

VITOR HUGO FAZOLI DA SILVA

SEGMENTAÇÃO DE IMAGENS DE MÉDIA E BAIXA
RESOLUÇÃO POR MEIO DO MODELO SEGMENT
ANYTHING

Orientador: Prof Dr.Gabriel
Marcelino Alves

SÃO JOÃO DA BOA VISTA

2024

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Objetivos	6
1.1.1	Objetivo Geral	6
1.1.2	Objetivos Específicos	6
2	REVISÃO DA LITERATURA	7
2.1	Segmentação Baseada em Camadas	7
2.2	Segmentação baseada em blocos	8
2.3	Segment Anything	9
2.3.1	Visão geral do SAM	10
2.4	Trabalhos Correlatos	11
2.4.1	Segmentation by grouping junctions	11
2.4.2	Monocular depth estimation based on deep learning: An overview	12
3	METODOLOGIA	13
3.1	Tecnologias utilizadas	15
3.2	Procedimentos e técnicas	16
3.2.1	Base de dados	17
3.3	Métricas Utilizadas	18
4	ANÁLISE DOS RESULTADOS	21
4.1	Desenvolvimento do sistema	21
4.2	Agilidade e Eficiência	23
4.3	Análise de resultados	24
5	CONCLUSÕES	28
	REFERÊNCIAS	29

RESUMO

Este Trabalho de Conclusão de Curso apresenta uma pesquisa sobre a segmentação de imagens utilizando o modelo Segment Anything Model (SAM). O objetivo central é avaliar a eficácia do SAM na segmentação de imagens de média e baixa resolução, destacando sua relevância para aplicações diversas na área de visão computacional. A pesquisa inicia-se com uma revisão da literatura sobre segmentação de imagens, abordando as principais técnicas e abordagens existentes. Posteriormente, é detalhada a metodologia empregada no estudo, que incluiu a criação de uma base de imagens composta por 50 pixel artes de diferentes resoluções e contextos visuais. Os resultados obtidos evidenciam que o SAM apresenta um desempenho satisfatório na segmentação de imagens de média e baixa resolução, mesmo em cenários com presença de ruído ou variações significativas de iluminação.

Palavras-chave: Segmentação de imagens. Modelo Segment Anything. Média e baixa resolução.

ABSTRACT

This thesis presents research on image segmentation using the Segment Anything Model (SAM). The main objective is to evaluate the effectiveness of SAM in segmenting medium and low-resolution images, highlighting its relevance for various applications in the field of computer vision. The research begins with a literature review on image segmentation, discussing the main existing techniques and approaches. The methodology employed in the study is then detailed, which included the creation of an image database composed of 50 pixel arts with different resolutions and visual contexts. The results obtained show that SAM performs satisfactorily in segmenting medium and low-resolution images, even in scenarios with noise or significant lighting variations.

Keywords: Image segmentation. Segment Anything Model. Medium and low resolution.

1 INTRODUÇÃO

A segmentação de imagens tem se mostrado crucial na visão computacional, desempenhando um papel fundamental em diversas aplicações, como diagnóstico médico, monitoramento ambiental, segurança pública e condução autônoma. A capacidade de dividir uma imagem em regiões distintas permite extrair informações específicas e realizar análises mais precisas. No entanto, essa tarefa é frequentemente desafiada por fatores como ruído, variações de iluminação e baixa qualidade das imagens, especialmente em contextos onde estamos entre as baixas e médias resoluções de imagens.

Além de identificar objetos ou regiões de interesse, a segmentação desempenha um papel preparatório importante em outras etapas de processamento de imagens. No caso da melhoria de iluminação, por exemplo, segmentar as áreas relevantes da imagem primeiro é essencial para aplicar correções direcionadas, minimizando interferências em partes irrelevantes e preservando a integridade visual. Essa abordagem é especialmente valiosa em aplicações que exigem alto nível de detalhe, como análise médica ou inspeção industrial, onde pequenas falhas de iluminação podem comprometer os resultados.

A iluminação, por sua vez, é um problema recorrente em processamento de imagens. Variações de intensidade, sombras e reflexos podem dificultar não apenas a segmentação, mas também qualquer análise posterior. Por isso, estratégias que combinam segmentação com técnicas avançadas de equalização, balanceamento de cores ou iluminação adaptativa têm o potencial de melhorar significativamente a qualidade e a utilidade das imagens processadas. Este trabalho busca explorar como a segmentação inicial pode facilitar esses ajustes, destacando sua importância como etapa crítica em fluxos complexos de processamento.

Com o advento das inteligências artificiais modernas, especialmente os modelos de *deep learning*, surge uma nova perspectiva para resolver essas limitações. Neste contexto, o Segment Anything Model (SAM) emerge como uma tecnologia inovadora, propondo uma abordagem generalista e flexível para a segmentação de imagens.

Desenvolvido com um conjunto de dados extenso de 1 bilhão de máscaras provenientes de 11 milhões de imagens, o SAM representa um avanço significativo na capacidade de identificar e isolar objetos em diferentes contextos. O presente trabalho tem como objetivo principal investigar a eficácia do SAM na segmentação de imagens, com ênfase em imagens de média e baixa resolução. Através de uma análise rigorosa que combina métricas quantitativas como *Mean Squared Error* (MSE) e *Normalized Cross-Correlation* (NCC), busca-se compreender as potencialidades e limitações deste modelo de inteligência artificial.

Diferentemente das abordagens tradicionais que requerem treinamento específico para cada conjunto de dados, o SAM propõe uma metodologia de *zero-shot*, capaz de realizar segmentações precisas sem necessidade de retreinamento. Esta característica o torna particularmente interessante para aplicações que demandam adaptabilidade e eficiência.

Ao explorar as capacidades do SAM, esta pesquisa não apenas avalia sua performance técnica, mas também contribui para a compreensão mais ampla das possibilidades emergentes na intersecção entre inteligência artificial e processamento de imagens. Os resultados obtidos podem fornecer insights valiosos para pesquisadores e profissionais que buscam soluções inovadoras em segmentação de imagens.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo do trabalho é avaliar a segmentação de imagens de média e baixa resolução por meio do modelo de inteligência artificial Segment Anything.

1.1.2 Objetivos Específicos

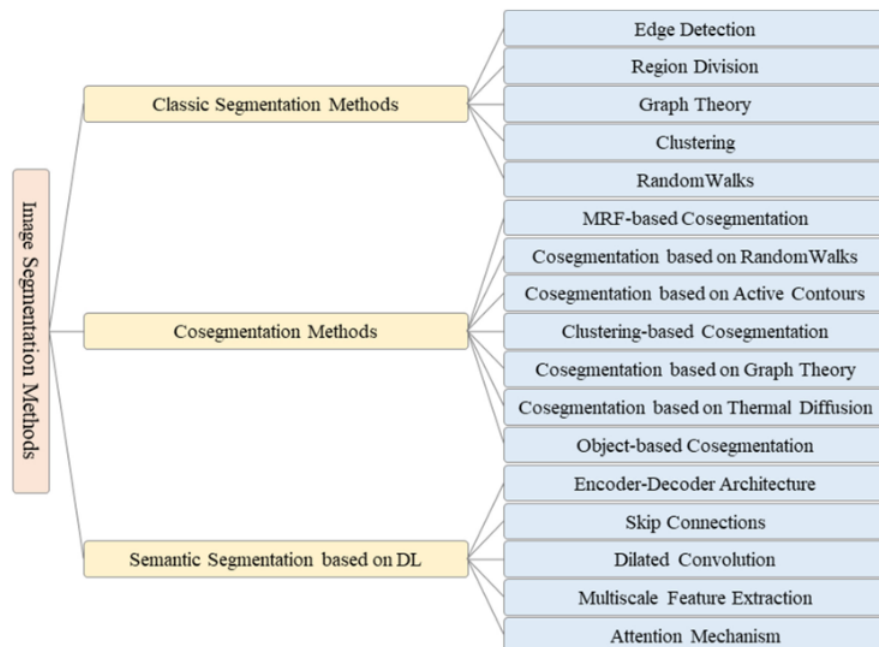
- Preparar uma base de imagens de média e baixa resolução.
- Preparar imagens de referência para avaliação da segmentação.

2 REVISÃO DA LITERATURA

A segmentação de imagem, fundamentada em conceitos matemáticos, antecede o desenvolvimento das inteligências artificiais modernas e oferece uma vasta gama de métodos que exploram a relação entre *pixels* adjacentes para identificar similaridades e descontinuidades na imagem. Segundo Taneja, Ranjan e Ujjlayan (2015) e Zaitoun e Aqel (2015), essas técnicas se dividem principalmente em segmentações baseadas em camadas e em blocos, cada uma com abordagens e subcategorias específicas, como a detecção de bordas e a análise de profundidade e aparência dos objetos.

Já segundo Yu et al. (2023), os métodos de segmentação de imagens podem ser organizados em três categorias principais: segmentação clássica, co-segmentação e segmentação semântica baseada em *Deep Learning*. Cada uma dessas categorias aborda a segmentação com diferentes técnicas e algoritmos, cada qual voltado para necessidades específicas de processamento e aplicação. A Figura 1 mostra esses diferentes métodos de segmentação, organizando-os em categorias principais e destacando os principais algoritmos em cada abordagem.

Figura 1 – Digrama de tipos de segmentação de imagem



2.1 Segmentação Baseada em Camadas

No documento feito por Yu et al. (2023) a **segmentação clássica** inclui técnicas que se concentram em características locais da imagem, como bordas e regiões como

citado pelos outros autores também, mas ele aborda alguns conceitos diferentes como a **co-segmentação**, que foca na extração de objetos comuns em múltiplas imagens, considerando informações de contexto para identificar objetos similares. Por fim, a **segmentação semântica com *Deep Learning*** incorpora redes neurais profundas que identificam objetos com precisão ao aprender características complexas diretamente de grandes conjuntos de dados anotados.

A segmentação baseada em camadas, conforme proposta por Yang et al. (2012), é uma abordagem que organiza objetos em uma hierarquia de camadas, onde cada camada representa um objeto ou uma parte de objeto na imagem. Essa técnica permite que objetos sejam compostos e organizados em uma estrutura que leva em consideração tanto a aparência quanto a profundidade relativa de cada objeto na cena.

O método de Yang et al. (2012) utiliza máscaras de forma derivadas de detectores de objetos, as quais são compostas em camadas para criar uma segmentação que engloba tanto rótulos de classe quanto rótulos de instância. Com isso, a abordagem considera a organização espacial e relacional dos objetos, permitindo a detecção de elementos mesmo quando oclusos ou parcialmente visíveis, ao inferir o layout da cena com base em uma distribuição probabilística das camadas.

Cada camada é ordenada em profundidade, atribuindo objetos com maior pontuação em primeiro plano e outros objetos em segundo plano. Essa estruturação permite que o sistema identifique transições e relacionamentos entre diferentes camadas, ajudando a criar segmentações precisas, principalmente em cenários complexos onde múltiplas instâncias de objetos podem se sobrepor ou interagir de forma dinâmica. A partir dessa hierarquia, o modelo é capaz de reconciliar informações de alto nível com detalhes de baixo nível, integrando informações de cor e textura com os contornos e formatos dos objetos segmentados.

A segmentação em camadas não só melhora a precisão da rotulação de pixels, mas também possibilita uma compreensão mais profunda das interações entre objetos. Essa característica é particularmente vantajosa para a segmentação de imagens que contêm múltiplas instâncias e classes, onde a separação de camadas facilita a correta identificação de fronteiras e a resolução de ambiguidades visuais.

2.2 Segmentação baseada em blocos

Já os métodos de Segmentação de Imagem Baseados em Blocos, segundo Zaitoun e Aqel (2015), podem ser categorizados em duas propriedades principais: descontinuidade e similaridade. Essas técnicas de segmentação dividem-se em várias abordagens, incluindo aquelas que se concentram em características como cor, continuidade, similaridade e bordas, permitindo a criação de subcategorias específicas de acordo com o processo de divisão

aplicado.

Entre as abordagens principais, destacam-se os métodos baseados em bordas abordado por Mayangky et al. (2024), que se concentram em detectar descontinuidades na intensidade da imagem para identificar transições abruptas entre diferentes objetos ou regiões. Alguns dos métodos clássicos de detecção de bordas incluem a **Detecção de Bordas de Roberts**, que utiliza operadores cruzados para calcular o gradiente espacial e é amplamente valorizada pela simplicidade e eficiência, sendo ideal para aplicações que exigem baixo custo computacional; a **Detecção de Bordas de Prewitt**, que calcula a magnitude e orientação das bordas usando uma máscara de convolução 3x3, tornando-se mais robusta do que o método de Roberts, embora ainda suscetível a ruídos; e a **Detecção de Bordas de Sobel**, que aplica uma máscara 3x3 rotacionada em 90° para suavizar ruídos enquanto calcula o gradiente das bordas, sendo amplamente utilizada devido à sua eficácia na detecção de bordas.

Com os avanços na inteligência artificial e nas técnicas de computação evolutiva, surgiram métodos de detecção de bordas mais sofisticados. Entre eles, o método **Baseado em Lógica Fuzzy** que é descrito por Gonzalez, Melin e Castillo (2017) como uma lógica que utiliza conjuntos fuzzy que permitem que cada pixel pertença a múltiplas regiões, oferecendo flexibilidade em imagens com transições suaves. Já o método **Baseado em Algoritmos Genéticos** inspira-se na teoria da evolução, utilizando processos de seleção, cruzamento e mutação para identificar as bordas de maneira eficiente, sendo particularmente útil em padrões complexos. O método **Baseado em Redes Neurais**, por sua vez, utiliza redes neurais artificiais treinadas para aprender padrões de bordas ajustando os pesos entre suas camadas, sendo altamente eficaz na detecção de bordas em cenários com variabilidade de padrões.

A integração do big data com o avanço dos hardwares modernos permitiu que a inteligência artificial atingisse novos patamares de desempenho na segmentação de imagens. A habilidade de processar grandes volumes de dados possibilita que a IA identifique padrões e características em imagens de maneira rápida e precisa. Segundo Carvalho (2021), o uso de *deep learning* e de técnicas de machine learning aprimora ainda mais essa precisão, tornando a segmentação detalhada e eficiente, essencial para diversas aplicações inovadoras.

2.3 Segment Anything

Inserido nesse contexto e com base no artigo criado por (Kirillov et al., 2023) o modelo Segment Anything (SAM) é um novo avanço na área de segmentação de imagens, proporcionando uma abordagem generalista que visa resolver diversos problemas de segmentação utilizando diferentes tipos de dados e prompts. Com um conjunto de dados

extenso e uma arquitetura inovadora, o SAM permite a criação de máscaras de segmentação em tempo real e com capacidade de generalização para novos conjuntos de dados.

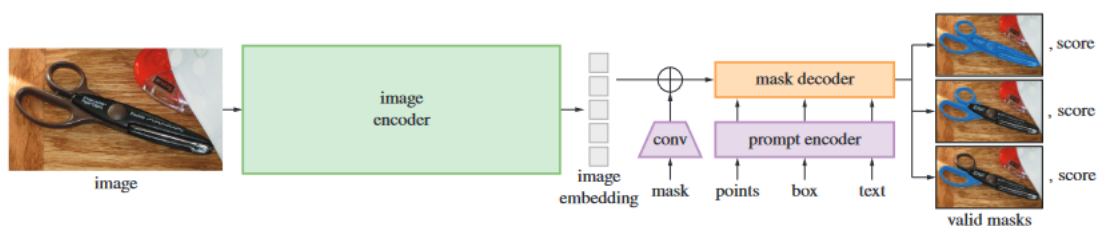
A segmentação de imagens tem sido revolucionada pelo surgimento de métodos de *deep learning*, como o Segment Anything Model (SAM), que permite uma segmentação versátil e de alta escala. A segmentação de imagem visa identificar e isolar regiões ou objetos dentro de uma imagem, o que é essencial em várias aplicações de visão computacional. Métodos convencionais, entretanto, exigiam grande quantidade de dados rotulados e intensa supervisão manual. O SAM introduziu uma abordagem de zero-shot, possibilitando segmentações precisas em novos conjuntos de dados sem a necessidade de re-treinamento, utilizando prompts como caixas delimitadoras e pontos para indicar regiões de interesse (Ke et al., 2023).

Apesar de seu impacto, o SAM apresenta limitações ao lidar com bordas complexas e objetos finos. Para superar essas limitações, o HQ-SAM foi proposto como uma extensão de alta qualidade do SAM, incorporando um token de saída especializado e técnicas de fusão de características globais e locais para melhorar a definição das bordas e a precisão da segmentação em objetos detalhados e sobrepostos. Esses avanços garantem que o HQ-SAM preserve a flexibilidade e a generalização zero-shot do SAM original enquanto aumenta significativamente a acurácia da segmentação (Ke et al., 2023).

2.3.1 Visão geral do SAM

A arquitetura do SAM é composta por três principais componentes: um codificador de imagens, um codificador de prompts e um decodificador de máscaras. O codificador de imagens gera um embedding (representação) da imagem que pode ser reutilizado para diferentes prompts. O codificador de prompts permite que o SAM aceite uma variedade de entradas, como pontos, caixas delimitadoras, e até mesmo textos para identificar as áreas de interesse na imagem. O decodificador de máscaras, então, gera as máscaras de segmentação apropriadas a partir dessas entradas. O SAM é projetado para ser eficiente, processando prompts em tempo real, isso é mostrado na Figura 2

Figura 2 – Digrama mostrando o processo realizado pelo SAM



Fonte: <https://github.com/facebookresearch/segment-anything?tab=readme-ov-file>

Para treinar o SAM, foi desenvolvido o conjunto de dados SA-1B, que contém mais de 1 bilhão de máscaras de segmentação provenientes de 11 milhões de imagens.

Essas imagens são diversas, de alta resolução, e foram obtidas respeitando questões de privacidade. Esse conjunto de dados é, até o momento, o maior já construído para a tarefa de segmentação, superando os bancos de dados existentes em termos de diversidade e volume. O SAM é capaz de gerar máscaras de segmentação automaticamente a partir dessas imagens, tornando-se uma ferramenta poderosa para a criação de novos conjuntos de dados e modelos de visão computacional.

O Segment Anything Model (SAM) foi amplamente testado em tarefas de segmentação, como a segmentação de objetos a partir de pontos, detecção de bordas e segmentação de objetos sobrepostos, demonstrando forte desempenho e flexibilidade. Em cenários complexos, o SANErf-HQ aprimora o SAM para segmentação em 3D de alta qualidade, utilizando prompts e campos de densidade para garantir consistência entre múltiplas visualizações. Os resultados mostram uma melhoria significativa em relação aos métodos anteriores, mantendo generalização zero-shot e alta acurácia ao longo de diferentes conjuntos de dados (Liu et al., 2024).

O SAM apresenta várias vantagens em relação a métodos tradicionais de segmentação, como o Crescimento de Regiões. Enquanto os métodos tradicionais exigem parâmetros e pré-processamento mais específicos para funcionar corretamente, o SAM, por meio de sua arquitetura flexível e escalável, permite segmentação em uma ampla gama de imagens com menos intervenção manual. Além disso, sua capacidade de processar múltiplas máscaras para um único prompt o torna eficaz para cenários onde a ambiguidade está presente, como a detecção de partes de objetos sobrepostos.

2.4 Trabalhos Correlatos

O campo da visão computacional tem experimentado avanços significativos, especialmente nas áreas de segmentação de imagens e estimação de profundidade monocular. Com o advento do *Deep Learning*, pesquisadores têm alcançado resultados expressivos em aplicações como diagnóstico médico, navegação autônoma e reconstrução em três dimensões.

Nesta seção, será apresentado os trabalhos mais relevantes que fundamentam nossa proposta, estabelecendo o estado da arte atual e identificando oportunidades de contribuição para o avanço do campo. A seguir, são discutidos os principais trabalhos que influenciaram diretamente nossa abordagem.

2.4.1 Segmentation by grouping junctions

A segmentação de imagens tem evoluído significativamente com a introdução de métodos baseados em aprendizado profundo. Minaee et al. (2020) realizam uma extensa revisão sobre métodos de segmentação com redes neurais, destacando abordagens como

redes totalmente convolucionais (FCN), arquiteturas encoder-decoder e redes piramidais de múltiplas escalas, que são amplamente adotadas. Estes métodos permitiram avanços consideráveis na precisão e na generalização dos modelos, especialmente para aplicações em imagens médicas e na análise de cenas complexas, onde a precisão é crucial.

2.4.2 Monocular depth estimation based on deep learning: An overview

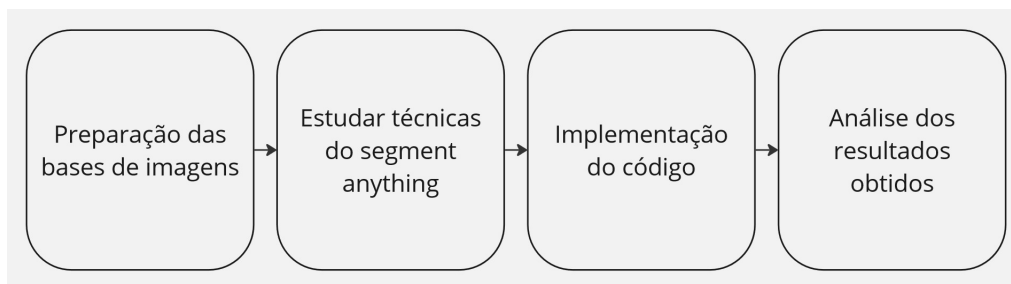
A estimação de profundidade monocular é um problema fundamental em visão computacional. Métodos baseados em aprendizado profundo (deep learning) têm sido estudados e alcançaram resultados promissores. Existem três categorias de métodos: supervisionados, não supervisionados e semi-supervisionados. Arquiteturas de rede neural como CNNs, RNNs e GANs são utilizadas. O artigo de (Zhao et al., 2020) apresenta uma visão geral desses métodos e discute desafios e oportunidades.

3 METODOLOGIA

Nesta seção, serão apresentados os materiais e métodos utilizados no estudo, detalhando a metodologia adotada e os procedimentos experimentais realizados. O processo inicia com o levantamento de dados necessários para a análise e estudo, seguido de uma avaliação criteriosa dos processos envolvidos. Com base nessas análises, será estruturado o desenvolvimento do trabalho, garantindo que todas as etapas sejam realizadas de forma sistemática e alinhadas aos objetivos do projeto.

Serão discutidos os principais passos estabelecidos para a execução completa do projeto idealizado, conforme ilustrado na Figura 3.

Figura 3 – Diagrama dos passos para a conclusão da ferramenta



Fonte: Elaborado pelo autor

O diagrama apresentado acima aborda os passos necessários para a realização da segmentação de imagem

A realização da segmentação de imagem é um processo complexo que envolve várias etapas importantes como foi mostrado o diagrama na Figura 3. Uma das principais técnicas utilizadas atualmente é o Zero-Shot, que permite às inteligências artificiais aprender a segmentar imagens sem a necessidade de exemplos prévios. Isso é possível graças ao uso de modelos de linguagem treinados em grandes conjuntos de dados, que permitem a transferência de conhecimento para novas tarefas como exemplificado por Cao et al. (2023).

Antes de nos aprofundar nos conceitos do segment Anything é importante mostrar alguns conceitos sobre as técnicas de segmentação usadas, pois existem diversos tipos de segmentação usados por inteligências artificiais, cada um com objetivos e abordagens específicas para a análise de imagens. A segmentação de imagens é essencialmente o processo de dividir uma imagem em diferentes regiões ou segmentos, com base em características como cor, textura ou bordas, para facilitar a análise e o entendimento do conteúdo da imagem. Entre as abordagens mais comuns estão a segmentação semântica, a segmentação instanciada e a segmentação baseada em objetos.

A segmentação semântica é o processo em que cada pixel da imagem é classificado

em uma categoria específica, como céu, água, chão, etc. Esse tipo de segmentação não diferencia as instâncias de objetos dentro de uma mesma categoria. Por exemplo, em uma imagem de um campo com várias árvores, todas as árvores seriam tratadas como um único objeto, sem distinção entre elas.

Já a segmentação instanciada vai além, separando as diferentes instâncias de um mesmo tipo de objeto. Ou seja, mesmo que uma imagem contenha múltiplas árvores, a IA seria capaz de distinguir uma árvore da outra, tratando-as como objetos individuais. Esse tipo de segmentação é crucial para tarefas que exigem uma compreensão mais detalhada da cena, como a análise de objetos em vídeos ou imagens complexas.

A segmentação baseada em objetos, por sua vez, foca na identificação e separação dos objetos em uma imagem com base em suas características visuais. Diferentemente da segmentação semântica, que classifica pixels de maneira geral, a segmentação baseada em objetos busca identificar e delimitar objetos específicos dentro da cena, como pessoas, veículos ou animais. Esse tipo de segmentação é frequentemente utilizado em sistemas de visão computacional para tarefas como reconhecimento de objetos, detecção e rastreamento.

Essas técnicas de segmentação são utilizadas em conjunto com modelos avançados de aprendizado de máquina, como o Segment Anything Model (SAM), que aproveita grandes bases de dados e métodos como o Zero-Shot Learning para adaptar seus algoritmos a novos tipos de imagens e cenários. Cada uma dessas abordagens tem sua aplicação dependendo do tipo de tarefa a ser realizada e das necessidades do projeto, com a segmentação instanciada e a baseada em objetos oferecendo uma compreensão mais detalhada e precisa dos elementos presentes nas imagens.

Um exemplo de modelo que utiliza essa técnica é o Segment Anything, que foi treinado em uma base de 11 milhões de imagens¹. Essa base é composta por imagens que tem a média de resolução de 1500×2250 . Além disso, o Segment Anything também foi treinado em diferentes tipos de imagens, incluindo objetos, cenas e ações, o que o torna um modelo versátil e capaz de lidar com diferentes tipos de segmentação.

No entanto, criar uma base de imagens própria para treinar um modelo de segmentação pode ser um trabalho manual imenso. Isso é especialmente verdadeiro quando se trata de pixel art, pois não foi encontrada nenhuma base de dados pública que contenha imagens desse tipo já segmentada. Portanto, seria necessário criar uma base de imagens do zero, o que exigiria um grande esforço e tempo.

Diante disso, as etapas para realizar a segmentação de imagem podem ser resumidas da seguinte forma: em primeiro lugar, é necessário preparar as bases de imagens, o que pode envolver a criação de uma base própria como o caso deste trabalho, com 50 pixel artes de diferentes resoluções e cenas. Em seguida, é importante estudar as técnicas utilizadas pelo

¹ informações obtidas através do link: <https://ai.meta.com/datasets/segment-anything/>

Segment Anything e outros modelos de segmentação, para entender como eles funcionam e como podem ser adaptados para a tarefa específica.

Após isso, é necessário implementar o código para testar o modelo de segmentação. Isso deve envolver a utilização de bibliotecas de aprendizado de máquina. Por fim, é importante analisar os resultados obtidos, para avaliar a eficácia do modelo e identificar áreas para melhoria.

Em resumo, a segmentação de imagem é um processo complexo que envolve várias etapas importantes, desde a preparação das bases de imagens até a análise dos resultados obtidos. A utilização de técnicas de Zero-Shot e modelos como o Segment Anything deve facilitar o processo, mas ainda é necessário um grande esforço e tempo para criar uma base de imagens próprias para a comparação com os resultados gerados pela inteligência artificial.

3.1 Tecnologias utilizadas

Nesta seção, será abordado sobre as diferentes bibliotecas estudadas para que a análise pudesse ser feita com o melhor aproveitamento da linguagem. Será mostrado também alguns tópicos referente ao estudo do python como linguagem em geral.

Para a criação da ferramenta de análise será utilizado o ambiente de desenvolvimento Visual Studio Code utilizando Python, foi optado este ambiente por sua praticidade com os diversos formatos de arquivos que serão utilizados ao decorrer do projeto. O python também servirá como única linguagem de programação para a facilidade com a criação e uso das bibliotecas de inteligência Artificial que irá adiantar muitos dos processos necessários para a criação da análise.

Para iniciar o estudo das bibliotecas antes, deve se entender melhor sobre os ambientes do python como o ambiente virtual, além de estudar como é o funcionamento das IDEs com o ambiente python dá mesma forma.

- **PyTorch:** Para o uso do *SegmentAnything*, o *PyTorch* é uma das dependências cruciais, já que ele aborda todo o processamento usado por placas de vídeo e sistemas integrados com maior facilidade. O estudo dessa biblioteca se faz necessário apenas para a instalação de seus softwares dependentes, como o uso do NVIDIA CUDA, entre outros sistemas terceiros, para o funcionamento do *Segment Anything*.
- **Scikit-image:** O *Scikit-image* será extremamente necessário para a análise dos recursos e resultados obtidos, pois é com ele que conseguimos facilmente gerar as estatísticas como o MSE, por exemplo. Não apenas isso, mas o estudo dessa ferramenta deve vir análogo ao estudo dos métodos de estimativa comparando duas imagens.

- **OpenCV:** O OpenCV (`cv2`) será empregado para lidar com tarefas relacionadas ao processamento de imagens, como leitura, conversão e redimensionamento. Ele será utilizado para ler as imagens a partir do disco (`cv2.imread`) e convertê-las entre diferentes espaços de cores para garantir a consistência de dados durante o processamento. Por exemplo, a função `cv2.cvtColor` será usada para converter as imagens do formato BGR para RGB, alinhando-as ao formato de cores esperado pelo modelo de segmentação. Além disso, o OpenCV será usado para redimensionar as imagens (`cv2.resize`) a fim de assegurar que elas possuam o mesmo tamanho antes de realizar comparações e cálculos de correlação. Um dos métodos-chave será o cálculo da Correlação Cruzada Normalizada (NCC), que será realizado por meio da função `cv2.matchTemplate`. Esse método permitirá medir a similaridade entre a imagem segmentada e a imagem esperada, fornecendo uma métrica quantitativa para avaliar a precisão do processo de segmentação.
- **Numpy:** No projeto, o `numpy` será fundamental para a manipulação eficiente de arrays multidimensionais, que representarão as imagens processadas. O `numpy` será utilizado para criar e gerenciar esses arrays, facilitando a realização de operações matemáticas e comparações pixel a pixel nas imagens. Por exemplo, a função `np.zeros` será utilizada para inicializar uma matriz de zeros que servirá como base para a imagem segmentada final na função de geração de máscara. Além disso, o `numpy` permitirá a execução de operações eficientes, como a comparação entre arrays de imagens para verificar a igualdade de pixels com a função `np.array_equal`. As imagens serão normalizadas convertendo os valores de pixel para um formato de ponto flutuante, possibilitando uma análise mais precisa ao dividir os valores por 255.0.

3.2 Procedimentos e técnicas

Para o desenvolvimento desta pesquisa, foram realizados testes preliminares explorando toda a capacidade do modelo Segment Anything (SAM), com o intuito de validar a eficácia dos processos envolvidos na análise de segmentação de imagens. Durante essa fase inicial, foi constatado que o tempo de processamento se mostrou elevado, mesmo em um sistema com hardware moderno. Como resultado, optou-se por não utilizar a tecnologia NVIDIA CUDA, uma vez que o uso intensivo de processamento pela GPU não seria imprescindível para o escopo do projeto.

Com essa decisão tomada, iniciou-se uma nova bateria de testes envolvendo diferentes tipos de imagens, abrangendo diversas categorias visuais, a fim de avaliar o potencial de segmentação do SAM em variados contextos. Os resultados foram, em sua maioria, satisfatórios, com o modelo demonstrando desempenho adequado na maioria dos casos analisados.

Posteriormente, iniciou-se o desenvolvimento de imagens específicas, criadas pelo autor em estilo pixel art, para validar a performance do modelo em situações de segmentação por cor. Essa etapa exigiu ajustes no código, especialmente no que diz respeito à manipulação das camadas e formatos das imagens, além da tipagem adequada em Python. Esse processo envolveu pesquisas extensivas e múltiplas tentativas até a obtenção de um resultado satisfatório na geração de grupos de cores a partir das imagens.

Com o sistema de segmentação de imagens em funcionamento, iniciou-se a etapa de criação manual das camadas de segmentação. O autor definiu como deveriam ser os resultados esperados da segmentação proposta pela IA, segmentando um conjunto de 50 imagens em estilo pixel art com base em cores específicas para realizar as análises. A partir desse ponto, foi possível aplicar métodos existentes para a comparação entre duas imagens.

Inicialmente, foi desenvolvido um método dentro do código em python para validar a quantidade de pixels corretos, esse algoritmo inicialmente no pixel 0x0 ele gera uma conexão entre a cor original e a cor esperada, assim ele segue até o final da imagem, se caso ele encontrar uma cor diferente no original ele gera um erro, se não ele prossegue assim ele pode calcular com uma boa margem, já que quando se fala de pixel arte, onde a amplitude entre as cores é maior.

3.2.1 Base de dados

A seleção de uma base de imagens adequada é um passo fundamental para a realização de experimentos em visão computacional, especialmente em estudos que envolvem segmentação de imagens. Neste trabalho, foi utilizada uma base de imagens composta por amostras de média e baixa resolução, com o objetivo de avaliar a eficácia do modelo Segment Anything (SAM) em diferentes contextos de qualidade visual.

As imagens foram escolhidas de modo a abranger uma variedade de cenários, incluindo personagens, cenários, estruturas e objetos, a fim de garantir a representatividade e a diversidade dos dados. Além disso, as resoluções das imagens variaram entre 32x32 *pixels* e 176x256, permitindo uma análise comparativa detalhada do desempenho do SAM em diferentes cenários de pixel arte.

A base foi composta por imagens próprias e conjuntos de imagens gratuitas da plataforma Itch.io², para assegurar a consistência nos experimentos, todas as imagens passaram por um pré-processamento, que incluiu segmentar manualmente todas as imagens. Esse processo foi essencial para padronizar os dados e minimizar a interferência de variáveis externas que pudessem comprometer a análise dos resultados.

A seguir será apresentado algumas das imagens da base para uma melhor definição dos cenários obtidos na análise como na Figura 4.

² Acesso através do link: <https://itch.io/>

Figura 4 – Imagem de exemplo original



Fonte: <https://itch.io/>

E após a segmentação manual, temos a seguinte imagem, lembrando que alguns padrões de cores foram utilizados para que fosse mais fácil identificar e comparar com os dados vindos do (SAM).

Figura 5 – Imagem de exemplo pós segmentação manual



Fonte: Elaborado pelo autor

Já na Figura 5, podemos ver cores bem fortes mostrando qual deveria ser a máscara realizada para a comparação futura.

3.3 Métricas Utilizadas

Para avaliar a qualidade dos resultados obtidos, foram utilizadas duas métricas amplamente conhecidas na área de visão computacional: o *Erro Médio Quadrático* (*Mean Squared Error - MSE*) e a *Correlação Cruzada Normalizada* (*Normalized Cross-Correlation - NCC*). Abaixo, são descritas as fórmulas e os algoritmos utilizados para o cálculo dessas métricas.

- **Erro Médio Quadrático**

O MSE mede a diferença média ao quadrado entre os valores previstos e os valores reais, sendo uma métrica que avalia o erro absoluto entre a segmentação obtida e a referência. Quanto menor o valor do MSE, mais próximo o resultado está do esperado. A fórmula do MSE é dada por:

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

Onde:

- x_i : Valor previsto (ou resultado da segmentação).
- y_i : Valor real (ou referência).
- N : Número total de pixels na imagem.

- **Correlação Cruzada Normalizada**

A Correlação Cruzada Normalizada (NCC) é uma métrica amplamente utilizada no processamento de imagens para medir a similaridade estrutural entre duas imagens. Diferentemente de outras métricas, como o Erro Médio Quadrático (MSE), a NCC considera a relação de intensidade entre os pixels de ambas as imagens, desconsiderando variações de escala ou deslocamento que possam ocorrer. O valor da NCC é normalizado, variando entre -1 e 1. Um valor de NCC igual a 1 indica que as imagens são estruturalmente idênticas, ou seja, há uma correspondência perfeita entre seus pixels. Por outro lado, um valor de NCC igual a -1 reflete uma relação totalmente inversa entre as imagens, em que uma é o negativo estrutural da outra. Quando a NCC está próxima de 0, isso sugere uma baixa correlação, indicando pouca ou nenhuma semelhança estrutural entre as imagens.

A fórmula para calcular a NCC utiliza a média e os valores de intensidade dos pixels das duas imagens, considerando a soma ponderada das diferenças entre os valores individuais de cada pixel e suas respectivas médias. Por ser uma métrica normalizada, a NCC é especialmente robusta contra alterações de brilho ou contraste, concentrando-se apenas na estrutura relativa das imagens. Essa característica faz com que a NCC seja amplamente empregada em tarefas como registro de imagens, rastreamento de objetos e detecção de padrões, onde a identificação precisa de semelhanças estruturais é essencial. Sua fórmula é dada por:

$$NCC = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2 \cdot \sum_{i=1}^N (y_i - \bar{y})^2}}$$

Onde:

- x_i : Valor de intensidade do pixel na imagem segmentada.
- y_i : Valor de intensidade do pixel na imagem de referência.
- \bar{x} : Média dos valores dos pixels na imagem segmentada.
- \bar{y} : Média dos valores dos pixels na imagem de referência.

4 ANÁLISE DOS RESULTADOS

Neste capítulo, são apresentados e discutidos os resultados obtidos na aplicação dos métodos de segmentação de imagens propostos neste trabalho, com ênfase nas imagens de média e baixa resolução. Utilizando o modelo Segment Anything (SAM), avaliou-se a eficácia do método utilizado, que utilize inteligência artificial para aprimorar a precisão da segmentação.

Para uma análise quantitativa robusta, foram adotadas as métricas Mean Squared Error (MSE) e Normalized Cross-Correlation (NCC), além de um método específico criado para complementar a avaliação dos segmentos. Estes indicadores foram aplicados para avaliar a qualidade das segmentações produzidas, comparando-as com padrões de segmentação existentes. Este capítulo tem, portanto, como objetivo apresentar as análises qualitativas e quantitativas dos resultados, destacando os avanços e as limitações do método proposto em relação aos métodos tradicionais.

4.1 Desenvolvimento do sistema

Nesta seção será apresentado todos os processos em python realizados para a criação do código de análise dos dados criados e os já existentes realizados pelo autor, o desenvolvimento foi iniciado com a implementação das mascaras geradas pelo segment anything como mostra a Figura 6.


Figura 6 – Código utilizando o SAM para retornar a segmentação



Nesse processo a função faz o acesso do modelo que no caso utilizado foi o model vit h pesando 2.38 Gigabytes de memória, para essa função é possível a utilização de dois caminhos de processamento da imagem atual, o primeiro deles é utilizando a GPU de uma placa de vídeo o que em tese poderia diminuir o gargalo gerado nas CPUs e melhorar a velocidade em que é executado, logo após a geração das mascaras pelo SAM, o próximo passo é justamente iniciar a criação das mascaras manualmente.

Para essa etapa foi utilizado várias pixel artes retiradas do site: itch.io que tem grande parte da comunidade servindo *assets* gratuitos para projetos pessoais ou jogos *indie*, a execução dito foi realizada com uma visão humana para servir de referencia para os resultados gerados da inteligencia artificial, mas para as respostas do SAM não temos a geração de uma imagem em si mas de um *array* de camadas criadas com cada segmento gerado, para isso se tornar uma imagem é necessário este código mostrado na Figura 7.

Figura 7 – Código para o método próprio de análise



```
expected_image = cv2.imread('Assets/Expected/' + image_path)

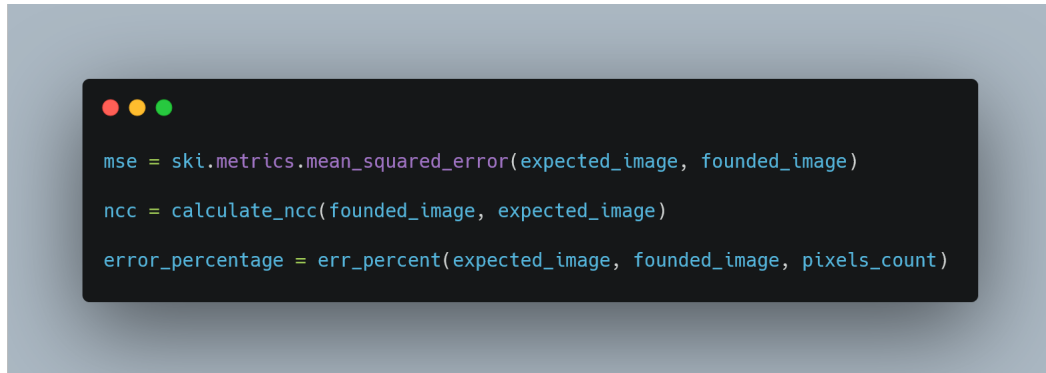
#Passando as duas imagens para um mesmo padrão de cores
founded_image = cv2.cvtColor(founded_image, cv2.COLOR_BGR2RGB)
expected_image = cv2.cvtColor(expected_image, cv2.COLOR_BGR2RGB)

# Normalizando as imagens
founded_image = founded_image.astype(np.float32) / 255.0
expected_image = expected_image.astype(np.float32) / 255.0
pixels_count = founded_image.shape[0] * founded_image.shape[1]
founded_image = generate_mask(masks)
```

Após os dois tipos de imagem já estarem preparados para a comparação é necessário a utilização de mais de um método de análise para compararmos e abrangermos mais pontos em comum sobre o resultado esperado, por isso é utilizado neste momento três tipos de análises, sendo duas delas configuradas pela biblioteca Scikit, a outra foi realizada pelo autor com a tentativa de gerar resultados diferentes dos métodos padrões.

O processo foi desenvolvido analisando a imagem pixel a pixel, começando no canto superior esquerdo, o primeiro pixel (coordenada zero). A varredura percorre cada linha da esquerda para a direita e, ao final de cada linha, passa para a próxima. Cada pixel possui duas referências: uma para a imagem original, segmentada pelo autor, e outra para a segmentação feita pela inteligência artificial (IA). Para cada pixel na imagem original, cria-se uma conexão com o pixel correspondente na imagem gerada pela IA, e o processo continua até que uma diferença de cor seja detectada entre os pixels correspondentes das duas imagens. Quando isso ocorre, o código verifica qual imagem apresentou a diferença primeiro; se a diferença for detectada primeiro na imagem original, o código marca essa mudança no grupo segmentado criando assim o início de uma nova conexão entre os pixels, mas, se a diferença aparecer apenas na imagem da IA, o pixel é marcado como incorreto, indicando um possível erro da IA, pois a imagem original ainda mantém a mesma cor. Ao final, a análise calcula a porcentagem de erros comparando o número de pixels incorretos com o total de pixels da imagem, todo este processo é feito nessa função do código-fonte como mostra a Figura 8.

Figura 8 – Código para receber os valores de cada tipo de análise



```
mse = ski.metrics.mean_squared_error(expected_image, founded_image)
ncc = calculate_ncc(founded_image, expected_image)
error_percentage = err_percent(expected_image, founded_image, pixels_count)
```

Além da análise própria, também foram utilizados o método NCC e o MSE, que serão abordados a seguir. O primeiro deles, o NCC (Normalized Cross Correlation), é uma técnica usada para medir a similaridade entre imagens. Nela, utiliza-se geralmente uma imagem de referência e uma imagem-alvo; o índice de correlação varia de -1 a 1, onde valores mais próximos de 1 indicam alta similaridade, valores próximos de zero indicam baixa ou nenhuma correlação com a imagem de referência, e o valor -1 representa uma correlação inversa com a imagem. No entanto, o NCC não é vantajoso para casos em que há imagens complexas para analisar, o que é o oposto das imagens-alvo deste estudo.

O segundo método, o MSE (*Mean Squared Error*), é uma técnica utilizada para avaliar a diferença entre duas imagens ao medir o erro médio entre os pixels correspondentes. O MSE calcula a média dos quadrados das diferenças de intensidade entre os pixels das duas imagens, gerando um valor que representa o grau de discrepância entre elas. Quanto maior o valor do MSE, maior a diferença entre as imagens comparadas; valores próximos de zero indicam alta similaridade. Esse método é simples e eficaz para medir discrepâncias, mas pode ser sensível a pequenas variações de pixel, sendo ideal para imagens onde os detalhes e as variações sutis são relevantes.

A combinação dessas três análises permite uma avaliação mais completa e detalhada da similaridade e discrepância entre as imagens, onde cada método contribui com uma perspectiva distinta. O cruzamento dos resultados oferece um panorama mais robusto, permitindo identificar falhas localizadas, medir a correlação global e quantificar as diferenças de intensidade de forma precisa. Essa integração facilita a obtenção de dados mais confiáveis e detalhados sobre o desempenho da segmentação, ampliando a compreensão sobre as sutilezas entre as imagens e fortalecendo a precisão das conclusões finais.

4.2 Agilidade e Eficiência

Nessa seção será abordado o fator velocidade para o processo ser finalizado, ou seja, independente dos resultados qual a margem de tempo destinada apenas a realização

da segmentação das imagens, lembrando que isso seria apenas um fragmento de todo o desenvolvimento da ferramenta.

Como o esperado, foi identificado nos computadores usados para executar o código uma quantidade de tempo razoável até sua finalização, apesar disso, entre eles houveram mínimas diferenças de tempo de execução, mesmo com abruptas diferenças de processamento, em seguida será mostrado as propriedades de cada um deles, para uma melhor comparação da média dos resultados obtidos em termos de desempenho, como mostra a Tabela 1.

Tabela 1 – Comparação de Desempenho entre Computador e Notebook

Propriedade	Computador	Notebook
Processador	AMD Ryzen 5 1400 Quad-Core	Intel Core i5-1235U
Número de Threads	8	12
Frequência Base	99.8 MHz	99.8 MHz
Frequência Máxima	3192.4 MHz	3790.7 MHz
Memória RAM	16 GB DDR4	8 GB DDR4
Frequência da Memória	Não especificada	1596.1 MHz
Chipset	AMD B350	Intel Alder Lake rev. 04
Placa Mãe	Asus Prime A320M-K/BR	Modelo LNVNB161216
¹	4 GB NVIDIA 1050 TI	Intel UHD Graphics

Com a tabela é possível notar que apesar da comparação semelhante quando nos referimos ao processamento das máquinas, é extremamente necessário levar em consideração o uso da placa gráfica para o processamento com o suporte do CUDA, que nada mais é do que um sistema criado pela NVIDIA a fim de utilizar as placas gráficas como potencializadores para integração de inteligências artificiais, contudo, os resultados em termos de velocidade de execução foram muito próximos como é possível notar na Tabela 1, com esse resultado é perceptível que para utilizar do grande potencial do suporte do CUDA para rodar as aplicações utilizando o processamento da placa de vídeo, esse valor só ultrapassa o processador se caso a placa de vídeo já for mais recente.

4.3 Análise de resultados

A análise mostrou que a porcentagem de acerto variou de 45,17% a 97,90%, indicando maior sensibilidade do modelo a imagens com mais pixels, que tiveram acertos médios superiores. Imagens de 1024 pixels apresentaram menor variância no MSE (0,0465) e desvio padrão de 0,2157, evidenciando maior robustez em resoluções intermediárias.

Os grupos apresentados se referem ao grupo de imagens que será apresentado na Tabela 2, apenas algumas das 50 imagens foram usadas na tabela:

Tabela 2 – Tabela de dados

ID	Dimensão	MSE	NCC	ID	Dimensão	MSE	NCC
1	32x32	0.423	0.204	26	18x20	0.483	0.698
2	32x32	0.414	0.695	27	13x13	0.468	0.573
3	32x32	0.488	0.533	28	176x256	0.323	0.804
4	32x32	0.410	0.607	29	15x13	0.498	0.659
5	32x32	0.329	0.623	30	124x134	0.261	0.872
6	32x32	0.327	0.709	31	125x109	0.250	0.702
7	32x32	0.368	0.553	32	12x16	0.469	0.572
8	32x32	0.394	0.709	33	21x27	0.429	0.687
9	32x32	0.391	0.686	34	30x32	0.502	0.511
10	32x32	0.390	0.715	35	42x46	0.375	0.544
11	32x32	0.417	0.666	36	96x112	0.340	0.410
12	32x32	0.355	0.647	37	29x33	0.349	0.588
13	32x32	0.388	0.514	38	112x80	0.363	0.581
14	32x32	0.363	0.684	39	7x9	0.354	0.826
15	32x32	0.389	0.706	40	18x19	0.488	0.539
16	32x32	0.360	0.532	41	54x42	0.448	0.445
17	32x32	0.566	0.542	42	81x91	0.206	0.664
18	32x32	0.390	0.518	43	80x80	0.378	0.676
19	32x32	0.356	0.602	44	9x16	0.535	0.689
20	32x32	0.402	0.720	45	55x27	0.528	0.755
21	32x32	0.551	0.387	46	55x27	0.369	0.647
22	32x32	0.375	0.634	47	11x13	0.468	0.334
23	32x32	0.357	0.661	48	176x256	0.488	0.602
24	32x32	0.384	0.656	49	16x128	0.382	0.617
25	112x80	0.370	0.567	50	32x224	0.380	0.498

Nos resultados apresentados a seguir, destacamos algumas imagens representativas analisadas. Essas imagens foram selecionadas para ilustrar visualmente os padrões associados aos valores obtidos em cada grupo. As figuras mostram as diferenças e semelhanças nas características dos segmentos, oferecendo uma compreensão mais clara dos resultados numéricos. Cada grupo é representado por uma seleção de imagens específicas que refletem os dados discutidos, permitindo uma análise complementar e qualitativa dos resultados como mostra nas Figuras 9, 10, 11 e 12.

Figura 9 – A imagem com o melhor resultado (NCC)



Figura 10 – A imagem com o pior resultado (NCC)

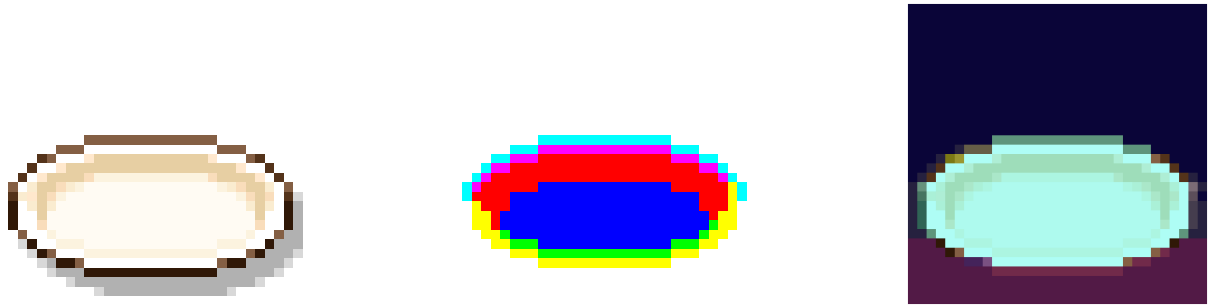
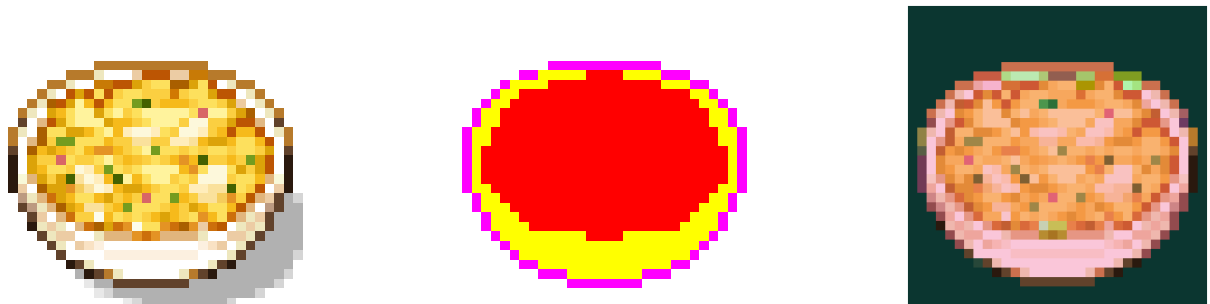


Figura 11 – A imagem com o melhor resultado (MSE)



Figura 12 – A imagem com o pior resultado (MSE)



Durante a análise dos resultados utilizando as métricas MSE e NCC, foi possível identificar algumas diferenças marcantes em seus comportamentos.

Considerando que as métricas assimilam como melhor resultado as figuras com menor taxa de erro e pior resultado as figuras que apresentam mais divergências da imagem original, percebe-se que no caso do MSE, os resultados não foram compatíveis com o esperado.

Na Figura 11, identificada como melhor resultado, não foram capturadas erros marcantes e aspectos importantes, como bordas e texturas, os quais são facilmente visualizados.

O mesmo acontece no pior resultado, pois a figura é próximo ao objeto original, mas a análise não conseguiu identificar tais similaridades, reforçando as limitações dessa métrica para análises perceptivas e estruturais seguindo o contexto das mascaras realizadas pelo (SAM).

Isso ocorre porque o MSE avalia diferenças absolutas entre os pixels, ignorando o

contexto visual da imagem.

Por outro lado, com o NCC, o cenário foi um pouco diferente. No melhor resultado, o NCC pareceu corresponder melhor à qualidade perceptiva da imagem, já que identifica a similaridade estrutural entre as imagens. Isso ficou evidente na imagem que apresenta altos níveis de segmentação, com contornos bem definidos e características mais preservadas. Comparado ao MSE, o NCC mostrou-se mais confiável para avaliar a relação entre as estruturas das imagens, especialmente em cenários onde a segmentação é um fator crítico.

De forma geral, essa análise reforça que o MSE, nem sempre reflete a qualidade perceptiva ou funcional dos resultados. Métricas como o NCC, que consideram a similaridade estrutural, podem ser mais adequadas para avaliar imagens segmentadas. Para melhorar a análise, que avalia similaridades perceptuais, e explorar ajustes no método utilizado para gerar as imagens, principalmente em cenários onde o MSE apresentou maior discrepância.

5 CONCLUSÕES

O presente trabalho explorou métodos de segmentação de imagens com foco na aplicação do modelo Segment Anything (SAM) em imagens de média e baixa resolução. Foram abordadas diversas técnicas de segmentação, incluindo métodos baseados em camadas e blocos, e foi realizado um estudo aprofundado das ferramentas e bibliotecas em Python, como PyTorch, Scikit-image e OpenCV, que sustentaram o desenvolvimento dos experimentos. A partir das análises realizadas, constatou-se que o SAM oferece bons resultados em imagens de média resolução, com valores de *Mean Squared Error* (MSE) e *Normalized Cross-Correlation* (NCC) indicativos de precisão e consistência.

No entanto, os resultados em baixa resolução apresentaram maior instabilidade, demonstrando a limitação da IA em reconhecer bordas e detalhes em imagens de menor qualidade. A análise dos tempos de execução também apontou que a implementação do SAM em computadores com suporte a GPU (CUDA) contribui para a eficiência, mas mesmo com hardware moderno, a variabilidade entre tempos sugere limitações na escalabilidade do método para aplicações em tempo real ou em larga escala.

Este estudo contribui para a área de visão computacional ao validar a eficácia do SAM em contextos específicos de resolução, além de propor uma metodologia para análise de segmentação. A aplicação prática desses métodos em sistemas de segmentação de baixa resolução requer melhorias para garantir maior estabilidade e precisão, indicando possíveis caminhos para trabalhos futuros que explorem alternativas híbridas entre SAM e outras técnicas, visando aumentar a eficácia da IA em diversos cenários de imagem. Em síntese, o trabalho revelou a viabilidade e as limitações do SAM, contribuindo para a construção de uma base teórica e prática sobre a aplicabilidade de modelos de segmentação em imagens de baixa qualidade.

Com base nos resultados obtidos e nas limitações identificadas ao longo de todo o atual estudo, algumas direções para trabalhos futuros podem ser apontadas. Uma delas é investigar técnicas de *depth estimation*. Outra linha promissora é a investigação da identificação de profundidade com base nos grupos de segmentação obtidos. Essa abordagem permitiria a extrapolação de informações tridimensionais a partir de dados bidimensionais, contribuindo para aplicações que demandam análise de cenas em profundidade, como em realidade aumentada, mapeamento dentre outros. A combinação dos grupos segmentados com técnicas de reconstrução ou estimativa de profundidade poderia revelar novos padrões e ampliar a aplicabilidade para médias e baixas resoluções o que pode oferecer a artistas uma maior eficiência quando se trata de elevar o design, ainda mais quando se trata de pixel artes.

REFERÊNCIAS

- CAO, W. et al. A review on multimodal zero-shot learning. **WIREs Data Mining and Knowledge Discovery**, v. 13, n. 2, p. e1488, 2023. Disponível em: <<https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1488>>. 13
- CARVALHO, A. C. P. d. L. F. d. Inteligência artificial: riscos, benefícios e uso responsável. **Estudos Avançados**, Instituto de Estudos Avançados da Universidade de São Paulo, v. 35, n. 101, p. 21–36, Jan 2021. ISSN 0103-4014. Disponível em: <<https://doi.org/10.1590/s0103-4014.2021.35101.003>>. 9
- GONZALEZ, C. I.; MELIN, P.; CASTILLO, O. Edge detection method based on general type-2 fuzzy logic applied to color images. **Information**, v. 8, n. 3, 2017. ISSN 2078-2489. Disponível em: <<https://www.mdpi.com/2078-2489/8/3/104>>. 9
- KE, L. et al. **Segment Anything in High Quality**. 2023. Disponível em: <<https://arxiv.org/abs/2306.01567>>. 10
- KIRILLOV, A. et al. Segment anything. **arXiv:2304.02643**, 2023. 9
- LIU, Y. et al. **SANeRF-HQ: Segment Anything for NeRF in High Quality**. 2024. Disponível em: <<https://arxiv.org/abs/2312.01531>>. 11
- MAYANGKY, N. et al. Analyzing the comparative methods of prewitt, robinson, krishch and roberts in detecting the edges of rice leaves. **Jurnal Techno Nusa Mandiri**, v. 21, n. 1, p. 37 – 43, Mar. 2024. Disponível em: <<https://ejournal.nusamandiri.ac.id/index.php/techno/article/view/5509>>. 9
- MINAEE, S. et al. **Image Segmentation Using Deep Learning: A Survey**. 2020. Disponível em: <<https://arxiv.org/abs/2001.05566>>. 11
- TANEJA, A.; RANJAN, P.; UJJLAYAN, A. A performance study of image segmentation techniques. In: **2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)**. [S.l.: s.n.], 2015. p. 1–6. 7
- YANG, Y. et al. Layered object models for image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 9, p. 1731–1743, 2012. 8
- YU, Y. et al. Techniques and challenges of image segmentation: A review. **Electronics**, v. 12, n. 5, 2023. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/12/5/1199>>. 7
- ZAITOUN, N. M.; AQEL, M. J. Survey on image segmentation techniques. **Procedia Computer Science**, v. 65, p. 797–806, 2015. ISSN 1877-0509. International Conference on Communications, management, and Information technology (ICCMIT'2015). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050915028574>>. 7, 8
- ZHAO, C. et al. Monocular depth estimation based on deep learning: An overview. **Science China Technological Sciences**, Springer, v. 63, n. 9, p. 1612–1627, 2020. 12