

VITOR HUGO FAZOLI DA SILVA

ANÁLISE E DESENVOLVIMENTO DE MÉTODOS DE SEGMENTAÇÃO DE IMAGENS DE MÉDIA E BAIXA RESOLUÇÃO COM O USO DO MODELO SEGMENT ANYTHING (SAM)

Orientador: Prof Dr.Gabriel
Marcelino Alves

SÃO JOÃO DA BOA VISTA

2023

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	Motivação e Contexto	10
2.3.2	Arquitetura do SAM	10
2.3.3	Conjunto de Dados SA-1B	10
2.3.4	Aplicações e Resultados	11
2.3.5	Comparações com Métodos Tradicionais	11
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	Levantamento de Requisitos	14
3.2	Estudo: Python e suas bibliotecas	14
3.2.1	PyTorch	15
3.2.2	Scikit-image	15
3.2.3	OpenCV	15
3.2.4	Numpy	15
3.3	Estudo: Inteligência Artificial	16
3.3.1	Segmentação por mascara	16
3.3.2	Segmentação por pontos de clique	16
3.3.3	Segmentação por caixa de delimitação	16
3.3.4	Segmentação por texto	16
3.3.5	Segmentação interativa	17
3.3.6	Segmentação automática	17
3.4	Procedimentos e Técnicas	17
3.5	Avaliar vantagens e limitações das ferramentas abordadas	18
3.6	Modelagem da ferramenta	19
3.7	Desenvolvimento do sistema	20
3.8	Justificativa das Tecnologias a serem adotadas	22
4	ANÁLISE DOS RESULTADOS	24
4.1	Agilidade e Eficiência	24
4.2	Resultados/Impactos	25
4.3	Cronograma do Trabalho	25
5	CONCLUSÕES PARCIAIS	27

REFERÊNCIAS 28

RESUMO

A iluminação é de grande importância para pinturas e esboços digitais nos dias de hoje, pois fornece a percepção de volume ao ambiente. Porém, a iluminação ainda é difícil de replicar em uma imagem digital, pois representa algo muito diferente de um ambiente tridimensional, ainda mais se usarmos uma imagem em baixa resolução. Nesse sentido, O objetivo do trabalho é simular a iluminação em imagens digitais de média e baixa resolução por meio do desenvolvimento de uma ferramenta. Para se alcançar o objetivo, pretende-se que a ferramenta considere informações de luz em um espaço tridimensional para simular a iluminação na imagem bidimensional. Portanto o que se espera deste trabalho é que imagens de média e baixa resolução apresentem boa iluminação.

Palavras-chave: Iluminação. Algoritmo. Ambiente. Bidimensional.

1 INTRODUÇÃO

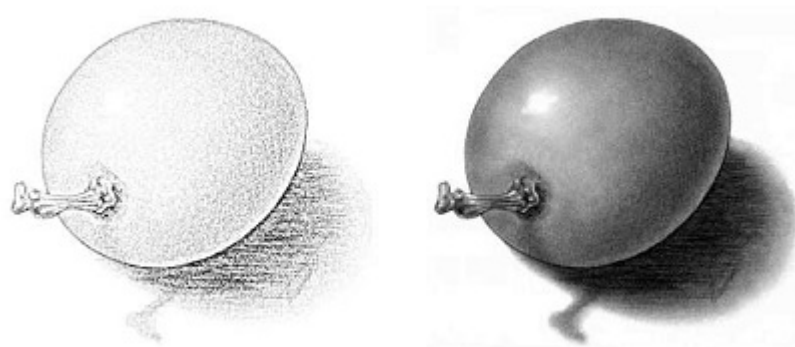
Desde que as pinturas começaram, se formaram vários estilos de arte, ainda que no princípio tudo era voltado a chegar ao realismo das pinturas, por não existir nada que pudesse retratar rostos e momentos melhor do que as artes.

No entanto, isso mudou quando surgiram as câmeras, que poderiam captar tudo de forma praticamente instantânea, com isso era o momento de se pensar que as pinturas iriam sumir, mas ela se renovou e ao invés de buscar o realismo, agora ela estava em busca de trazer novas sensações, como as pinturas abstratas, estilos como cubismo ou surrealismo que usavam de várias curvas e cores para demonstrar expressões (Cezimbra, 2021).

Esses estilos artísticos foram uma revolução no mundo da arte, como o surrealismo por exemplo, que tratava de possibilidades infinitas em suas obras, porque ali tudo era possível e o único limite era imaginação, Nesse período foram apresentadas várias obras que ao olhar não faziam sentido algum, porém era esse o princípio da ideia, pois elas eram criadas para passar uma sensação a quem olhasse, como a de liberdade, entre outros.

Atualmente a iluminação é de extrema importância para as pinturas e desenhos digitais pois é ela quem traz o volume ao ambiente, como pode ser observado na Figura 1.

Figura 1 – Duas imagens demonstrando a diferença da iluminação



(Grafitti, 2019)

Como podemos observar que na figura acima, a imagem da esquerda apresenta apenas um rascunho, sem uso das técnicas de iluminação a portanto a noção de volume e profundidade não fica muito claro, por outro lado a imagem da direita sendo algo mais agradável nas pinturas como é possível notar, além do mais várias composições de cores da luz retratada em uma imagem pode trazer vislumbres fantásticos.

Com a chegada dos primeiros modelos em três dimensões, como cubos, pirâmides e esferas entre outros, foi possível enxergar através de uma tela, um objeto qualquer que

fosse e isso mudou tudo, hoje podemos ver vários exemplos de modelos que está em nossa realidade, como a impressora 3D ou o *Billboard*¹, pois é com ela que foi possível ver a grande beleza dos modelos, é por esse motivo pelo qual atualmente a iluminação é essência para nos causar a sensação de volume, sem pensar nos algoritmos mais robustos como *ray tracing* que segundo Shirley e Morley (2008) retrata, é um algoritmo onde através de uma janela, os raios são direcionados para as imagens. As superfícies são perdidas ou atingidas por cada pixel, que é representado por um raio. Ao atingir uma superfície, o raio se refrata e continua em um novo curso, causando a formação de luzes adicionais nos arredores.

Porém para os artistas chegarem perto dos algoritmos de iluminação que criam visuais impecáveis em três dimensões, as pinturas e artes foram obrigadas a criar um volume e melhorar a iluminação, pois como a luz funcionava de forma sistemática em três dimensões, as artes puderam abrir espaço em uma variação de cores e paletas diferenciadas que trazem um dinamismo maior.

Por isso atualmente a iluminação se tornou algo tão importante para o mundo, que não importa mas o estilo que é usado, é uma parte essencial. É nesse momento em que chegamos ao cerne do problema, pois com essa importância que a iluminação tem sobre a arte e os avanços tecnológicos cada vez mais ligados a ter um *design* rápido e com eficiência, ferramentas que criam a luz em pinturas, desenhos, objetos, *sprites* ou até logos poderia impactar a arte de maneira eficiente e até mesmo para aumentar a comunidade, fazendo com que várias pessoas iniciantes que não conseguem bons resultados possam aparecer no mundo do *design*.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo do trabalho é simular a iluminação em imagens digitais de média e baixa resolução por meio do desenvolvimento de uma ferramenta.

1.1.2 Objetivos Específicos

- Encontrar algoritmos que têm o propósito de iluminar cenas bidimensionais;
- Criar um protótipo e desenvolver a ferramenta para iluminação;
- Realizar testes na ferramenta Relight que será analisado para comparação dos resultados.

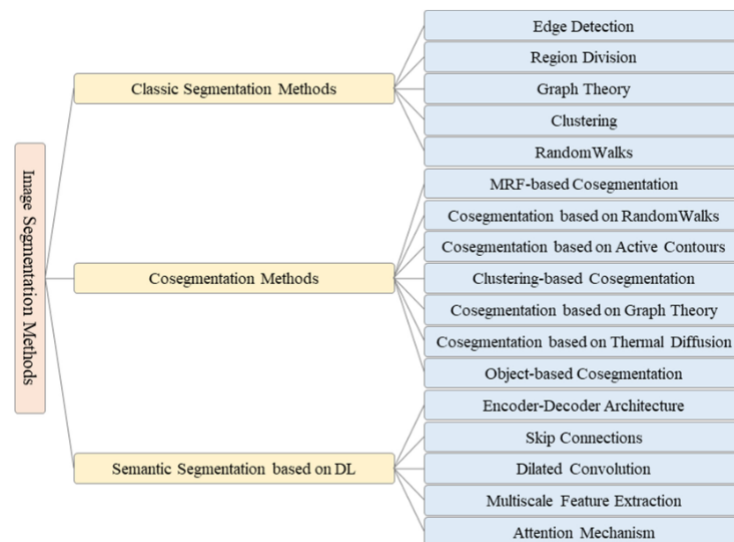
¹ apresentado em Tóquio, uma televisão imensa que pode captar imagens como se fossem 3D, criando um aspecto de profundidade pela sua curvatura

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 Aql (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 2 mostra esses diferentes métodos de segmentação, organizando-os em categorias principais e destacando os principais algoritmos em cada abordagem.

Figura 2 – Diagrama criado por Yu et al. (2023)



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, conside-

rando 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.

2.3.1 Motivação e Contexto

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).

2.3.2 Arquitetura 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.

2.3.3 Conjunto de Dados SA-1B

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.

2.3.4 Aplicações e Resultados

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).

2.3.5 Comparações com Métodos Tradicionais

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

Nesta seção são apresentados os trabalhos correlatos ao proposto neste trabalho

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.

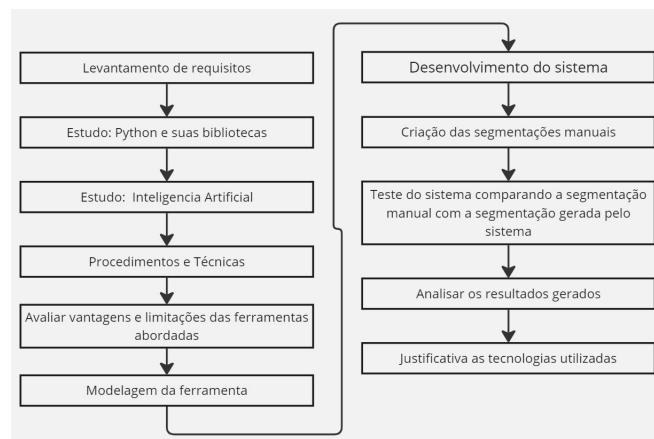
2.4.2 Monocular depth estimation based on deep learning: An overview

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

Descrever metodologia, materiais e métodos utilizados no estudo, bem como os procedimentos experimentais realizados, nesta etapa será descrito vários assuntos sobre os passos a serem realizados iniciando com o levantamento dos dados, para uma análise e estudo posteriormente analisando todos os processos que são realizados, afim de iniciar o desenvolvimento.

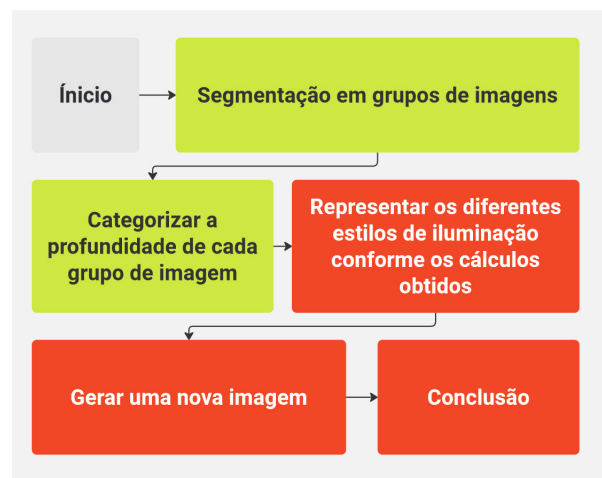
Figura 3 – Passos feito na plataforma miro



Fonte: Feito pelo autor

Serão discutidos os principais passos estabelecidos para a execução completa do projeto idealizado, conforme ilustrado na Figura 4.

Figura 4 – Passos feito na plataforma miro



Fonte: Feito pelo autor

Como mostrado o diagrama foi elaborado utiliza duas cores distintas para representar o estado de cada fase do processo: verde, indicando as etapas em andamento; e

vermelho, para aquelas ainda não iniciadas. As fases não iniciadas foram postergadas devido a dificuldades técnicas encontradas ao longo do desenvolvimento, as quais exigem uma análise mais aprofundada e a reformulação de estratégias para assegurar a eficiência e adequação da solução proposta.

3.1 Levantamento de Requisitos

Nessa sessão será abordado a pesquisa qualitativa feita a partir da ferramenta *Relight* que usa de um algoritmo para identificação de camadas em uma imagem, assim, quando a imagem é colocada na ferramenta ela faz um mapeamento das áreas altas e baixas para criar as camadas, por esse motivo as camadas que são denominadas como mais alta recebe mais luz do que as mais baixas.

Porém a ferramenta tem uma funcionalidade a mais onde é possível definir em que nível está a luz, fazendo com que a iluminação do objeto possa vir das camadas mais a baixo para as camadas acima, está ferramenta com a luz funciona como um pincel de adicionar em softwares de pintura, pois ela recebe a luz já existente na foto e complementa por cima com a luz própria, podendo ser de qualquer cor.

A seguir será mostrado na tabela, onde terá a avaliação qualitativa referente a vários tipos de imagens testadas a partir da ferramenta, que para efeitos deste trabalho serão utilizados os valores de alta resolução como acima de 720 pixels, média resolução entre 720 e 256 pixels e baixa resolução abaixo de 256 pixels, retratado na tabela 1.

Imagem	Alta Resolução	Média Resolução	Baixa Resolução
Objetos	Funciona	Funciona	Restrição
Humanos	Funciona	Funciona	Restrição
Paisagens	Funciona	Problema	Problema

Tabela 1 – Avaliação qualitativa da ferramenta

Nessa tabela, foi adotado três categorias para descrever o funcionamento: Funciona, Restrição e Problema. A primeira categoria é utilizada para descrever que a imagem utiliza se comportou de forma correta, Restrição é a categorização para as imagens que funcionam mas em alguns casos ou áreas da imagem geram problemas, já a categoria Problema indica os casos onde a luz não reconhecia as formas de maneira correta.

3.2 Estudo: Python e suas bibliotecas

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 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.

3.2.1 PyTorch

Para o uso da SegmentAnything o pyTorch é uma das dependências cruciais, já que ele aborda todo o processamento usado por placas de video 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.

3.2.2 Scikit-image

O Scikit 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 análoga ao estudo dos métodos de estimativa comparando duas imagens.

3.2.3 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.

3.2.4 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.

3.3 Estudo: Inteligência Artificial

Nesse estudo será demonstrado algumas das partes que serão necessárias para a criação da análise posteriormente, inicialmente foi preciso uma serie de estudos sobre todas as possibilidades relacionadas as possíveis segmentações realizadas, a seguir cada tipo de segmentação será melhor explicada.

3.3.1 Segmentação por mascara

A segmentação por máscara utiliza um modelo para identificar e isolar regiões específicas dentro de uma imagem. O modelo gera uma máscara binária onde a região segmentada é representada por 1 e o fundo por 0. Este método é eficaz para separar objetos ou áreas de interesse em imagens, permitindo uma análise mais detalhada dessas regiões específicas.

3.3.2 Segmentação por pontos de clique

Neste método, o usuário seleciona pontos dentro da área que deseja segmentar. É utilizado esses pontos como referências para definir os limites da região de interesse. A segmentação é então ajustada com base na localização desses pontos, oferecendo um controle mais preciso sobre as regiões segmentadas, especialmente em imagens complexas.

3.3.3 Segmentação por caixa de delimitação

Aqui, o usuário fornece uma caixa delimitadora ao redor do objeto ou área de interesse. O modelo então realiza a segmentação dentro dessa caixa. Esse método é útil para rapidamente identificar e segmentar objetos que podem ser facilmente contidos dentro de uma área retangular, facilitando a definição da região de interesse.

3.3.4 Segmentação por texto

A segmentação por texto permite ao usuário descrever a região ou objeto desejado em termos textuais. O modelo utiliza essa descrição para identificar e segmentar a área correspondente na imagem. Este método é útil quando a descrição do objeto é mais clara

em palavras do que em detalhes visuais, oferecendo uma maneira eficiente de segmentar com base em descrições contextuais.

3.3.5 Segmentação interativa

Este método combina várias abordagens interativas, como pontos de clique e caixas de delimitação, permitindo que o usuário refine a segmentação com base no feedback contínuo. A segmentação inicial pode ser ajustada e melhorada conforme o usuário fornece mais informações ou faz ajustes na área segmentada, resultando em uma segmentação mais precisa e adaptada às necessidades específicas.

3.3.6 Segmentação automática

Na segmentação automática, é aplicado algoritmos de segmentação sem intervenção do usuário. O modelo usa padrões aprendidos para identificar e segmentar automaticamente os objetos de interesse na imagem. Esse método é ideal para processar grandes volumes de dados ou quando a segmentação precisa ser realizada de forma rápida e eficiente em cenários bem definidos.

3.4 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 próprio, denominado pelo autor como "Método de Espalhamento". Esse método parte do canto da imagem e realiza uma varredura pixel a pixel ao redor de cada ponto analisado, comparando diretamente com o resultado esperado. Caso a cor do pixel na imagem gerada seja diferente da cor prevista no resultado esperado, era gerada uma pontuação negativa. Se as cores coincidisse, o método prosseguia, continuando a busca por outras colorações na imagem base.

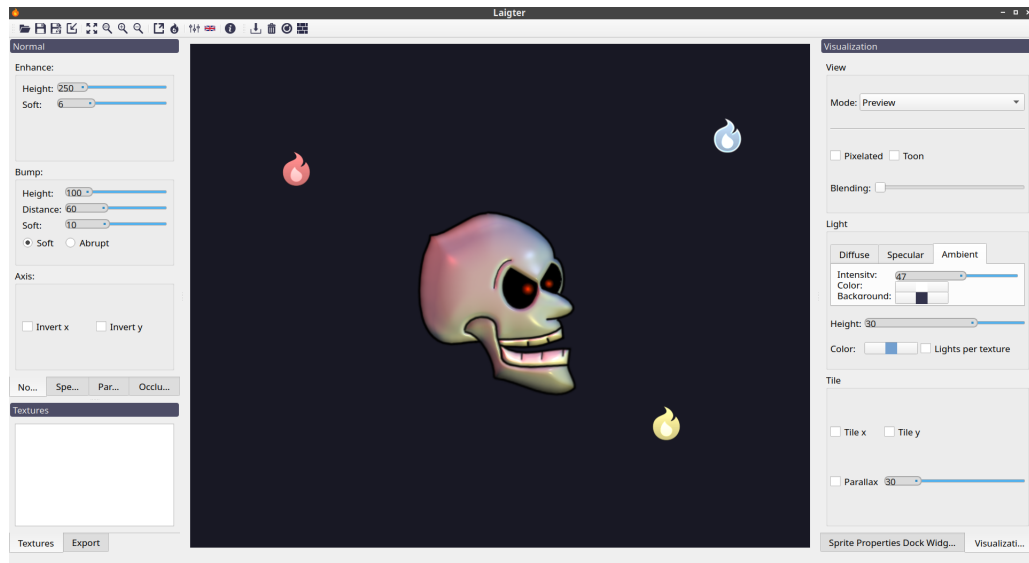
3.5 Avaliar vantagens e limitações das ferramentas abordadas

O *Relight* é uma ferramenta muito boa para processos de iluminação tanto para cenários como personagens, com uma boa modificação como profundidade, posição, cor e luminosidade, dando para o usuário a liberdade de dar personalidade as suas imagens.

Mas ainda existe várias limitações e elas que serão retratadas nessa seção, como foi visto na análise a cima na tabela é possível notar que quanto menor a imagem é, menos funcional ela passa a ser, por exemplo imagens em baixa resolução ou pixeladas começam a ter muitos problemas pois a luz não consegue distinguir os objetos na cena e muito menos paisagens pois com tantas cores perto uma das outras transforma a cena em uma desordem visual.

Além da ferramenta abordada acima existe também algumas outras opções no mercado para este tipo de processo, como o *laigter* mas que ainda se envolve em uma limitação, que é o mapa de normais em apenas uma camada, o *laigter* atual na imagem fornecida pelo usuário como retratado na figura 5.

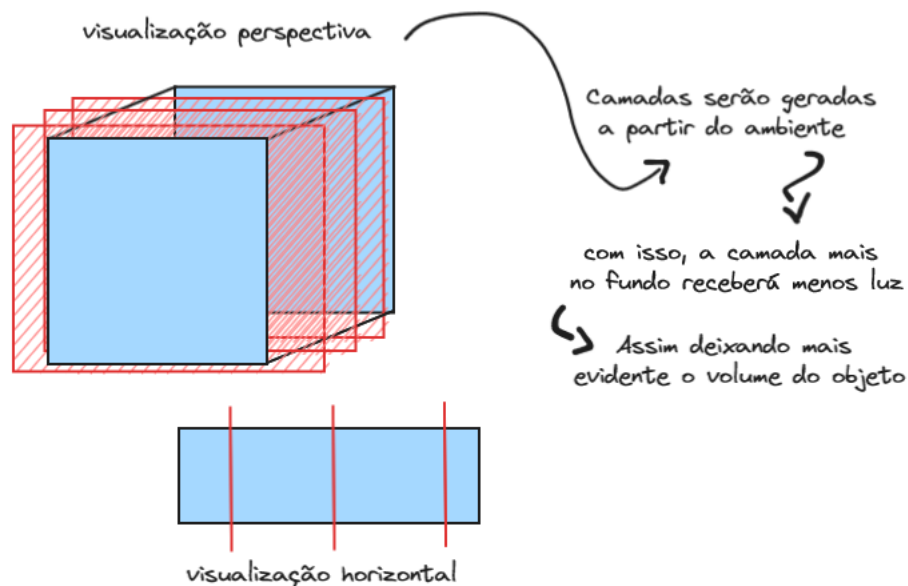
Figura 5 – Imagens retiradas do GitHub da ferramenta



3.6 Modelagem da ferramenta

Uma malha deverá ser criada por cima de toda a imagem e definir um inteiro para a profundidade, cada *pixel* da imagem deve receber uma profundidade como pode se observar na Figura 6

Figura 6 – Rascunho elaborado pelo autor



Assim ele vai numerar toda a malha de *pixels* na tela com números que podem ter uma variação maior dependendo do tamanho da imagem (no caso de exemplo)

Será calculado de 1 a 9, mas com imagens imensas em alta resolução podemos pensar em usar de 1 a 100 ou até mais

Podemos pensar que dessa forma será possível utilizar essa profundidade para produzir uma luz pelas laterais ou pela frente e até atrás de elementos, sem perder sua funcionalidade até mesmo em imagens pequenas

Além disso, cada um desses valores definidos em cada *pixel* da imagem, vão servir de referencia para criação de vários objetos, ele levará em consideração a distância e a variação de cor que foi atingida

Como por exemplo, se existir um objeto a frente da camada 1 até a 10 e um objeto atrás que está na camada 15, essa distancia de 5 camadas irá fazer o objeto dá frente se separar com o de trás transformando a cena com dois objetos ao invés de um

3.7 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 7.

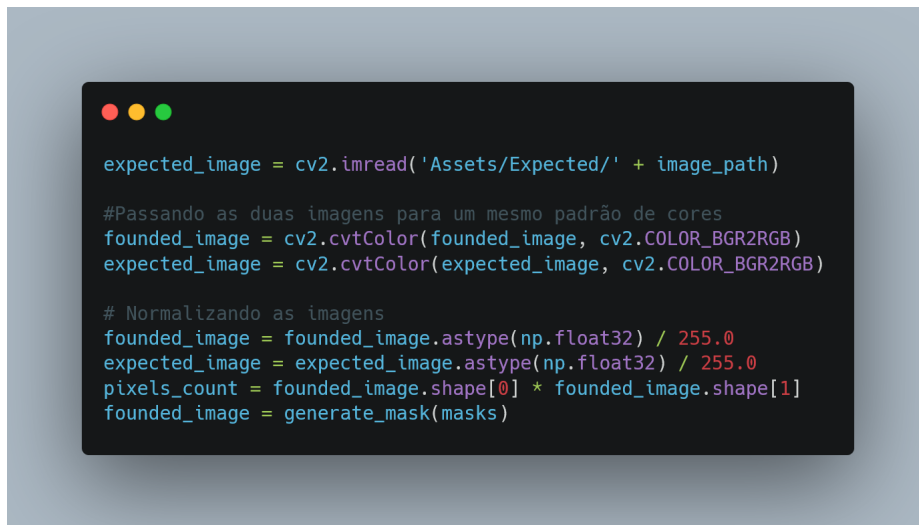
Figura 7 – Código elaborado pelo autor



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 8.

Figura 8 – Código elaborado pelo autor



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

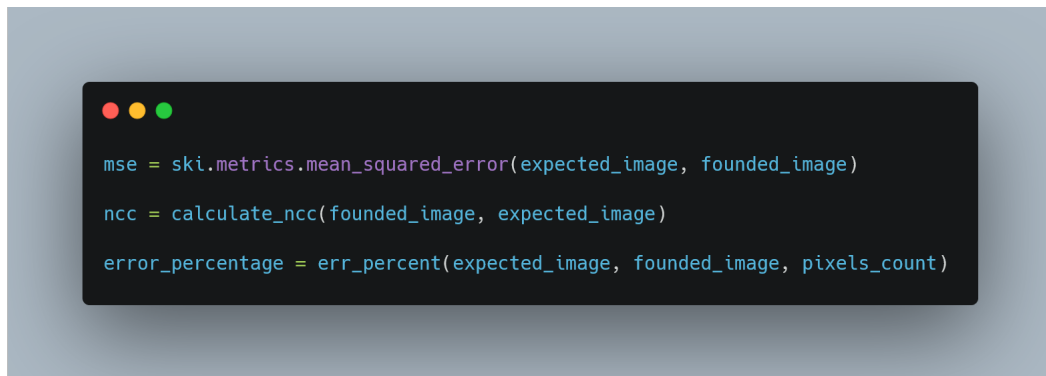
#Passando as duas imagens para um mesmo padrão de cores
founded_image = cv2.cvtColor(founded_image, cv2.COLOR_BGR2RGB)
expected_image = cv2.cvtColor(expected_image, cv2.COLOR_BGR2RGB)

# 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 9.

Figura 9 – Código elaborado pelo autor



```
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.

3.8 Justificativa das Tecnologias a serem adotadas

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 ferramenta pela facilidade com a criação e uso das tecnologias de inteligencia Artificial que irá adiantar muitos dos processos necessários para a criação da análise.

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 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.

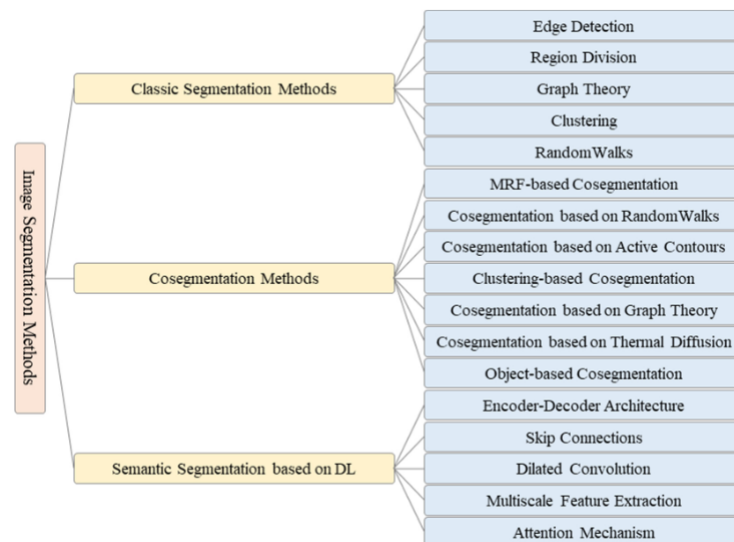
Como o esperado, foi identificado nos computadores usados para executar o código uma quantidade de tempo razoável até sua finalização, apesar disso, entre eles houveram mínimas diferenças de tempo de execução, mesmo com abruptas diferenças de processamento, em seguida será mostrado as propriedades de cada um deles, para uma melhor comparação da média dos resultados obtidos em termos de desempenho, como mostra a 2.

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 no gráfico

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

Tabela 2 – Comparação de Desempenho entre Computador e Notebook

Figura 10 – Análise de desempenho



4.2 Resultados/Impactos

Resultados.

4.3 Cronograma do Trabalho

Segue abaixo o cronograma de trabalho das atividades realizadas até a Avaliação Final de TCC.

1. Elaboração da Introdução
2. Elaboração da Revisão da Literatura

5 CONCLUSÕES PARCIAIS

Foi descrita a modelagem da ferramenta própria, que envolve a criação de uma malha por cima da imagem para definir a profundidade e a variação de cor, permitindo a manipulação da luz. Além disso, foi citado as tecnologias adotadas, como o Visual Studio para programação em *C#* e a Unity como a game *engine* escolhida.

Para avaliar a eficácia da ferramenta, foi planejado realizar uma avaliação qualitativa dos resultados obtidos, incluindo a análise da alteração de imagens, a viabilidade da ferramenta e a eficácia dos cálculos realizados. Para fim é esperado que a ferramenta esteja concluída mesmo que de forma manual, para que ocasionalmente possa ter um progresso posterior em outros trabalhos.

REFERÊNCIAS

- 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
- CEZIMBRA, D. J. **História da Arte e Design**. 1. ed. [S.l.]: Contentus, 2021. 142 p. 5
- 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
- GRAFITTI. **Aprenda a criar efeitos de luz e sombra para as suas ilustrações**. 2019. Disponível em: <<https://blog.grafittiartes.com.br/aprenda-criar-efeitos-de-luz-e-sombra-para-as-suas-ilustracoes>>. Acesso em: 27 de mar. de 2023. 5
- 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
- SHIRLEY, P.; MORLEY, R. **Realistic Ray Tracing, Second Edition**. Taylor & Francis, 2008. ISBN 9781568814612. Disponível em: <<https://books.google.com.br/books?id=knPN6mnhJ8QC>>. 6
- 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