UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROJETO ORIENTADO EM COMPUTAÇÃO II

VICTOR PRATES FIGUEIREDO

SemMatch: Arcabouço para Avaliação e Validação de Modelos de Correspondência Semântica de Imagens

Repositório: github.com/vprates-22/SemMatch

Tipo: Tecnológico

Orientador: Erickson Rangel do Nascimento

Co-orientador: Felipe Cadar Chamone

Belo Horizonte

2025

Resumo

Este trabalho apresenta o desenvolvimento do SemMatch, um arcabouço de software projetado para a avaliação sistemática e validação de modelos de correspondência de imagens. A rápida e contínua sofisticação dos modelos de Deep Learning em Visão Computacional gerou uma necessidade crítica de ferramentas padronizadas que possibilitem testes consistentes, justos e reprodutíveis. O projeto endereça a ausência de uma estrutura automatizada e extensível capaz de ir além das métricas de precisão geométrica, propondo uma análise qualitativa que investiga a influência da informação semântica nas falhas de correspondência.

O SemMatch possui uma arquitetura modular, o que permite a unificação e o suporte a datasets reconhecidos como ScanNet, MegaDepth e HPatches. Sua principal contribuição metodológica é a implementação de um pipeline inovador que utiliza o modelo de segmentação de fundação SAM 2 e a métrica perceptual LPIPS para categorizar os erros em falhas de textura ou erros semânticos genuínos.

O arcabouço final consiste em um sistema robusto em Python para a execução dos testes e análise estatística, complementado por módulos de reporting que geram relatórios estáticos (PDF/HTML) e uma aplicação Flask para a exploração visual dinâmica dos resultados. Ao fornecer uma plataforma completa de diagnóstico e validação, o SemMatch cumpre o objetivo de preencher a lacuna de ferramentas diagnósticas na área, elevando a qualidade e a reprodutibilidade da pesquisa em correspondência de imagens.

Abstract

This work presents the development of SemMatch, a software framework designed for the systematic evaluation and validation of image matching models. The rapid and continuous sophistication of Deep Learning models in Computer Vision has generated a critical need for standardized tools that enable consistent, fair, and reproducible testing. The project addresses the absence of an automated and extensible structure capable of going beyond geometric accuracy metrics, proposing a qualitative analysis that investigates the influence of semantic information on matching failures.

SemMatch has a modular architecture, which allows for the unification and support of recognized datasets such as ScanNet, MegaDepth, and HPatches. Its main methodological contribution is the implementation of an innovative pipeline that utilizes the SAM 2 foundation segmentation model and the LPIPS perceptual metric to categorize errors into texture failures or genuine semantic errors.

The final framework consists of a robust Python system for test execution and statistical analysis, complemented by reporting modules that generate static reports (PDF/HTML) and a Flask application for dynamic visual exploration of the results. By providing a complete diagnostic and validation platform, SemMatch fulfills the objective of filling the gap in diagnostic tools in the area, elevating the quality and reproducibility of research in image matching.

Sumário

1. Introdução

O campo da Visão Computacional vem passando por uma evolução acelerada, impulsionada pelo surgimento de modelos de correspondência de imagens cada vez mais sofisticados. Esses modelos são essenciais para aplicações de alto nível, como navegação autônoma, reconstrução 3D e Localização e Mapeamento Simultâneo (SLAM), que dependem da extração consistente de relações geométricas entre múltiplas vistas. Contudo, apesar do rápido avanço da área, a validação e a comparação desses modelos ainda enfrentam desafios técnicos e metodológicos que afetam tanto a reprodutibilidade quanto a aplicabilidade dos resultados apresentados na literatura. A ausência de um arcabouço de software capaz de padronizar, automatizar e qualificar adequadamente esse processo tem limitado a evolução sistemática da área.

O problema pode ser entendido a partir de duas dimensões centrais. A primeira envolve a fragmentação do ecossistema de benchmarking: a comunidade utiliza diferentes datasets, como ScanNet1500[6], MegaDepth1500[7] e HPatches[8], combinados a métricas que não se articulam em um fluxo integrado. Como consequência, a comparação entre novos modelos se torna um procedimento manual, lento e frequentemente inconsistente. A segunda dimensão refere-se à limitação qualitativa das métricas predominantes, que se concentram em medidas geométricas como AUC de Pose e acurácia. Embora importantes, essas métricas não qualificam a natureza do erro e deixam sem resposta se uma falha de correspondência decorre apenas de elementos pouco texturizados ou se representa uma interpretação semântica equivocada da cena pelo modelo.

Nesse contexto, o objetivo geral deste relatório técnico é apresentar o desenvolvimento do SemMatch, um arcabouço em Python projetado exatamente para enfrentar essas duas limitações estruturais: a fragmentação dos benchmarks e a ausência de métricas que capturem a dimensão semântica dos erros de correspondência. O SemMatch foi concebido como uma solução de engenharia completa, capaz de oferecer tanto padronização quanto profundidade analítica.

O desenvolvimento do arcabouço concentrou-se na consolidação de uma arquitetura modular que unifica múltiplos benchmarks sob uma interface consistente. Essa reestruturação permitiu que novos datasets e módulos de avaliação pudessem ser integrados de forma simples, garantindo a longevidade e a extensibilidade do projeto. Além da unificação arquitetural, o SemMatch introduz um pipeline metodológico voltado à análise semântica dos erros. Utilizando modelos de segmentação, como o SAM 2, combinados a métricas perceptuais como o LPIPS, o arcabouço é capaz de qualificar os tipos de falha cometidos pelos modelos, preenchendo uma lacuna que até então não era tratada pela literatura.

Outro eixo fundamental do desenvolvimento foi a criação de módulos de relatório capazes de apresentar resultados tanto em interfaces interativas quanto em formatos estáticos, como PDF e HTML, facilitando a documentação formal e a disseminação dos experimentos. Por fim, o SemMatch foi concebido como um projeto de código aberto, alojado em repositório público e construído com padrões rigorosos de engenharia, incluindo testes unitários, linters e documentação completa, permitindo que seja adotado e expandido de maneira confiável.

Assim, este relatório apresenta não apenas a motivação e o contexto que justificam o SemMatch, mas também as soluções técnicas que o constituem e que o posicionam como um arcabouço robusto, extensível e extremamente útil para a avaliação moderna de modelos de correspondência de imagens.

2. Referencial Teórico

O desenvolvimento do SemMatch foi fortemente guiado por trabalhos recentes que propuseram abordagens eficientes e modulares para o problema de correspondência de imagens. As soluções exploradas no Glue Factory[1] e no DescriptorReasoning_ACCV_2024[2] foram particularmente influentes na definição da arquitetura do framework, principalmente na forma como os modelos foram estruturados, integrados e avaliados. Ambos os trabalhos apresentam pipelines organizados e bem definidos, com separação clara entre módulos responsáveis pela correspondência, entrada de dados e visualização, o que inspirou diretamente a estrutura modular adotada neste projeto.

Além disso, para o entendimento de como os modelos atuais implementam a semântica no momento das correspondências, o trabalho de Cadar et al.[3] serviu como principal base conceitual. O trabalho demonstra como a incorporação de informações semânticas em modelos de pareamento de imagem pode levar a resultados melhores que os obtidos atualmente.

A extração de máscaras semânticas foi viabilizada com o uso do Segment Anything Model 2 (SAM 2)[4], cuja capacidade de segmentação de objetos em imagens complexas permitiu realizar mapeamentos mais precisos entre os pontos incorretos e suas respectivas regiões na cena. Essa ferramenta tornou possível a construção de visualizações baseadas em máscaras e pontos, fundamentais para a análise qualitativa implementada na interface.

Por fim, o uso da métrica LPIPS[5], permitiu quantificar a similaridade perceptual entre objetos distintos, mesmo quando estes se apresentavam em regiões com aparências semelhantes. Essa métrica foi utilizada para estimar o grau de erro semântico nas correspondências, servindo como um componente complementar à análise geométrica e à segmentação.

Dessa forma, os trabalhos mencionados não apenas fundamentaram teoricamente o projeto, mas tiveram papel prático direto na definição das estratégias de implementação, organização da arquitetura e construção dos módulos de análise visual e semântica.

3. Desenvolvimento

O SemMatch foi desenvolvido a partir de uma filosofia de arquitetura modular e orientada a componentes, concebida para maximizar extensibilidade, reutilização de código e reprodutibilidade científica. Essa abordagem foi determinante para integrar múltiplos benchmarks, organizar diferentes etapas de avaliação e incorporar um pipeline semântico computacionalmente complexo. A separação clara de responsabilidades, aliada à coordenação centralizada da execução, garante que o arcabouço opere de maneira eficiente, escalável e metodologicamente consistente.

3.1. Arquitetura do SemMatch

O SemMatch foi escrito em Python e organizado em módulos interdependentes e coordenados por um componente central, o *Evaluator*. Este componente atua como ponto de entrada do arcabouço, carregando as configurações definidas pelo usuário, inicializando os datasets, instanciando o Orchestrator e gerenciando a execução do modelo de correspondência. O *Orchestrator*, por sua vez, coordena o pipeline analítico, incluindo geração de dados intermediários, execução dos *Analyzers* e consolidação das métricas.

O módulo de datasets realiza a padronização das bases utilizadas, permitindo que benchmarks como HPatches, MegaDepth e ScanNet sejam integrados de maneira coerente. O módulo statistics centraliza a análise de desempenho, com geradores de dados brutos, para estimativa de pose e preparação da análise semântica, *Analyzers* que calculam resultados geométricas e semânticas, e implementações das métricas tradicionais e personalizadas. Por fim, o módulo report transforma os resultados em visualizações interativas via Flask e relatórios estáticos em HTML e PDF, oferecendo suporte tanto à exploração detalhada quanto à documentação formal.

A separação modular permitiu isolar componentes computacionalmente pesados, como o pipeline semântico baseado no SAM 2 e LPIPS, garantindo carregamento persistente em GPU e reutilização eficiente de máscaras semânticas, reduzindo o tempo de execução em larga escala.

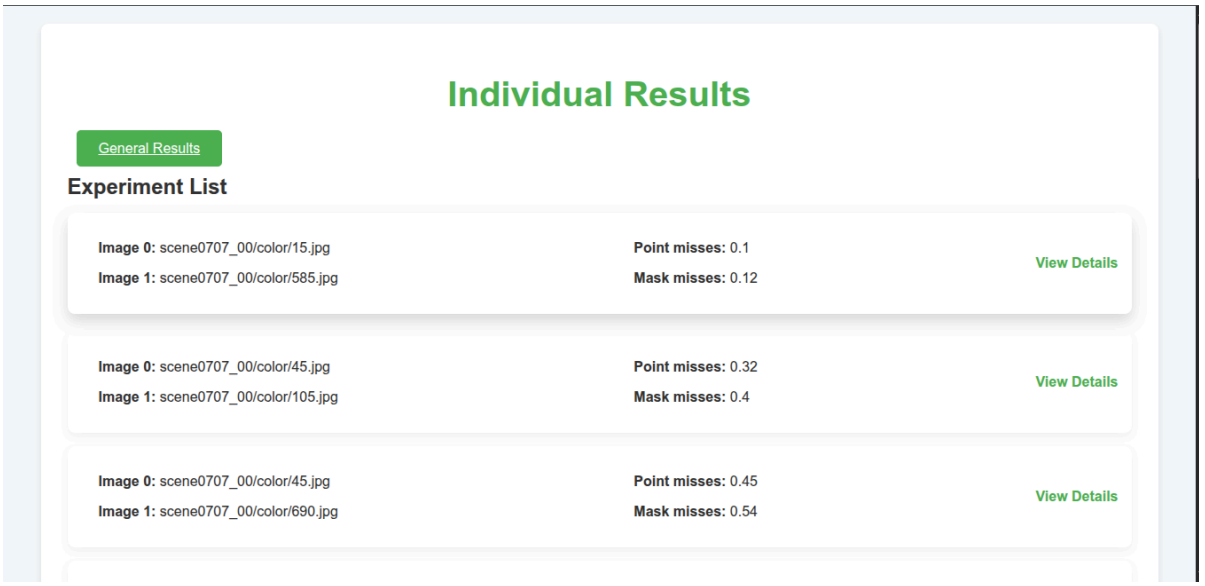
3.2. Pipeline de Análise Semântica

Após a execução do modelo de correspondência e a validação geométrica das correspondências via estimador de poses, os pontos classificados como incorretos são processados pelo pipeline semântico. Primeiramente, o SAM 2 é utilizado para segmentar a imagem de destino, identificando as regiões semânticas correspondentes a cada keypoint. Em seguida, os pontos projetados da imagem de origem são comparados com as máscaras na imagem de destino para determinar se a falha ocorre dentro do mesmo objeto (erro de textura) ou entre objetos diferentes (erro semântico).

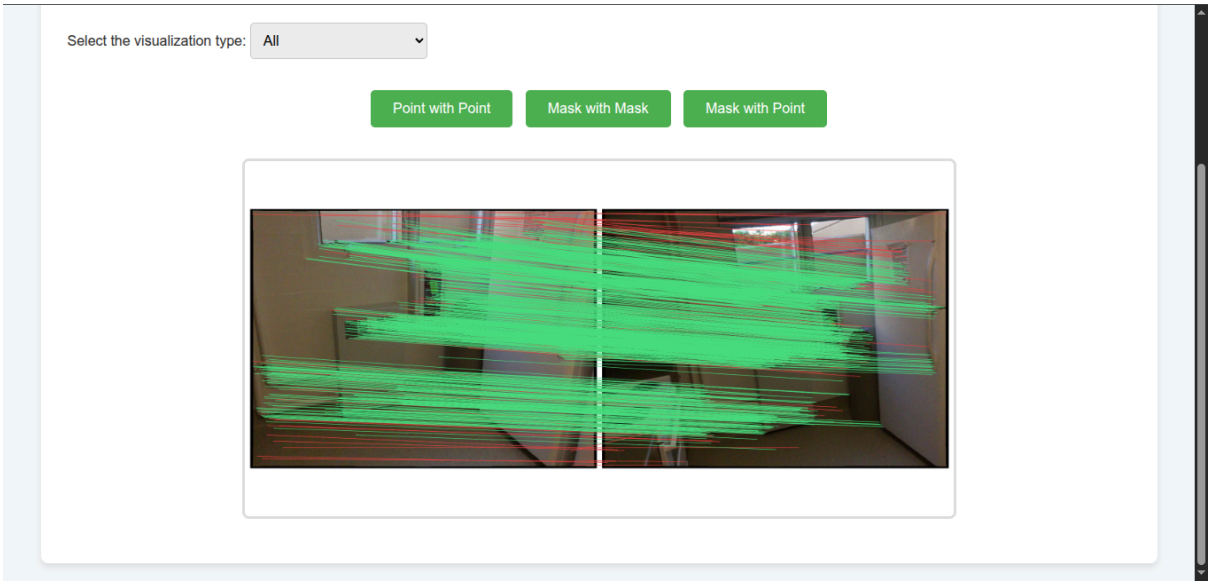
Nos casos em que os pontos incorretos pertencem a objetos diferentes, são extraídos patches centrados nos pontos corretos e incorretos, e a métrica LPIPS é aplicada para avaliar a similaridade perceptual entre eles. Valores altos indicam erro semântico genuíno, enquanto valores baixos revelam regiões visualmente semelhantes, oferecendo insights adicionais sobre limitações de generalização do modelo.

3.3. Reportes e Visualização

Os resultados consolidados são apresentados em uma interface web interativa desenvolvida com Flask. A página principal oferece uma visão geral das estatísticas de desempenho, incluindo taxas de acerto, erro e métricas derivadas da análise semântica. A partir dessa visão geral, é possível acessar detalhes de cada par de imagens, com informações sobre correspondências corretas e incorretas.



A visualização é organizada em três modos. O modo "pontos com pontos" exibe as imagens lado a lado com linhas conectando os keypoints, coloridas conforme a classificação das correspondências. O modo "máscaras com máscaras" destaca objetos segmentados, mostrando em cores diferentes quando pares incorretos pertencem ao mesmo objeto ou a objetos distintos. O modo "máscaras com pontos" permite analisar a relação entre as segmentações e os pontos, exibindo a máscara de origem e os keypoints correspondentes na imagem de destino, com diferenciação de acertos e erros e opções de filtragem interativa. Essa abordagem possibilita exploração detalhada e compreensão completa do comportamento dos modelos.



Além da interface interativa, o SemMatch gera relatórios estáticos em HTML e PDF que consolidam todas as métricas e análises realizadas. Esses relatórios oferecem uma visão estruturada dos resultados, permitindo documentação formal e compartilhamento dos experimentos de forma independente da interface web. O formato HTML mantém elementos visuais, como gráficos e tabelas interativas, enquanto o PDF fornece uma versão pronta para impressão e publicação científica. Dessa forma, os reportes do SemMatch atendem tanto à exploração detalhada durante a fase de desenvolvimento quanto à disseminação formal dos resultados.

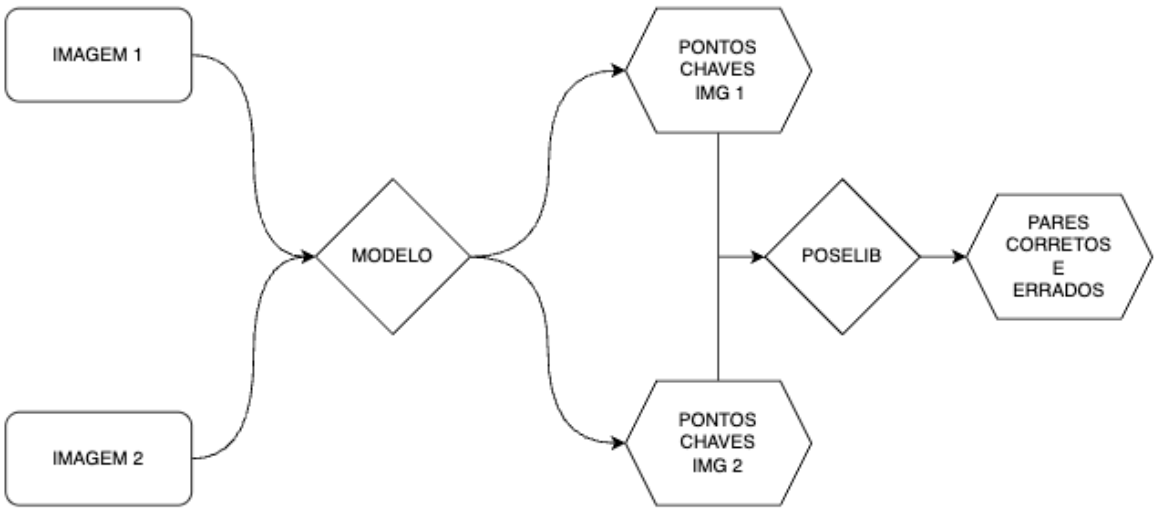
Semantic

	Accuracy	Precision	Recall	FalsePositiveRatio	F1Score
value	0.814	0.894	0.76	0.115	0.821
mean	0.806	0.89	0.676	0.115	0.745
std	0.079	0.075	0.217	0.095	0.179
min	0.602	0.713	0.099	0.0	0.175
25%	0.741	0.824	0.555	0.043	0.668
50%	0.798	0.91	0.739	0.086	0.799
75%	0.873	0.95	0.859	0.17	0.886
max	0.962	1.0	0.934	0.427	0.938

4. Metodologia de Avaliação e Análise

4.1. Avaliação Geométrica

A primeira etapa da metodologia consiste na avaliação geométrica dos modelos. Os pares de keypoints extraídos pelo modelo são filtrados com base em sua consistência geométrica, utilizando estimativas de pose para descartar correspondências incompatíveis. Apenas os pontos válidos são considerados para o cálculo das métricas principais.



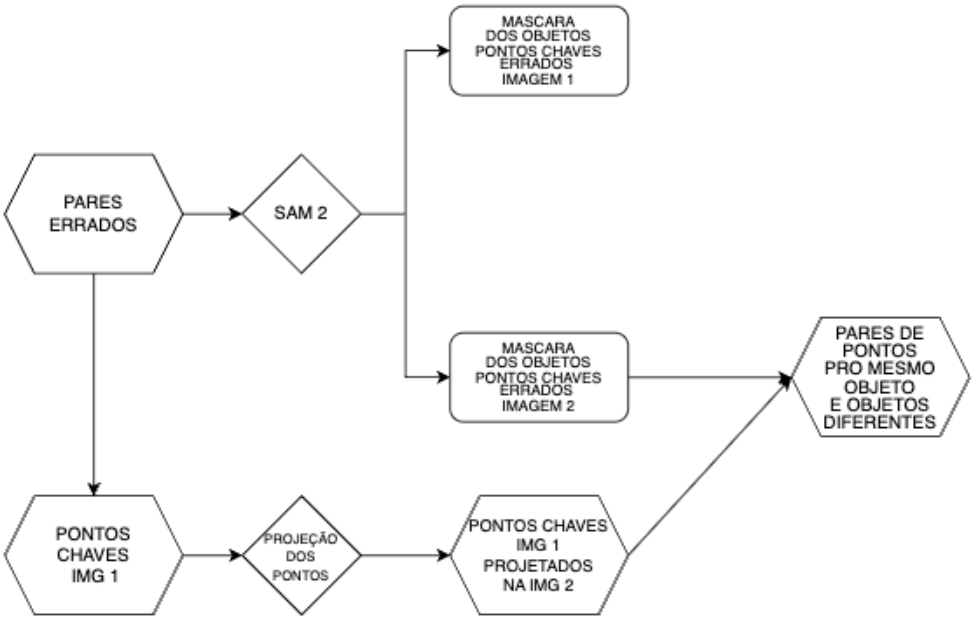
O desempenho é quantificado pela AUC de Pose, que mede a precisão das estimativas de rotação e translação geradas a partir das correspondências válidas. Essa métrica oferece uma visão consolidada do alinhamento geométrico obtido pelo modelo ao longo de diferentes limiares de erro.

4.2. Pipeline de Análise Semântica

Após a validação geométrica, os erros restantes são analisados pelo pipeline semântico. Este processo permite diferenciar falhas de textura de falhas semânticas, fornecendo uma avaliação qualitativa mais profunda.

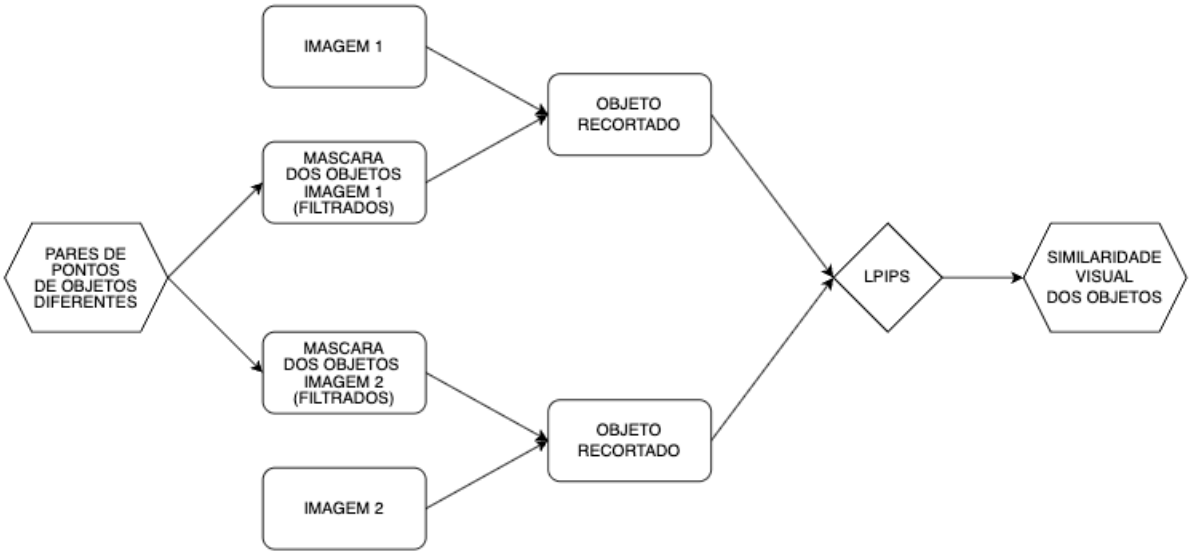
4.2.1. Segmentação e Análise Estrutural

O SAM 2 segmenta a imagem de destino para identificar regiões correspondentes aos keypoints incorretos. A projeção dos pontos da imagem de origem é comparada com as máscaras segmentadas. Quando ambos pertencem à mesma máscara, a falha é classificada como erro de textura. Caso contrário, é identificado um possível erro semântico, indicando confusão entre objetos diferentes.



4.2.2. Qualificação Perceptual com LPIPS

Para confirmar a natureza semântica do erro, são extraídos patches centrados nos pontos correto e incorreto e calculada a similaridade perceptual via LPIPS. Valores altos reforçam a presença de erro semântico, enquanto valores baixos indicam que, apesar de semanticamente distintos, os objetos apresentam similaridade visual significativa.



4.3. Categorização dos Erros

O pipeline classifica cada falha em erro de textura, quando ocorre dentro do mesmo objeto, ou erro semântico, quando envolve objetos distintos e é confirmado por LPIPS, quando a similaridade perceptual dos objetos se encontra abaixo de certo limiar. Essa categorização permite interpretar o comportamento do modelo em nível local e de alto nível, distinguindo limitações de precisão de limitações de compreensão semântica.

5. Conclusão

Este projeto alcançou o objetivo de projetar e desenvolver o SemMatch, um arcabouço de software modular e extensível voltado à avaliação de modelos de correspondência de imagens. O desenvolvimento abordou de forma direta a necessidade de padronização de benchmarks e supriu a lacuna semântica existente na análise qualitativa de erros em Visão Computacional, oferecendo uma ferramenta capaz de realizar avaliações sistemáticas e diagnósticos detalhados.

5.1. Sumário do Trabalho e Contribuições

O projeto demonstrou alto nível de maturidade técnica e rigor acadêmico, resultando em contribuições significativas para a pesquisa em correspondência de imagens. O SemMatch estabelece um arcabouço de avaliação unificado e modular, desenvolvido em Python, com arquitetura flexível que facilita a integração de novos modelos e datasets. A unificação de benchmarks essenciais, como ScanNet, MegaDepth e HPatches, permite a execução de testes consistentes e reprodutíveis, promovendo comparações justas entre diferentes abordagens.

Além disso, o arcabouço introduziu uma metodologia inovadora de análise semântica, baseada na segmentação de objetos pelo SAM 2 e na métrica perceptual LPIPS. Essa abordagem permite distinguir erros de textura local de falhas de interpretação semântica, fornecendo informações diagnósticas valiosas sobre os modelos avaliados. O SemMatch também entrega uma solução de engenharia completa, com um Orchestrator eficiente e módulos de reporting capazes de gerar relatórios estáticos em PDF e HTML, bem como visualizações dinâmicas por meio de uma interface web interativa, garantindo comunicabilidade e utilidade científica.

5.2. Cumprimento dos Objetivos

Todos os objetivos específicos definidos nas fases POC I e POC II foram integralmente cumpridos. A reestruturação da arquitetura no POC II foi decisiva para superar os desafios de performance associados ao uso de modelos de segmentação de alta complexidade, como o SAM 2, garantindo a viabilidade do pipeline semântico em larga escala.

5.3. Perspectivas e Trabalhos Futuros

Embora o SemMatch represente uma solução funcional, madura e pronta para uso, sua arquitetura modular permite diversas extensões. Entre as principais direções de pesquisa e desenvolvimento futuras, destacam-se a integração de novos modelos de segmentação e de descrição visual, a avaliação de performance computacional e a ampliação da produção de relatórios.

Em particular, faz sentido validar o SemMatch com diferentes modelos de correspondência de imagem — atualmente o arcabouço está preparado para aceitar novos métodos, mas ainda carece de uma avaliação aprofundada com uma gama mais ampla de algoritmos de matching, de forma a verificar sua robustez e generalidade. De modo similar, os experimentos com segmentação poderiam se beneficiar da testagem de modelos alternativos ao SAM 2, buscando soluções potencialmente mais eficientes ou adequadas para determinados domínios. Entre os candidatos recomendados para investigação estão arquiteturas clássicas e modernas de segmentação, como DeepLab v3+, Mask R-CNN, SegFormer, dentre outros.

Outro ponto relevante diz respeito à distinção de instâncias diferentes de um mesmo tipo de objeto. Atualmente, o SemMatch identifica erros semânticos com base em máscaras de objeto, mas não distingue quando dois objetos distintos pertencem à mesma classe, como dois girassóis diferentes. Desenvolver mecanismos para reconhecer instâncias separadas, mesmo dentro da mesma categoria semântica, permitiria uma análise mais precisa de erros e generalização do modelo.

Adicionalmente, poderia ser incorporada a geração de imagens e visualizações nos relatórios estáticos (HTML/PDF), de forma a tornar os resultados mais completos e autoexplicativos. Por exemplo, trechos selecionados de correspondências (corretas e incorretas) poderiam ser automaticamente incluídos nos relatórios, com visualização das máscaras, keypoints, comparações semânticas e métricas associadas, facilitando a análise offline e a disseminação dos resultados sem necessidade de interface web interativa.

6. Referências

[1] Computer Vision and Geometry Lab (2023). glue-factory. GitHub. https://github.com/cvg/glue-factory

[2] Laboratory of Computer Vision & Robotics - VeRLab (2024). DescriptorReasoning_ACCV_2024. GitHub. https://github.com/verlab/DescriptorReasoning_ACCV_2024

[3] Cadar, F., Potje, G., Martins, R., Demonceaux, C., & Nascimento, E. R. (2024). Leveraging Semantic Cues from Foundation Vision Models for Enhanced Local Feature

Correspondence. In Proceedings of the Asian Conference on Computer Vision (pp. 1268-1283).

[4] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Räedle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). SAM 2: Segment Anything in Images and Videos. arXiv. https://arxiv.org/abs/2408.00714

[5] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 586–595).

[6] Li, Zhengqi, and Noah Snavely. "Megadepth: Learning single-view depth prediction from internet photos." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[7] Dai, Angela, et al. "Scannet: Richly-annotated 3d reconstructions of indoor scenes." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[8] HPatches: A benchmark and evaluation of handcrafted and learned local descriptors, Vassileios Balntas*, Karel Lenc*, Andrea Vedaldi and Krystian Mikolajczyk, CVPR 2017. [arXiv pdf] (https://arxiv.org/pdf/1704.05939.pdf)