

**UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA**

MAC0215: Atividade Curricular em Pesquisa

Fluxo Óptico para Análise de Calçadas

Renzo Real Machado Filho

Orientadores:

Roberto Marcondes Cesar Junior

Rafael Jeferson Pezzuto Damaceno

**São Paulo
2025**

Resumo

Este projeto tem como objetivo introduzir os fundamentos de visão computacional e aprendizado de máquina multimodal, com foco na análise de dados urbanos por meio do sistema SideSeeing. O trabalho concentrou-se no estudo e na implementação de algoritmos de fluxo óptico, uma ferramenta essencial para detectar padrões de movimento e obstáculos em vídeos capturados em campo. A metodologia adotada explorou desde abordagens clássicas, como os métodos de Lucas-Kanade e Horn-Schunck, até o estado da arte baseado em *Deep Learning*. Nesse contexto, foi avaliado o desempenho da arquitetura DP-Flow (*Dual-Pyramid Flow*) em dados reais, verificando-se sua superioridade na preservação de bordas e robustez a ruídos em comparação aos métodos tradicionais. Paralelamente à análise visual, o projeto abrangeu a engenharia de dados multimodais com o desenvolvimento do módulo *SideSeeing Exporter*, uma ferramenta de automação para a visualização integrada de sensores, mapas de calor e rotas GPS. Ao final, a pesquisa consolidou um domínio introdutório em visão computacional prática e contribuiu com ferramentas essenciais para a validação técnica e futura análise semântica de acessibilidade urbana.

Palavras-chave: Visão Computacional, Fluxo Óptico, Acessibilidade Urbana, SideSeeing, Deep Learning.

Sumário

1	Introdução	3
2	Materiais e Métodos	3
3	Resultados e Discussões	4
3.1	Visão e Processamento de Imagens	4
3.2	Fluxo Óptico	6
3.2.1	O Método de Lucas e Kanade	6
3.2.2	A Representação Matiz e Saturação	8
3.2.3	O Método Global de Horn e Schunck	9
3.2.4	O Estado da Arte	11
3.2.5	Redes Neurais Convolucionais (CNNs)	12
3.2.6	DPFlow: Adaptive Optical Flow with Dual-Pyramid . . .	13
3.3	Sideseeing Project	15
4	Conclusões	17

1 Introdução

O planejamento urbano contemporâneo enfrenta o desafio de equilibrar mobilidade, sustentabilidade e vitalidade. Nesse contexto, o avanço da inteligência artificial e da visão computacional oferece ferramentas poderosas para a análise automatizada desses ambientes. A capacidade de computadores interpretarem imagens, vídeos e dados sensoriais de forma integrada, por meio do aprendizado multimodal, é fundamental para extrair padrões que podem orientar o desenvolvimento de políticas públicas mais eficazes.

Este relatório detalha as atividades desenvolvidas ao longo do semestre de Iniciação Científica. O trabalho estrutura-se na utilização do ecossistema *SideSeeing* para a manipulação, o processamento e a interpretação de conjuntos de dados multimodais, aplicados à acessibilidade urbana.

A pesquisa concentra-se no estudo e na implementação de algoritmos de fluxo óptico. Essa técnica, essencial para a detecção de movimento e obstáculos, permite estimar a dinâmica de pixels em vídeos capturados em campo. Embora os métodos clássicos tenham sido cruciais para estabelecer os pilares da área, a aplicação em cenários reais exige, atualmente, abordagens baseadas em *Deep Learning*.

Nesta primeira etapa do projeto, o foco recaiu sobre a fundamentação teórica — partindo do processamento básico de imagens até o estado da arte em métodos para fluxo óptico — e na engenharia de dados necessária para a organização e visualização das coletas realizadas.

2 Materiais e Métodos

A metodologia adotada durante esse período combinou revisão bibliográfica com implementação prática, estruturando-se em três frentes principais: processamento de imagens, algoritmos de fluxo óptico e engenharia de dados multimodais. Os dados utilizados ao longo do projeto foram coletados através do framework *SideSeeing* [1] utilizando dispositivos móveis. Para o desenvolvimento dos algoritmos e das ferramentas de análise, empregaram-se bibliotecas de visão computacional e ciência de dados na linguagem Python. Destacam-se o uso do OpenCV, para a manipulação básica de imagens e implementação de algoritmos clássicos; NumPy e Pandas, para a manipulação vetorial e estruturação dos dados dos sensores; e as bibliotecas Plotly e Leaflet, aplicadas na construção de visualizações interativas e mapas geoespaciais.

A abordagem técnica seguiu uma evolução cronológica e de complexidade, permitindo comparar a eficácia de diferentes paradigmas. Inicialmente, foram implementadas operações de ponto, como ajuste de brilho e balanço de cores, além de técnicas de equalização de histograma, com o objetivo de normalizar as condições de iluminação dos quadros de vídeo antes da estimativa de movimento. Na sequência, foram estudados métodos de fluxo óptico baseados em restrições matemáticas tradicionais, como o algoritmo de Lucas-Kanade [2], uma abordagem esparsa avaliada por sua capacidade de rastrear pontos de in-

teresse locais, e o método de Horn-Schunck [3], uma abordagem densa baseada em regularização global, utilizada para gerar campos vetoriais em regiões homogêneas da imagem.

Para superar as limitações dos métodos clássicos em cenários não controlados, a investigação avançou para a aplicação de Redes Neurais Convolucionais (CNNs). O trabalho culminou na avaliação do modelo DPFlow (*Dual-Pyramid Flow*) [4]. Essa arquitetura foi testada com vídeos reais do conjunto de dados, verificando-se sua robustez na preservação de bordas de movimento e na redução de ruído em áreas estáticas, superando os desafios encontrados nas abordagens puramente matemáticas.

Paralelamente ao estudo dos algoritmos, e para viabilizar a interpretação ágil dos dados coletados, foi desenvolvido um módulo de automação denominado *SideSeeing Exporter* (arquivo `export.py`). Essa ferramenta processa os metadados brutos e os sinais dos sensores para gerar *dashboards* interativos em HTML. O sistema integra mapas de calor (*HeatMaps*) da intensidade de sinais Wi-Fi e rotas GPS, gráficos sincronizados de acelerômetro e giroscópio, além de uma visualização unificada das coletas, facilitando a validação técnica dos experimentos.

3 Resultados e Discussões

Nessa seção, mostraremos o aprendizado coletado ao longo do semestre. Passaremos por conceitos fundamentais de visão computacional e processamento de imagens até chegar em fluxo óptico. Partindo para uma parte mais prática, veremos, também, aplicações no projeto SideSeeing.

3.1 Visão e Processamento de Imagens

O estudo de visão computacional e processamento de imagens busca, através de uma série de técnicas, recuperar a forma tridimensional e a aparência de objetos em imagens. Em geral, queremos descrever o mundo e reconstruir suas propriedades, como forma, iluminação e distribuição de cores [5].

Abordaremos, inicialmente, as transformações de imagens mais básicas, conhecidas como *operações de ponto*. Essas transformações manipulam cada pixel independentemente de seus vizinhos. Nessa categoria, incluem-se o ajuste de brilho e de contraste, a correção gama, a transformação de cores, a composição, o mascaramento e a equalização de histograma. A seguir, temos alguns exemplos dessas aplicações.



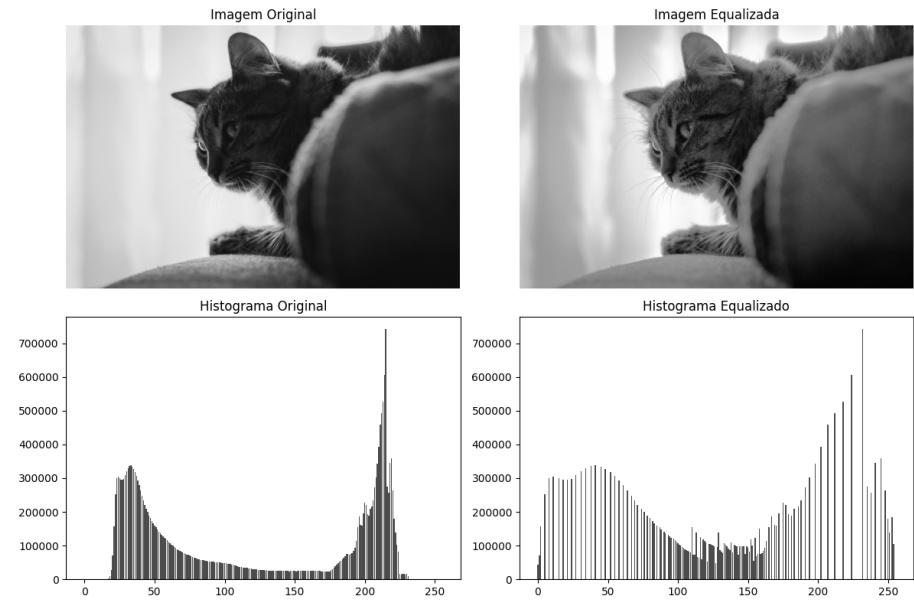
Na figura acima, implementamos um algoritmo de balanço de cores. Para modificar a tonalidade de uma imagem, basta multiplicar os valores de intensidade de cada canal (RGB). Matematicamente, para um pixel p com componentes (R, G, B) , essa transformação é dada por $p' = (\alpha, \beta, \gamma) \cdot (R, G, B)$, onde α, β, γ são os coeficientes de alteração.

Outro problema interessante é tratar imagens com baixo contraste. Para isso, podemos utilizar a técnica de Equalização de Histograma. Um histograma de uma imagem representa a frequência de cada nível de intensidade de cinza (ou cor). Assim, uma imagem escura, por exemplo, terá a maioria de seus pixels com valores de intensidade baixos, resultando em um histograma desbalanceado.

A equalização atua redistribuindo igualmente esses valores de intensidade, com objetivo de aproximar uma distribuição uniforme, onde cada nível tenha, idealmente, o mesmo número de pixels. Ao fazer isso, a diferença de intensidade entre os pixels é acentuada, o que melhora significativamente o contraste global da imagem. Formalmente, é necessário computar a Função de Distribuição Acumulada (CDF), denotada por

$$c(I) = \frac{1}{N} \sum_{i=0}^I h(i) = c(I-1) + \frac{1}{N} h(I),$$

onde I é o nível de intensidade atual (0-255 para imagens 8-bit), $h(I)$ é a frequência do nível de intensidade I (valor do histograma) e N é quantidade de pixels na imagem. A seguir, um exemplo de aplicação.



3.2 Fluxo Óptico

Enquanto o processamento de imagens tradicional, discutido na seção anterior, foca na análise de quadros estáticos, o fluxo óptico introduz uma dimensão temporal, sendo uma das técnicas fundamentais para a análise de movimento em vídeos.

Formalmente, o fluxo óptico é definido como o campo de vetores que descreve o movimento aparente dos pixels ou os padrões de brilho entre frames consecutivos de uma sequência de imagens [3].

No contexto deste projeto, ele será uma ferramenta futura para quantificar o movimento da câmera em relação ao ambiente e para identificar e/ou segmentar obstáculos na cena. Passaremos, nas seções seguintes, por algoritmos clássicos até chegarmos nas arquiteturas modernas de *Deep Learning*.

3.2.1 O Método de Lucas e Kanade

Historicamente, a estimativa de fluxo óptico apoiou-se em métodos numéricos que introduzem suposições locais para viabilizar sua solução. Essas abordagens são consequência do fato de que a equação de restrição de fluxo óptico (1), por si só, constitui um problema malposto. Tal equação não fornece informação suficiente para determinar o vetor de movimento dos pixels, permitindo apenas a determinação de uma das componentes. Esta ambiguidade é conhecida como o Problema da Abertura.

Para contornar isso, o método de Lucas e Kanade [2] introduz uma suposição extra que agrupa informação local:

O campo de movimento e o fluxo óptico são constantes para cada pixel dentro de uma pequena vizinhança W .

Seja $x = [u, v]^T$ constante para todos os pixels $(l, k) \in W$, a equação de restrição de fluxo é

$$I_x(l, k)u + I_y(l, k)v + I_t(l, k) = 0 \quad (1)$$

onde os termos I_x , I_y e I_t representam, respectivamente, as derivadas parciais da função de intensidade da imagem $I(x, y, t)$.

Esta suposição transforma o problema em um sistema linear sobredeterminado da forma $Ax = B$, onde A é a matriz $(n \times 2)$ contendo os gradientes espaciais de todos os n pixels na vizinhança W , x é o vetor de fluxo óptico $[u, v]^T$ e B é o vetor $(n \times 1)$ contendo o negativo dos gradientes temporais, $B = -[I_t(l_1, k_1), \dots, I_t(l_n, k_n)]^T$. Com isso, podemos resolver eficientemente por mínimos quadrados, $x = (A^T A)^{-1} A^T B$.

Todavia, ainda é necessário atentar-se as propriedades da matriz $A^T A$. Para que haja uma solução única, duas condições são necessárias: $A^T A$ deve ser inversível e deve ser bem condicionada.

A análise do condicionamento é realizada através dos autovalores, λ_1 e λ_2 . Idealmente, a razão entre eles deve ser próxima de ($|\frac{\lambda_1}{\lambda_2}| \approx 1$) e devem ter magnitude significativamente maior que zero ($\lambda_1, \lambda_2 > \epsilon$). A análise destes autovalores permite classificar a qualidade da estimativa de fluxo em diferentes tipos de regiões da imagem.

Assim, isso nos leva à maior limitação do método de Lucas e Kanade. Como a matriz, em geral, só é bem condicionada em regiões de alta variação de gradiente em ambas as direções, como em cantos e regiões texturizadas, a técnica falha em bordas retas ou áreas planas. O resultado final é, portanto, um fluxo óptico esparso, calculado apenas em pontos de interesse confiáveis e não em toda a imagem. Nas próximas imagens, teremos alguns exemplos que demonstram esse fato.



Figura 1: Estimativa de fluxo óptico usando Lucas-Kanade em um vídeo gravado próximo ao IME-USP



Figura 2: Estimativa de fluxo óptico usando Lucas-Kanade aplicado em frames disponíveis no dataset Moseg [6]

Observe que a análise de fluxo óptico ficou restrito a poucos pontos e à regiões específicas da imagem, como previsto anteriormente, já que o algoritmo ignora áreas homogêneas — como o asfalto ou a superfície lisa dos carros. Embora essa abordagem seja robusta para o rastreamento de certos pontos, ela falha em fornecer uma descrição completa da cena. Por isso, nas próximas seções, discutiremos outras maneiras de estimar o movimento dos pixels.

3.2.2 A Representação Matiz e Saturação

Enquanto no método de Lucas e Kanade os vetores de movimento são calculados apenas em pontos de interesse, o que permite o uso de diagramas de setas, os métodos densos que veremos a seguir procuram gerar um vetor para cada pixel da imagem. Nesses casos, a representação por setas torna-se inviável

e visualmente poluída. Portanto, uma alternativa muito comum na literatura é a representação contínua baseada em cores.

Essa representação fundamenta-se no espaço de cor HSV (*Hue*, *Saturation*, *Value*), mapeando os vetores de fluxo óptico para componentes cromáticas da imagem. O resultado é um mapa de fluxo colorido que permite uma leitura da dinâmica da cena.

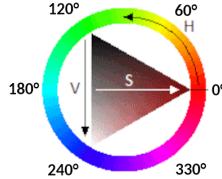


Figura 3: Componentes do sistema de cores HSV, imagem disponível em [7]

Matiz (*H - Hue*): Codifica a direção do vetor de movimento. O ângulo do vetor de fluxo é mapeado diretamente para o círculo cromático (0° a 360°). Por convenção,:

- 0° (Vermelho): Movimento para a direita;
- 90° (Amarelo/Verde): Movimento para baixo;
- 180° (Ciano): Movimento para a esquerda;
- 270° (Azul/Magenta): Movimento para cima.

Dessa forma, regiões com direções de movimento similares compartilharão a mesma tonalidade na imagem.

Saturação (*S - Saturation*): Representa a magnitude da velocidade do movimento. Existe uma relação direta entre a norma do vetor e a pureza da cor,

- *Baixa magnitude*: Resulta em cores dessaturadas (próximas ao branco ou cinza), indicando pouco ou nenhum movimento.
- *Alta magnitude*: Resulta em cores vivas e vibrantes, indicando altas velocidades.

Valor (*V - Value*): É tipicamente fixado para maximizar a luminosidade e garantir que a percepção visual seja guiada exclusivamente pelas variações de direção (Matiz) e velocidade (Saturação).

3.2.3 O Método Global de Horn e Schunck

Em contraste com a abordagem esparsa inicial, o método de Horn e Schunck [3] foi pioneiro em uma abordagem global para a estimativa do fluxo

óptico. O objetivo deste método é calcular um campo de fluxo denso, ou seja, um vetor de movimento para cada pixel da imagem, superando a limitação de atuar apenas em alguns pontos.

Assim como o algoritmo anterior, essa técnica parte da equação de restrição de fluxo óptico (1), mas busca encontrar uma solução única em todas as regiões da imagem (incluindo as homogêneas, onde o problema da abertura é máximo). Para viabilizar isso, Horn e Schunck [3] introduzem uma suposição de regularização,

O campo de fluxo óptico varia suavemente em toda a imagem.

Isso implica que vetores de fluxo de pixels vizinhos devem ser similares, permitindo que a informação de movimento de regiões texturizadas, onde o gradiente é alto, se propague para regiões não texturizadas, onde a estimativa seria ambíguo.

O problema, então, é formulado como a minimização de um funcional de energia global E , definido sobre todo o domínio da imagem Ω , que combina dois termos: o erro na constância do brilho (E_d) e o termo de suavidade (E_s). Assim, temos

$$E = \iint_{\Omega} \left(\underbrace{(I_x u + I_y v + I_t)^2}_{E_d} + \alpha^2 \underbrace{(\|\nabla u\|^2 + \|\nabla v\|^2)}_{E_s} \right) dx dy \quad (2)$$

onde E_d penaliza desvios da equação de restrição do fluxo óptico, E_s penaliza variações bruscas nos vetores u e v e α é um fator de regularização que pondera a influência da suavização.

A minimização deste funcional pode ser resolvida através do Cálculo das Variações, resultando em um par de equações de Euler-Lagrange. A solução numérica é tipicamente obtida por um método iterativo (como Jacobi ou Gauss-Seidel), em que a nova estimativa de velocidade para um pixel (u^{n+1}, v^{n+1}) depende da média das velocidades de seus vizinhos (\bar{u}^n, \bar{v}^n) .

Para a visualização dos resultados. utilizaremos a representação Matriz e Saturação descrita anteriormente.

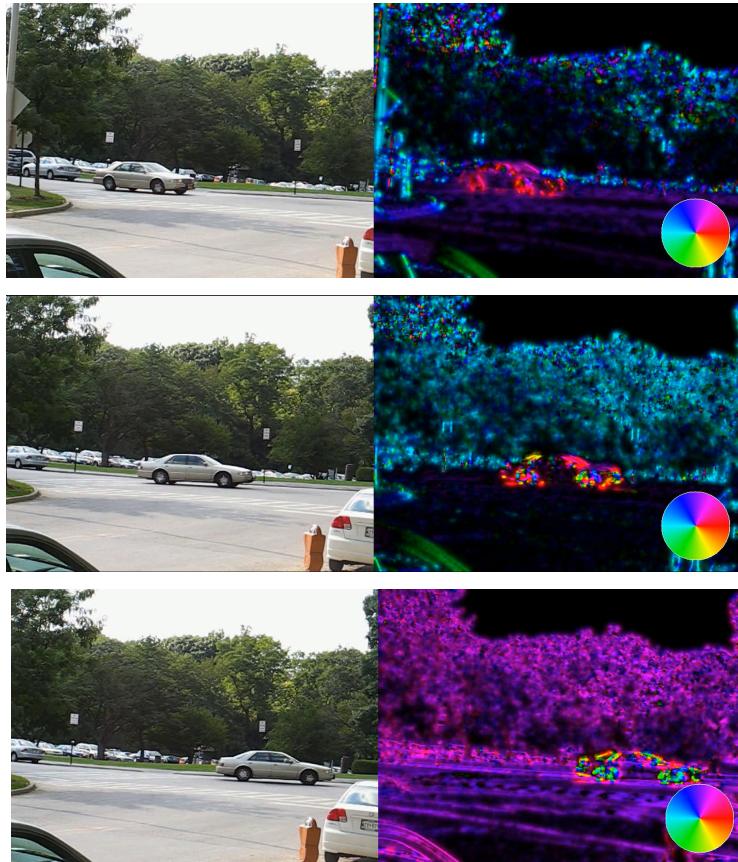


Figura 4: Estimativa de fluxo óptico usando Horn-Schunck aplicado em frames disponíveis no dataset Moseg [6]

A principal vantagem alcançada é calcular o movimento em locais antes impossíveis para o método de Lucas-Kanade, como superfícies homogêneas. Outro ponto relevante é identificar zonas estáticas na imagem. As áreas escuras indicam que a magnitude da velocidade estimada é zero. No exemplo, o céu e o asfalto não se movem.

No entanto, a estimativa produzida ainda possui grandes limitações. A suposição de suavidade global tende a borrar as descontinuidades de movimento, gerando erros nas bordas dos objetos. Por fim, nota-se a presença de certo ruído nas regiões de fundo, com pontos coloridos nas árvores.

3.2.4 O Estado da Arte

Os algoritmos clássicos vistos estabeleceram o alicerce teórico para grande parte das técnicas futuras de fluxo óptico. Hoje, a literatura classifica as abordagens de estimativa de movimento em três categorias principais: *knowledge-*

driven, data-driven e hybrid-driven [8].

Os métodos *knowledge-driven* correspondem aos procedimentos discutidos inicialmente. Eles dependem de premissas explícitas, como a constância de brilho e a suavidade espacial. Apesar de matematicamente robustos, frequentemente falham em cenários complexos com grandes deslocamentos, oclusões ou variações de iluminação não modeladas.

Por isso, a área é dominada pelas estratégias *data-driven*, impulsionados pelo sucesso das Redes Neurais Convolucionais (CNNs) [9,10]. Arquiteturas baseadas em U-Net (para processamento multiescala) e Redes de Pirâmide Espacial (que refinam o fluxo do nível mais “grosso” para o mais “fino”) tornaram-se o padrão de referência, superando os métodos clássicos nas métricas bases e nos mais famosos benchmarks [8].

Entretanto, essa ascensão introduziu um novo desafio: a dependência de dados supervisionados. O treinamento dessas redes exige um conjunto de *ground truth* denso, ou seja, um vetor de movimento correto para cada pixel da imagem. A coleta desses dados em cenas reais é um processo extremamente complexo e inviável manualmente, forçando o uso de datasets sintéticos que nem sempre representam fielmente a realidade [8].

3.2.5 Redes Neurais Convolucionais (CNNs)

Para compreender as abordagens contemporâneas de estimativa de fluxo óptico, é necessário, antes de tudo, examinar a transição para métodos *data-driven*. Nesse contexto, apresentaremos os conceitos básicos sobre as Redes Neurais Convolucionais (CNNs ou *ConvNets*). Diferentemente das redes neurais tradicionais, que não escalam adequadamente para imagens de alta resolução devido ao vasto número de parâmetros necessários, elas são projetadas, primariamente, para processar dados que possuem uma topologia em grade, como imagens ou séries temporais [11], empregando operações de convolução para processar dados de forma local e hierárquica.

A arquitetura das CNNs é biologicamente inspirada nos estudos seminais de Hubel e Wiesel [12] sobre o funcionamento do córtex visual de gatos. Esse conceito foi formalizado e popularizado por LeCun [10] através da arquitetura LeNet-5, utilizada com sucesso no reconhecimento de dígitos manuscritos.

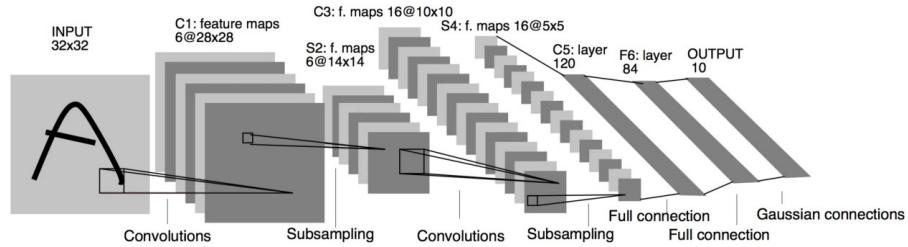


Figura 5: Arquitetura da LeNet-5, disponível em [13]

O funcionamento de uma CNN baseia-se, essencialmente, em três estágios principais [14] que se alternam ao longo da arquitetura da rede:

Camada de Convolução: Em vez de conectar cada neurônio a todos os pixels da imagem, a rede utiliza filtros de tamanho reduzido que deslizam sobre a entrada. Essa operação gera *feature maps*, permitindo que a rede aprenda a detectar padrões visuais — como bordas, texturas e formas — independentemente de sua posição na imagem.

Camada de Pooling: Após a convolução, é comum aplicar uma camada de *pooling*. O objetivo é reduzir progressivamente a dimensionalidade espacial da representação, diminuindo a quantidade de parâmetros e o custo computacional, além de tornar a rede mais robusta a pequenas distorções e translações na imagem de entrada.

Camadas Totalmente Conectadas: Nas etapas finais da arquitetura, os mapas de características são transformados em um vetor unidimensional e processados por camadas tradicionais, como MLPs.

A relevância contemporânea das CNNs foi consolidada no trabalho de [9] com a arquitetura AlexNet, que demonstrou uma superioridade significativa dessa abordagem em competições de reconhecimento de imagem em larga escala (ImageNet), marcando o início da revolução moderna do *Deep Learning*.

3.2.6 DPFlow: Adaptive Optical Flow with Dual-Pyramid

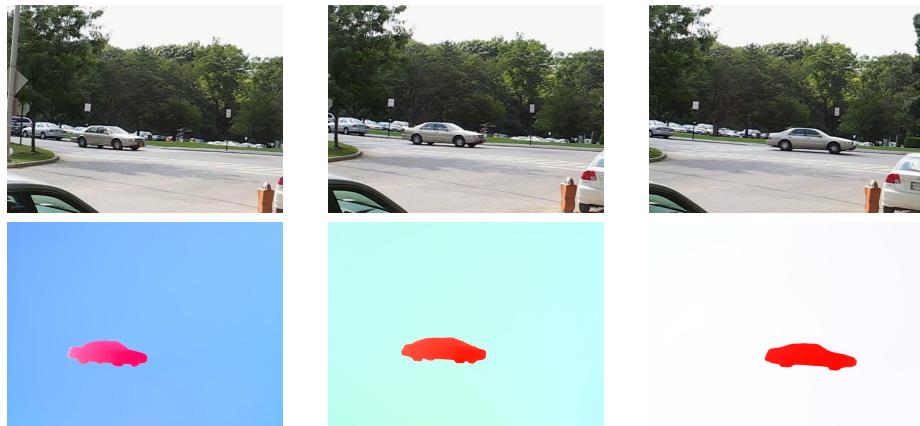
Finalmente, discutiremos um modelo consolidado para a estimativa de fluxo óptico. Também, apresentaremos algumas aplicações interessantes.

Para abordar as limitações de generalização em altas resoluções observadas nos métodos atuais, Morimitsu et al. propuseram o DPFlow (*Dual-Pyramid Flow*) [4]. Trata-se de uma arquitetura baseada em CNNs que adota uma estrutura codificador-decodificador, projetada para estimar fluxo óptico de forma adaptativa em resoluções que variam de 1K até 8K, mesmo quando treinada apenas com amostras de baixa resolução.

Diferentemente de abordagens anteriores que dependem de *tiling* (re-corte de imagem) ou mecanismos de atenção custosos, o DPFlow apoia-se em duas inovações baseadas em operações convolucionais puras: o Codificador de Pirâmide Dupla e a Unidade de Portão Cruzado (*Cross-Gated Unit - CGU*) [4].

A fusão dessas operações permite o trânsito de informações entre níveis profundos e rasos, garantindo a captura simultânea de contexto global e de detalhes locais finos.

As imagens a seguir apresentam os resultados de aplicar o modelo aos mesmos exemplos utilizados anteriormente. Assim, confirma-se a expectativa de aperfeiçoamento do DPFlow em relação aos métodos clássicos. O modelo demonstra capacidade de processar regiões homogêneas e apresenta notável robustez contra ruídos em áreas estáticas, preservando a neutralidade ao fundo. Além disso, a rede corrige as imperfeições nas bordas, onde a suavização global de Horn e Schunck falhava, produzindo estimativas com silhuetas nítidas e bem segmentadas do objeto em movimento.



(a) Estimativa de fluxo óptico produzida pelo DPFlow em frames disponíveis no dataset Moseg [6]



(b) Estimativa de fluxo óptico produzida pelo DPFlow em um vídeo gravado próximo ao IME-USP

Dada a eficácia do DPFlow em lidar com diferentes tipos de ambientes na cena e de gerar estimativas densas e nítidas, futuramente, busca-se integrar essa arquitetura ao ecossistema do projeto SideSeeing para análise de cenários urbanos. Na próxima seção, será descrito com mais detalhes essa ideia.

3.3 Sideseeing Project

O Projeto SideSeeing visa desenvolver métodos para obter e para analisar dados relacionados à acessibilidade urbana [1]. Para isso, é utilizado abordagens de Visão Computacional e Aprendizado de Máquina.

O aplicativo de sensoriamento multimodal, MultiSensor Data Collection, é uma ferramenta central para aquisição, carregamento e análise de dados em nível de rua. O sistema permite a captura de diversos sensores, incluindo acelerômetro, giroscópio, magnetômetro, GPS, câmera de vídeo e áudio. Além disso, para apoiar a análise dos dados coletados, a biblioteca python SideSeeing Tools possui scripts projetados para carregar, pré-processar e plotar os dados, lidando com todos os tipos de conteúdo dentro de cada instância de conjunto de dados [15].

Para expandir as capacidades do framework SideSeeing, esta etapa propõe e implementa um novo sistema focado na automação da visualização dos dados. Essa contribuição visa suprir a necessidade de um módulo que, além de facilitar a geração de relatórios, também é capaz de permitir a interpretação dos dados coletados. A seguir, mostraremos os resultados da primeira versão.

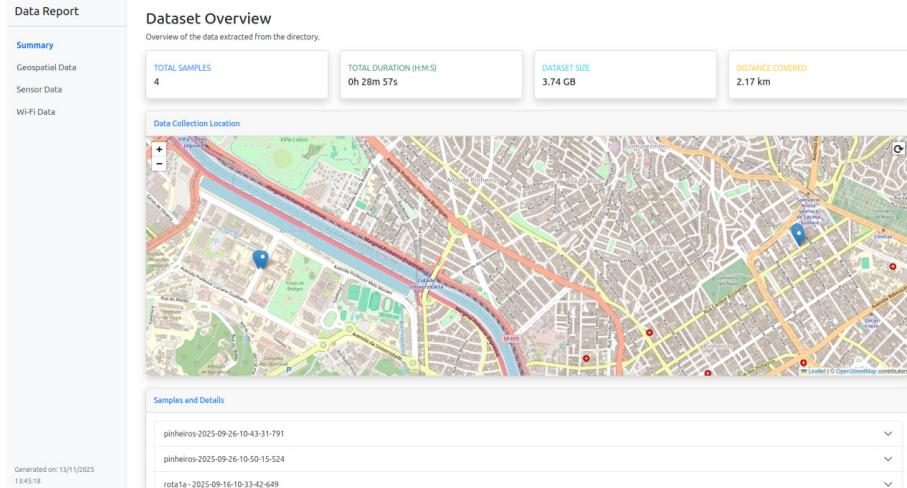


Figura 7: Aba central do automatizador de relatórios, SideSeeing Exporter

O desenvolvimento deste sistema, organizado no novo módulo `export.py`, estabelece um pipeline que processa os objetos `SideSeeingDS`, agrupa seus metadados e exporta dados multimodais (como sensores, GPS e Wi-Fi) para arquivos

JSON. Esses arquivos são carregados sob demanda por um dashboard HTML usando a engine Jinja2.

A interface abaixo permite ao usuário renderizar gráficos de sensores (via Plotly), mapas geoespaciais (via Leaflet) das rotas de captura e heatmaps de sinais 2.4GHz e 5GHz.

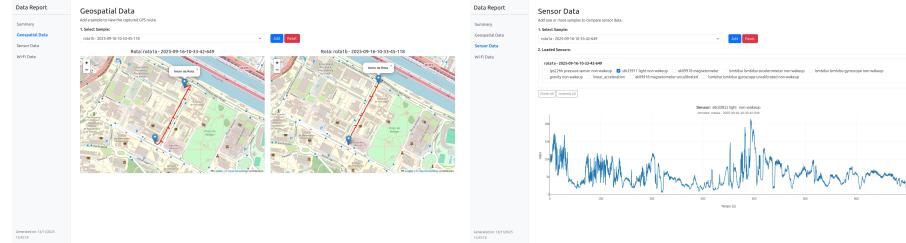


Figura 8: Abas Geoespaciais e de Sensores do SideSeeing Exporter

Embora essa versão piloto atenda aos requisitos propostos, observa-se ainda uma necessidade de refinamento estético e de usabilidade (IHC) para transformá-la em uma ferramenta final. Ainda em desenvolvimento, a próxima versão incorpora um design mais polido, com melhorias na organização dos componentes de interface e na integração entre as diferentes visualizações.

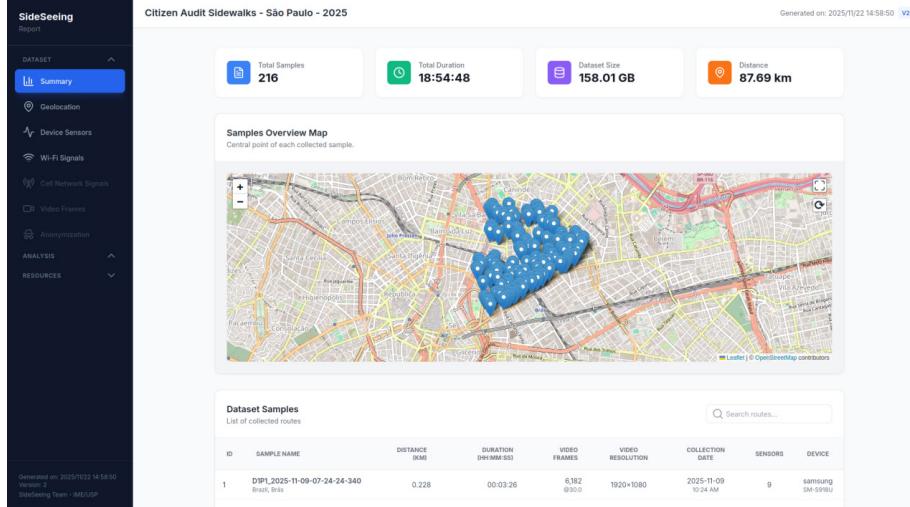


Figura 9: Interface da versão em desenvolvimento, projetada para a futura integração de análises via modelos de *Deep Learning*.

Para os trabalhos futuros, visa-se a expansão do sistema para realizar a análise semântica do cenário urbano. O objetivo é detectar e mapear, de forma autônoma, objetos de interesse — como bueiros, rampas de acessibilidade,

faixas de pedestres e obstáculos temporários — utilizando arquiteturas como YOLO [16] e DPFlow [4]. Além disso, o framework deverá ser capaz de suportar o processamento e a integração das ocorrências.

4 Conclusões

As atividades desenvolvidas nesse período permitiram o cumprimento dos objetivos de fundamentação teórica e prática previstos no plano de trabalho. Ao final, foi possível consolidar um domínio introdutório sobre visão computacional, abrangendo desde técnicas para tratamento de pixels até a compreensão de arquiteturas modernas de Deep Learning aplicadas ao fluxo óptico.

O estudo e as análises realizadas entre métodos clássicos, como Lucas-Kanade e Horn-Schunck, e redes neurais, como o DPFlow, contribuíram para o entendimento do estado da arte e das limitações da área. Além do aspecto teórico, o projeto entregou contribuições relevantes para o sistema SideSeeing. O desenvolvimento do *SideSeeing Exporter* representa um passo importante na automação da visualização e da interpretação dos dados coletados, enquanto, também, serve de alicerce para um sistema futuro de análise semântica urbana.

Para a próxima etapa da pesquisa, prevê-se a integração efetiva dos modelos de fluxo óptico (como o DPFlow) e detecção de objetos (como YOLO) ao pipeline desenvolvido.

Referências

- [1] R. J. P. Damaceno, L. Ferreira, F. Miranda, M. Hosseini, and R. M. Cesar Jr, “Sideseeing: A multimodal dataset and tools for sidewalk assessment,” in *Proceedings of The 4th Annual Workshop on The Future of Urban Accessibility*, 2024. [Online]. Available: <https://accessiblecities.github.io/UrbanAccess2024/>
- [2] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo,” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, Canada, 1981, pp. 674–679, iJCAI’81.
- [3] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981. [Online]. Available: [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2)
- [4] H. Morimitsu, X. Zhu, R. M. Cesar, X. Ji, and X.-C. Yin, “Dpflow: Adaptive optical flow estimation with a dual-pyramid framework,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 17810–17820.
- [5] R. Szeliski, *Computer Vision: Algorithms and Applications*. London, UK: Springer, 2010. [Online]. Available: <http://szeliski.org/Book/>

- [6] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 282–295.
- [7] Monolito Nimbus. (2024) Modelos de cores: Hsv e lab. [Online]. Available: <https://www.monolitonimbus.com.br/modelos-de-cores-hsv-e-lab/>
- [8] M. Zhai, X. Xiang, N. Lv, and X. Kong, “Optical flow and scene flow estimation: A survey,” *Pattern Recognition*, vol. 114, p. 107861, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320321000480>
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [12] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
- [13] G. Alves, “Entendendo redes convolucionais (cnns),” Medium (Neurônio BR), oct 2018, acesso em: 30 nov. 2025. [Online]. Available: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>
- [14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [15] R. Damaceno, L. Ferreira, F. Miranda, M. Hosseini, and R. Cesar Jr, “Sideseeing: A multimodal dataset and collection of tools for sidewalk assessment,” *arXiv preprint arXiv:2407.06464*, 2024.
- [16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.