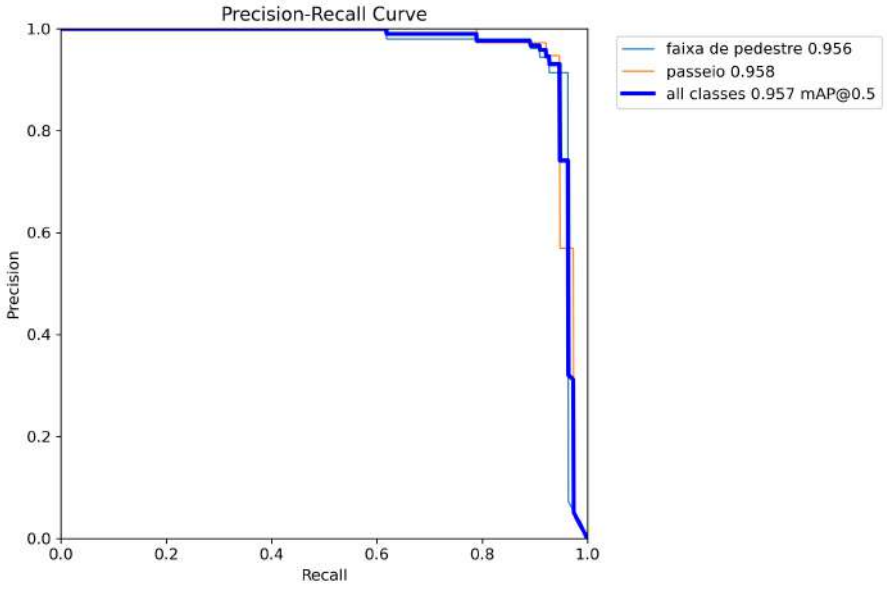
Fonte: Elaborada pelo autor.

elevado para todas as classes (azul) de 0.93 a um nível de confiança de 0.390, demonstrando um equilíbrio notável entre precisão e sensibilidade. Este alto *score* F1 em um nível de confiança relativamente baixo sugere que o modelo é robusto e confiável na identificação dos elementos críticos para a navegação segura de pessoas com visão reduzida, mesmo quando há incertezas nas predições.

A matriz de confusão normalizada na Figura 23 fornece uma avaliação detalhada do modelo *yris_segm.pt*. Os dados mostram uma alta taxa de classificação correta para faixas de pedestres (0.95) e passeios (0.95), indicando uma forte capacidade de discernimento do modelo entre essas duas classes essenciais. No entanto, há uma certa confusão entre as classes, como indicado pelos valores de 0.05 e 0.55 nas células de falsos positivos, onde passeios foram erroneamente classificados como faixas de pedestres e vice-versa. Estes resultados enfatizam a eficácia do modelo em distinguir entre elementos críticos para pedestres, mas também destacam áreas onde o treinamento adicional pode fortalecer a capacidade do modelo de evitar confusões entre classes semelhantes.

A Figura 24 exibe gráficos que detalham o progresso do treinamento do modelo *yris_segm.pt*. Estes gráficos mostram as métricas de perda *(loss)* para box, segmentação *(seg)*, classe *(cls)* e dimensão da caixa *(dfl)* nos conjuntos de treino e validação. Observa-se uma tendência decrescente consistente, indicando melhoria na capacidade do modelo de identificar e classificar corretamente os segmentos de interesse ao longo das iterações de treino.

Figura 21 – Curva Precisão-*Recall* para Segmentação



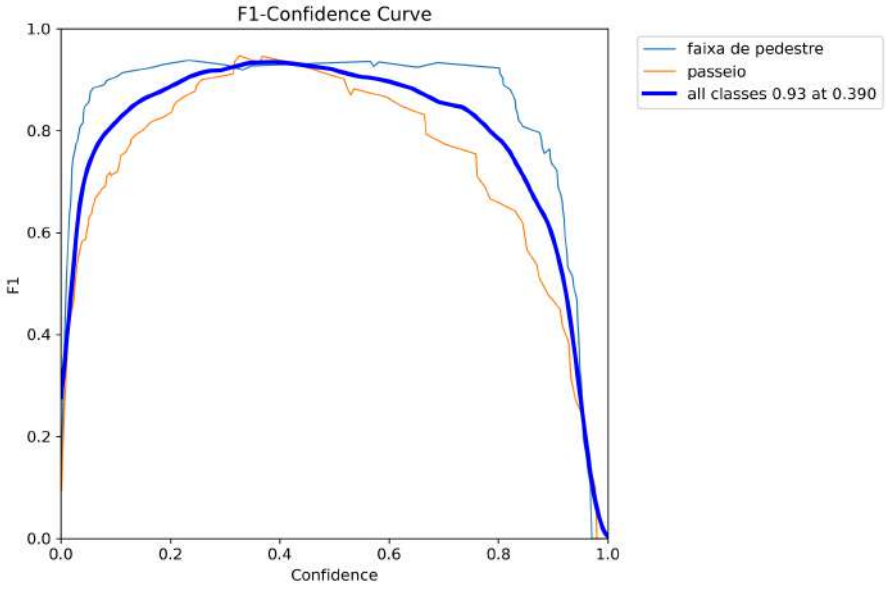
Fonte: Elaborada pelo autor.

Adicionalmente, as curvas de precisão e recall, tanto nos níveis básico *(B)* quanto médio *(M)*, mostram um aumento progressivo, refletindo melhorias na exatidão e na completude com que o modelo identifica os segmentos relevantes. As métricas de área sob a curva *(mAP)* para um único limiar de IoU *(mAP@0.5)* e para uma faixa de limiares *(mAP@0.5-0.95)* também apresentam crescimento substancial, confirmando a eficácia do treinamento na otimização do modelo para a tarefa de segmentação semântica.

4.3 Latência do Sistema

O sistema proposto demonstrou uma latência média de 0,77s para detecção visual e 0,54s para segmentação semântica, condizentes com as exigências de aplicações interativas, conforme detalhado no estudo de Arani *et al.* (2023). Entretanto, a geração de fala exibiu uma latência média de 6,51s por sentença, excedendo o limite de 500ms estabelecido para sistemas de TTS em tempo real, como discutido em Tan *et al.* (2021). Essa limitação se deve à complexidade computacional do modelo VITS utilizado, proporcional ao comprimento do texto de entrada. Para superar esse desafio, otimizações no módulo TTS *(speech.py)*, como o uso de modelos compactos especializados e processamento assíncrono, podem ser essenciais para alcançar a fluidez interativa necessária. Técnicas como essas têm potencial para acelerar a síntese de fala de 36 a 49 vezes, dependendo da especificação do processador e do modelo neural (VAINER; DUšEK, 2020). Assim, apesar das limitações atuais, o *design* modular do sistema facilita evoluções incrementais direcionadas para aplicações em tempo real ou de borda.

Figura 22 – Curva F1-*Score*-Confiança para Segmentação



Fonte: Elaborada pelo autor.

4.4 Análise Qualitativa

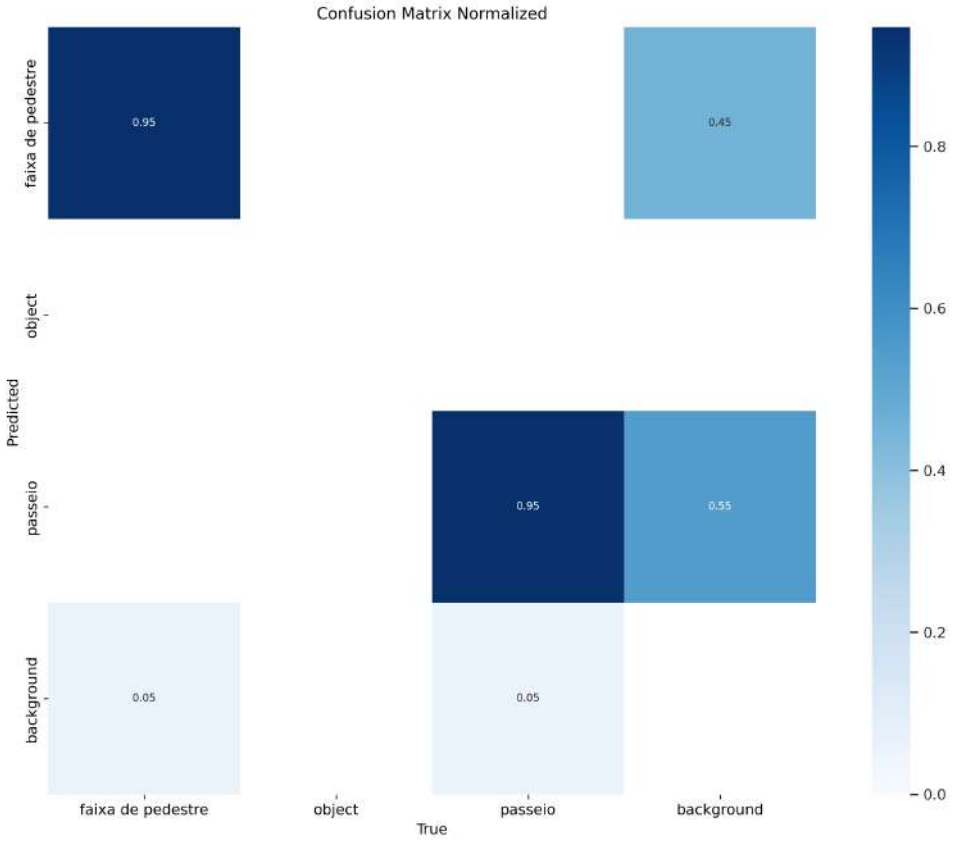
A investigação qualitativa complementa a quantitativa ao prover *insights* sobre capacidades e limitações do sistema sob condições do mundo real. Por meio da execução controlada de casos de uso e análise dos respectivos resultados gerados, evidenciam-se nuances do desempenho.

4.4.1 Casos de Sucesso

Inicialmente, foram executados 5 casos positivos exemplificando o funcionamento desejado sob variadas circunstâncias como apresenta a Figura 25. As saídas incluindo detecções, segmentações e respostas faladas foram inspecionadas quanto à inteligibilidade, coerência e utilidade prática.

Os casos de sucesso ilustrados na Figura 25 demonstram a proficiência dos modelos *yris_detect.pt* e *yris_segm.pt*. No conjunto de imagens superior, dedicada à detecção, o modelo identifica com precisão sinais de trânsito (estados do semáforo) e semáforos, atribuindo-lhes *scores* representativos conforme a precisão. Os *scores*, como 0.64 para 'vermelho veículos', indicam uma alta probabilidade de detecção correta. Já no conjunto de imagens inferior, o modelo exibe sua capacidade de segmentação semântica, diferenciando com sucesso as faixas de pedestres e passeios do ambiente circundante, e atribuindo *scores* como 0.81 para 'faixa de pedestre', evidenciando uma segmentação confiável.

Figura 23 – Matriz de Confusão para Segmentação



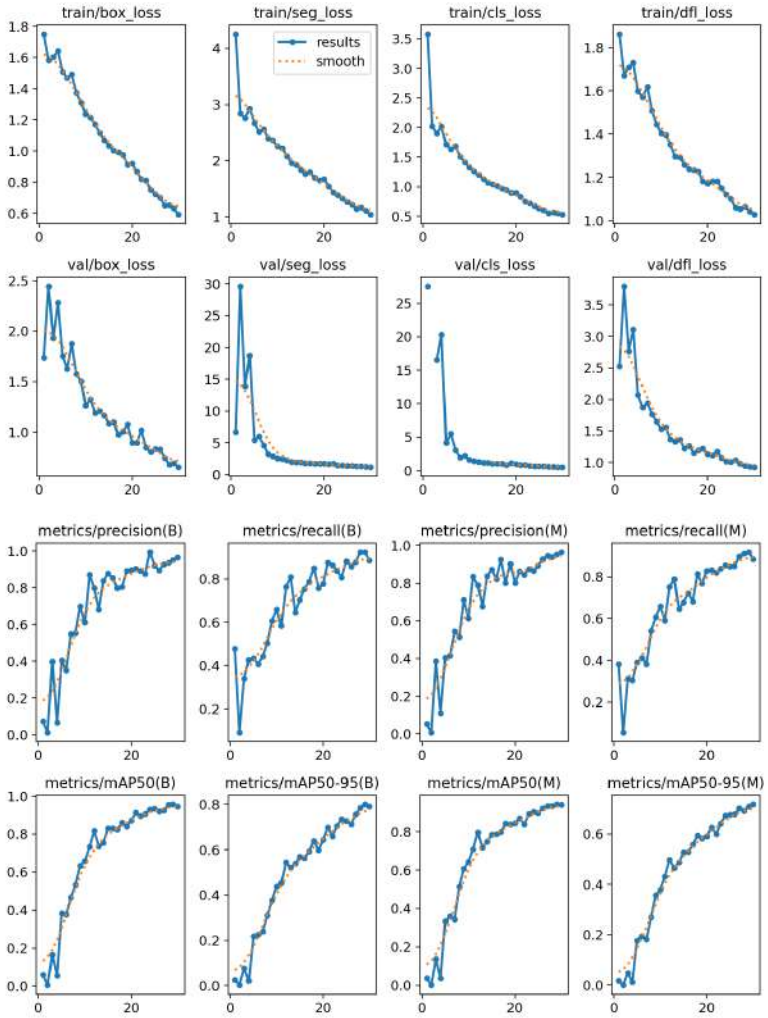
Fonte: Elaborada pelo autor.

4.4.2 Casos de Insucesso

Na sequência, outros 5 casos foram processados contendo situações desafiadoras, onde foi analisada eventual presença de erros, falhas ou limitações, investigando causas prováveis.

Os casos apresentados pela Figura 26 evidenciam cenários onde os modelos *yris_detect.pt* e *yris_segm.pt* enfrentaram dificuldade. Na detecção, houve casos em que os elementos, como semáforos e faixas de pedestre, não foram identificados com a precisão esperada, o que pode ser observado pela ausência de marcações claras em algumas imagens (conjunto de imagens superior). Para a segmentação semântica, embora o modelo tenha corretamente identificado e colorido as faixas de pedestres em vermelho 80% da imagens, algumas áreas como passeios apresentaram baixos *scores* de confiança, como 0.51, sugerindo uma incerteza significativa na classificação. Além disso, a sobreposição de segmentos, particularmente em áreas com alta densidade de elementos urbanos, resultou em algumas imprecisões na delimitação exata dos objetos de interesse. Estes resultados sublinham a necessidade de aprimoramento contínuo na capacidade do modelo de lidar

Figura 24 – Evolução do Treinamento para Segmentação



Fonte: Elaborada pelo autor.

com complexidades do ambiente real, como variações de iluminação e oclusões parciais, para melhor servir às necessidades de navegação e orientação de pessoas com baixa visão.

Figura 25 – Casos de Sucesso

Fonte: Elaborada pelo autor.

Figura 26 – Casos de Insucesso

Fonte: Elaborada pelo autor.

5 CONCLUSÃO

O sistema integrado de visão computacional e síntese de fala proposto neste trabalho representa uma contribuição como ferramenta de assistência a pessoas com deficiência visual em ambientes urbanos. Ao empregar modelos de ponta em detecção, segmentação e processamento de linguagem natural, buscou-se prover funcionalidades avançadas de interpretação visual e *feedback* audível automatizado.

Os experimentos quantitativos revelaram que os modelos visuais atingiram nível satisfatório de desempenho na identificação dos elementos urbanos cruciais, frequentemente com valores de precisão e revocação superiores a 0,95. Isso demonstra competência na extração dos atributos essenciais à orientação de pedestres com visão reduzida.

Contudo, a latência do módulo de síntese de fala, que excedeu 500ms por sentença em média, configura uma limitação significativa. A geração não fluída de áudio compromete a experiência interativa e a agilidade nas instruções sonoras, essenciais à segurança e autonomia dos usuários. As causas residem na complexidade computacional do modelo neural escolhido.

A despeito desta questão, o *pipeline* proposto representa uma PoC (prova de conceito) bem-sucedida. Seu cerne baseado na integração de visão computacional e processamento de linguagem natural revelou-se sólido e com alto potencial de impacto positivo. A modularidade facilita aprimoramentos incrementais no sentido de adequação às restrições temporais.

Assim, no equilíbrio entre pontos positivos e aspectos problemáticos, o sistema destaca-se por seu caráter inovador e multifacetado ao concatenar técnicas de vanguarda para solucionar desafio real e socialmente relevante. Tais qualidades ressaltam seu mérito como trabalho acadêmico e viabilidade como núcleo para futuros desdobramentos.

5.1 Trabalhos futuros

Tendo em vista os gargalos evidenciados na avaliação do sistema, delineiam-se diversas oportunidades de aprimoramento com alto potencial de retorno.

Sobressai a necessidade de reduzir a latência do módulo de síntese de fala para níveis adequados à interação em tempo real, inferior a 500ms. O processamento assíncrono das etapas mais custosas e a substituição por arquiteturas compactas otimizadas para dispositivos móveis figuram entre as alternativas promissoras.

Outra frente relevante seria a realização de testes extensivos com usuários reais em condições ecologicamente válidas. Os *insights* quanto à percepção subjetiva de utilidade e

usabilidade pelo público-alvo orientariam refinamentos ergonômicos e funcionais no sentido das necessidades e preferências humanas.

Por fim, dada a ubiquidade dos *smartphones*, investigar sua utilização como plataforma portátil para o sistema ampliaria sua acessibilidade. A integração com outros sensores e atuadores vestíveis também merece consideração. Essas alternativas consubstanciariam seu potencial como guia pessoal intuitivo no apoio à locomoção independente de pessoas com deficiência visual.

Em conjunto, essas medidas prospectivas visam aprimorar um sistema já promissor, mitigando aspectos problemáticos e enfatizando capacidades diferenciais, rumo à maturação de solução ímpar. Seu êxito demandaria esforço multidisciplinar conjugando ciência da computação, engenharia de software, interação humano-computador e outras expertises. Todavia, o benefício social propiciado por sua consolidação justificaria plenamente tal empreendimento.

REFERÊNCIAS

ARANI, E.; GOWDA, S.; MUKHERJEE, R.; MAGDY, O.; KATHIRESAN, S.; ZONOOZ, B. **A Comprehensive Study of Real-Time Object Detection Networks Across Multiple Domains: A Survey**. 2023.

ATIENZA, R. **EfficientSpeech: An On-Device Text to Speech Model**. 2023.

CHAI, J.; ZENG, H.; LI, A.; NGAI, E. W. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. **Machine Learning with Applications**, v. 6, p. 100134, 2021. ISSN 2666-8270. Available at: https://www.sciencedirect.com/science/article/pii/S2666827021000670.

CHEN, W.; XIE, Z.; YUAN, P.; WANG, R.; CHEN, H.; XIAO, B. A mobile intelligent guide system for visually impaired pedestrian. **The Journal of Systems Software**, v. 195, p. 111546, 2023.

COCO Consortium. **COCO - Common Objects in Context**. 2023. https://cocodataset.org/#home. Accessed: [2023-25-11].

DAVANTHAPURAM, S.; YU, X.; SANIIE, J. Visually impaired indoor navigation using yolo based object recognition, monocular depth estimation and binaural sounds. *In*: **2021 IEEE International Conference on Electro Information Technology (EIT)**. [*S.l.: s.n.*], 2021. p. 173–177.

DHOU, S.; ALNABULSI, A.; AL-ALI, A. R.; ARSHI, M.; DARWISH, F.; ALMAAZMI, S.; ALAMEERI, R. An iot machine learning-based mobile sensors unit for visually impaired people. **Sensors**, v. 22, n. 14, p. 5202, 2022.

GAUTAM, S.; SHARMA, P.; THAPA, K.; UPADHAYA, M. D.; THAPA, D.; KHANAL, S. R.; FILIPE, V. M. de J. **Screening Autism Spectrum Disorder in childrens using Deep Learning Approach : Evaluating the classification model of YOLOv8 by comparing with other models**. 2023.

GAšPAROVIć, B.; MAUšA, G.; RUKAVINA, J.; LERGA, J. Evaluating yolov5, yolov6, yolov7, and yolov8 in underwater environment: Is there real improvement? *In*: **2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)**. [*S.l.: s.n.*], 2023. p. 1–4.

GUPTA, A.; YADAV, D.; RAJ, A.; PATHAK, A. Real-time object detection using ssd mobilenet model of machine learning. **International Journal of Engineering and Computer Science**, v. 12, n. 05, p. 25729–25734, May 2023. Available at: https://www.ijecs.in/index.php/ijecs/article/view/4735.

GURAVAIAH, K.; BHAVADEESH, Y. S.; SHWEJAN, P.; VARDHAN, A. H.; LAVANYA, S. Third eye: Object recognition and speech generation for visually impaired. **Procedia Computer Science**, v. 218, p. 1144–1155, 2023.

HOSSAIN, J.; MOMTAZ, M. **Follow the Soldiers with Optimized Single-Shot Multibox Detection and Reinforcement Learning**. 2023.

HOWARD, A.; SANDLER, M.; CHEN, B.; WANG, W.; CHEN, L.-C.; TAN, M.; CHU, G.; VASUDEVAN, V.; ZHU, Y.; PANG, R.; ADAM, H.; LE, Q. Searching for mobilenetv3. *In*: **2019 IEEE/CVF International Conference on Computer Vision (ICCV)**. [*S.l.: s.n.*], 2019. p. 1314–1324.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. **MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications**. 2017.

IJPH, I. J. of P. H. **The Power of Advocacy: Advancing Vision for Everyone to Meet the Sustainable Development Goals**. 2022. [Online; accessed on 31 May 2023]. Available at: https://www.ssph-journal.org/articles/10.3389/ijph.2022.1604595/full.

ISLAM, R. B.; AKHTER, S.; IQBAL, F.; RAHMAN, M. S. U.; KHAN, R. Deep learning based object detection and surrounding environment description for visually impaired people. **Heliyon**, v. 9, n. 1, p. e16924, 2023.

KURATA, G.; AUDHKHASI, K.; THOMAS, S.; CHERN, A.; RAMABHADRAN, B.; TSIARTAS, A.; LI, B.; PICHENY, M. Knowledge distillation from ensembles for efficient high-quality neural tts. *In*: **Proc. Interspeech 2021**. [*S.l.: s.n.*], 2021. p. 1123–1127.

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. Ssd: Single shot multibox detector. *In*: . [*S.l.: s.n.*], 2016. To appear. Available at: http://arxiv.org/abs/1512.02325.

MARTINEZ-ALPISTE, I.; GOLCARENARENJI, G.; WANG, Q. *et al.* Smartphone-based real-time object recognition architecture for portable and constrained systems. **J Real-Time Image Proc**, Springer, v. 19, p. 103–115, 2022. Available at: https://doi.org/10.1007/s11554-021-01164-1.

MICROSOFT, M. N. **The eye in AI**. 2022. [Online; accessed on 31 May 2023]. Available at: https://www.microsoft.com/en-us/ai/seeing-ai.

OORD, A. v. d.; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A.; KAVUKCUOGLU, K. Wavenet: A generative model for raw audio. **arXiv preprint arXiv:1609.03499**, 2016.

PRATAP, V.; TJANDRA, A.; SHI, B.; TOMASELLO, P.; BABU, A.; KUNDU, S.; ELKAHKY, A.; NI, Z.; VYAS, A.; FAZEL-ZARANDI, M.; BAEVSKI, A.; ADI, Y.; ZHANG, X.; HSU, W.-N.; CONNEAU, A.; AULI, M. Scaling speech technology to 1,000+ languages. **arXiv**, 2023.

RangeKing. **RangeKing**. [*S.l.: s.n.*]: GitHub, 2023. https://github.com/RangeKing.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. *In*: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [*S.l.: s.n.*], 2016. p. 779–788.

REIS, D.; KUPEC, J.; HONG, J.; DAOUDI, A. **Real-Time Flying Object Detection with YOLOv8**. 2023.

REN, Y.; HU, C.; TAN, X.; QIN, T.; ZHAO, S.; ZHAO, Z.; LIU, T.-Y. **FastSpeech 2: Fast and High-Quality End-to-End Text to Speech**. 2022.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. *In*: **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [*S.l.: s.n.*], 2018. p. 4510–4520.

SHEN, J.; PANG, R.; WEISS, R. J.; SCHUSTER, M.; JAITLY, N.; YANG, Z.; CHEN, Z.; ZHANG, Y.; WANG, Y.; SKERRV-RYAN, R.; SAUROUS, R. A.; AGIOMVRGIANNAKIS, Y.; WU, Y. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *In*: **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [*S.l.: s.n.*], 2018. p. 4779–4783.

SOTELO, J.; MEHRI, S.; KUMAR, K.; SANTOS, J. F.; KASTNER, K.; COURVILLE, A.; BENGIO, Y. Char2wav: End-to-end speech synthesis. *In*: **ICLR Workshop**. [*S.l.: s.n.*], 2017.

TACHIBANA, H.; UENOYAMA, K.; AIHARA, S. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. *In*: **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. IEEE, 2018. Available at: https://doi.org/10.1109%2Ficassp.2018.8461829.

TAN, X.; QIN, T.; SOONG, F.; LIU, T.-Y. **A Survey on Neural Speech Synthesis**. 2021.

TERVEN, J.; CORDOVA-ESPARZA, D. **A Comprehensive Review of YOLO: From YOLOv1 and Beyond**. 2023.

Ultralytics. **Ultralytics/ultralytics: Ultralytics YOLO, PyTorch, and TensorFlow production-ready CV research code**. [*S.l.: s.n.*]: GitHub, 2023. https://github.com/ultralytics/ultralytics.

VAINER, J.; DUšEK, O. **SpeedySpeech: Efficient Neural Speech Synthesis**. 2020.

VÉSTIAS, M. P.; DUARTE, R. P.; SOUSA, J. T. de; NETO, H. C. Moving deep learning to the edge. **Algorithms**, MDPI, v. 13, n. 5, p. 125, 2020.

VISIO.AI. **The 100 Most Popular Computer Vision Applications in 2023**. 2023. [Online; acessado em 28/07/23]. Available at: https://viso.ai/applications/computer-vision-applications/.

WANG, W.; LI, Y.; ZOU, T.; WANG, X.; YOU, J.; LUO, Y. A novel image classification approach via dense-mobilenet models. **Mobile Information Systems**, v. 2020, p. 1–8, 2020. Article ID 7602384.

WANG, Y.; SKERRY-RYAN, R.; STANTON, D.; WU, Y.; WEISS, R. J.; JAITLY, N.; YANG, Z.; XIAO, Y.; CHEN, Z.; BENGIO, S.; LE, Q.; AGIOMYRGIANNAKIS, Y.; CLARK, R.; SAUROUS, R. A. **Tacotron: Towards End-to-End Speech Synthesis**. 2017.

WHO, W. H. O. **Blindness and vision impairment**. 2022. Acessado em: 31 de maio de 2023. Available at: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

YRIS. Open Source Dataset, **Estado Semaforo Veiculo-Pedestre Dataset**. Roboflow, 2023. https://universe.roboflow.com/yris/estado-semaforo-veiculo-pedestre. Visited on 2023-11-25. Available at: https://universe.roboflow.com/yris/estado-semaforo-veiculo-pedestre.

YRIS. Open Source Dataset, **Faixa de Pedestre-Passeio**$_v 2 Dataset. Roboflow$, 2023.. Visited on 2023-11-25. Available at: https://universe.roboflow.com/yris/faixa-de-pedestre-passeio_v2.

ZHANG, P.; LIU, B. **Commonsense Knowledge Assisted Deep Learning with Application to Size-Related Fine-Grained Object Detection**. 2023.

ŁAńCUCKI, A. **FastPitch: Parallel Text-to-speech with Pitch Prediction**. 2021.