

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luis Felipe Costa Fernandes de Menezes

**Emprego de visão computacional para
identificação de comportamento suspeito no
varejo**

Uberlândia, Brasil

2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luis Felipe Costa Fernandes de Menezes

**Emprego de visão computacional para identificação de
comportamento suspeito no varejo**

Trabalho de conclusão de curso apresentado
à Faculdade de Engenharia Mecânica da Uni-
versidade Federal de Uberlândia, como parte
dos requisitos exigidos para a obtenção título
de Bacharel em Engenharia Mecatrônica.

Orientador: Prof. Dr. Pedro Augusto Queiroz de Assis

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Mecânica

Bacharelado em Engenharia Mecatrônica

Uberlândia, Brasil

2026

Luis Felipe Costa Fernandes de Menezes

Emprego de visão computacional para identificação de comportamento suspeito no varejo

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Engenharia Mecatrônica.

Trabalho aprovado. Uberlândia, Brasil, 21 de maio de 2026:

Prof. Dr. Pedro Augusto Queiroz de Assis
Orientador

Prof. Dr. Fernando Lourenço de Souza

Dr. Arthur Pereira Borges

Uberlândia, Brasil
2026

Agradecimentos

Gostaria de agradecer à Universidade Federal de Uberlândia pela oportunidade de realizar minha formação em uma instituição de excelência. Agradeço também à Faculdade de Engenharia Mecânica e ao Curso de Engenharia Mecatrônica, pela infraestrutura disponibilizada, pelos recursos oferecidos e pelo corpo docente altamente qualificado, que desempenhou papel essencial em minha trajetória universitária.

Agradeço especialmente ao Prof. Dr. Pedro Augusto Queiroz Assis, não apenas pela paciência e dedicação na orientação deste trabalho, mas também por sua atuação exemplar como professor. Seu profissionalismo e comprometimento com o ensino certamente se tornaram uma grande referência para mim.

Expresso minha profunda gratidão à minha família, por todo o apoio e carinho ao longo dessa jornada. Em especial, aos meus pais, Enedina e Edilson, que sempre foram e continuam sendo minha base para tudo. Essa trajetória definitivamente não foi fácil, mas só foi possível graças aos esforços, à confiança e ao amor de vocês.

Sou grato também a todos os meus amigos, que tornaram essa caminhada mais leve. Cada conversa, momento de descontração e incentivo foi extremamente importante. Em especial, agradeço a Jorge, Nicholas e Sofia, que estiveram presentes de forma marcante ao longo dessa etapa.

Por fim, deixo meu agradecimento à EDROM e a todos os colegas da equipe. Foi por meio dessa experiência que passei a me identificar de forma mais concreta com o curso e com a carreira que desejo seguir, além de encontrar um ótimo ambiente de aprendizado.

A todos que, de alguma forma, contribuíram para esta conquista, deixo meu mais sincero muito obrigado!

Resumo

O presente trabalho aborda o problema crônico das perdas de inventário por furtos no setor varejista, com o objetivo de comparar empiricamente o desempenho de duas conhecidas arquiteturas de aprendizado profundo na identificação automatizada de comportamentos suspeitos em vídeos: uma Rede Neural Convolucional Tridimensional (I3D) e um Transformador de Visão (TimeSFormer). Para a condução do estudo, a metodologia englobou a construção de um banco de dados próprio através da agregação, filtragem e segmentação de amostras de bases públicas, submetendo os modelos a dezesseis configurações experimentais. Tais experimentos variaram a estratégia de ajuste de pesos, a resolução temporal e o uso de imagens de fluxo óptico combinadas ao padrão de cores convencional. A avaliação incluiu ainda uma simulação de monitoramento contínuo em vídeos reais não editados, utilizando uma abordagem de inferência por janela deslizante. Os resultados demonstram que o ajuste profundo de todos os parâmetros da rede configura uma etapa estritamente essencial para a convergência e o aprendizado em ambas as arquiteturas. O Transformador de Visão alcançou a melhor capacidade de generalização e controle de alarmes falsos no conjunto de teste, mas exige um investimento computacional substancialmente maior com elevado tempo de inferência. Em contrapartida, a Rede Neural Convolucional apresentou um comportamento preditivo conservador e com maior tendência a falsos positivos, destacando-se, no entanto, pela alta eficiência e baixa latência em sua execução. Conclui-se que a aplicação mercadológica exige um compromisso técnico direto entre a alta seletividade de modelos baseados em atenção e a escalabilidade e rapidez das redes convolucionais, além de constatar que ambos os sistemas compartilham vulnerabilidades críticas e falhas de detecção frente a oclusões severas e movimentos sutis no ambiente comercial.

Abstract

This work addresses the chronic problem of inventory losses due to theft in the retail sector, aiming to empirically compare the performance of two known deep learning architectures for the automated detection of suspicious behavior in videos: a Three-Dimensional Convolutional Neural Network (I3D) and a Vision Transformer (TimeSFormer). To conduct the study, the methodology involved building a proprietary dataset through the aggregation, filtering, and segmentation of samples from public datasets, subjecting the models to sixteen experimental configurations. These experiments varied the weight fine-tuning strategy, temporal resolution, and the use of optical flow images combined with standard RGB inputs. The evaluation also included a simulation of continuous monitoring on real, unedited videos using a sliding window inference approach. The results demonstrate that full fine-tuning of all network parameters is a strictly essential step for convergence and learning in both architectures. The Vision Transformer achieved the best generalization capability and false alarm control on the test set, attaining perfect precision; however, it requires substantially higher computational investment and exhibits high inference time. In contrast, the Convolutional Neural Network showed a more conservative predictive behavior with a greater tendency toward false positives, while standing out for its high efficiency and low latency. It is concluded that practical deployment requires a direct technical trade-off between the high selectivity of attention-based models and the scalability and speed of convolutional networks. Additionally, both systems share critical vulnerabilities and detection failures when faced with severe occlusions and subtle movements in retail environments.

Lista de ilustrações

Figura 1 – Relação hierárquica entre Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo.	18
Figura 2 – Arquitetura típica de uma RNC e a extração progressiva de mapas de características.	20
Figura 3 – Processo de divisão de uma imagem em <i>patches</i> e adição de <i>embeddings</i> posicionais na arquitetura <i>Vision Transformer</i>	21
Figura 4 – Comparação entre a operação de convolução padrão (2D) e a convolução espaço-temporal (3D) aplicada a volumes de vídeo.	23
Figura 5 – Visualização da fatoração do mecanismo de atenção em componentes temporais e espaciais na arquitetura TimeSFormer.	25
Figura 6 – Fluxo conceitual do processo de <i>Transfer Learning</i> e <i>fine-tuning</i> adaptando pesos de um domínio de origem para uma tarefa específica. . . .	26
Figura 7 – Exemplo de Curvas ROC, a Área Sob a Curva (AUC) é a métrica empregada para avaliar cenários de classificação desbalanceados. . . .	28
Figura 8 – Exemplo de amostra da classe “ <i>Shoptlifting</i> ” do banco de dados MNNIT. .	30
Figura 9 – Exemplo de amostra da classe “ <i>Shoptlifting</i> ” do banco de dados Shoptlifting Dataset 2.0.	31
Figura 10 – Exemplos de ambientes do banco de dados DCSASS.	32
Figura 11 – Exemplo de transformação de sequência de quadros original (acima) após a aplicação das técnicas adotadas para aumento de banco de dados (abaixo).	34
Figura 12 – Curvas de perda de treinamento para os experimentos do I3D (a) e TimeSFormer (b) nas modalidades “Completo” e “Cabeça”.	41
Figura 13 – Evolução das métricas de validação para o I3D.	42
Figura 14 – Evolução das métricas de validação para o TimeSFormer.	43
Figura 15 – Comparação geral de AUC-ROC entre todas as 16 configurações experimentais.	44
Figura 16 – Matrizes de confusão do melhor I3D e do melhor TimeSFormer no conjunto de teste.	45
Figura 17 – Exemplo de aplicação da abordagem de janela deslizante sobre vídeo real.	49
Figura 18 – Curva de probabilidade do modelo TimeSFormer para o vídeo <i>furto-supermercado</i>	51
Figura 19 – Curva de probabilidade do modelo TimeSFormer para o vídeo <i>furto-tenis1</i>	52

Figura 20 – Curva de probabilidade do modelo TimeSFormer para o vídeo <i>furto-tenis2</i>	53
Figura 21 – Curva de probabilidade do modelo I3D para o vídeo <i>furto-celular</i>	54
Figura 22 – Curva de probabilidade do modelo I3D para o vídeo <i>furto-pc</i>	54
Figura 23 – Curva de probabilidade dos modelo I3D e TimeSFormer para o vídeo <i>furto-roupa</i>	55
Figura 24 – Curva de probabilidade dos modelo I3D e TimeSFormer para o vídeo <i>furto-shopping</i>	55
Figura 25 – Curva de probabilidade do modelo I3D para o vídeo <i>furto-tenis1</i>	56
Figura 26 – Curva de probabilidade do modelo I3D para o vídeo <i>furto-supermercado</i>	56
Figura 27 – Curva de probabilidade do modelo TimeSFormer para o vídeo <i>furto-celular</i>	57
Figura 28 – Curva de probabilidade do modelo TimeSFormer para o vídeo <i>furto-pc</i>	57

Lista de tabelas

Tabela 1	–	Variações de experimentos realizados. Os símbolos “✓” indicam os experimentos conduzidos, enquanto “–” indica configurações que não foram treinadas.	36
Tabela 2	–	Parâmetros comuns de treinamento.	37
Tabela 3	–	Número total e treinável de parâmetros.	39
Tabela 4	–	Métricas no conjunto de teste dos melhores modelos (Precisão, Revocação e F2-Score).	44
Tabela 5	–	<i>Throughput</i> de inferência dos melhores modelos I3D e TimeSFormer. .	45
Tabela 6	–	Características dos vídeos reais de furto utilizados na avaliação. . . .	49

Lista de abreviaturas e siglas

CFTV	Circuito Fechado de Televisão
IA	Inteligência Artificial
RNC	Rede Neural Convolucional
GPU	<i>Graphics Processing Unit</i>
ViT	<i>Vision Transformer</i>
ML	<i>Machine Learning</i>
DL	<i>Deep Learning</i>
SL	<i>Supervised Learning</i>
RGB	<i>Red, Green, Blue</i>
I3D	<i>Inflated 3D ConvNet</i>
K400	<i>Kinetics-400</i>
SSv2	<i>Something-Something V2</i>
BCE	<i>Binary Cross-Entropy</i>
CEL	<i>Cross-Entropy Loss</i>
AUC-ROC	<i>Area Under the Receiver Operating Characteristic Curve</i>
FPS	<i>Frames Per Second</i>
Adam	<i>Adaptive Moment Estimation</i>
AdamW	<i>Adam with Decoupled Weight Decay</i>

Sumário

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo	17
2.2	Aprendizado Supervisionado e Classificação Binária	18
2.3	Análise de Imagens	19
2.3.1	Redes Neurais Convolucionais	19
2.3.2	Transformadores de Visão	20
2.4	Análise de Vídeos	21
2.5	Arquiteturas para Análise de Vídeos	22
2.5.1	<i>Inflated 3D ConvNet</i>	23
2.5.2	TimeSFormer	24
2.6	<i>Transfer Learning</i> e <i>fine-tuning</i>	24
2.7	Métricas e Função de Perda	26
3	DESCRIÇÃO DO BANCO DE DADOS	29
3.1	Construção do conjunto de dados	29
3.2	Representação e Pré-processamento	33
3.3	Aumento de Banco de Dados	33
3.4	Divisão do conjunto de dados	34
4	RESULTADOS DO TREINAMENTO	36
4.1	Configuração dos Experimentos	36
4.2	Preparação do Modelo I3D	37
4.3	Preparação do Modelo TimeSFormer	37
4.3.1	Interpolação dos <i>Time Embeddings</i>	38
4.3.2	Taxa de Aprendizado e Ausência de <i>Batch Normalization</i>	38
4.4	Comparação da Dimensão dos Treinamentos	39
4.5	Dinâmica de Treinamento e Convergência	39
4.6	Evolução das Métricas na Validação	40
4.7	Avaliação Global e Seleção de Modelos	42
4.8	Desempenho no Conjunto de Teste	42
4.9	Eficiência	45
4.10	Síntese dos Resultados	46

5	AVALIAÇÃO EM VÍDEOS REAIS	48
5.1	Metodologia de Inferência	48
5.2	Conjunto de Vídeos Reais	49
5.3	Resultados	49
5.3.1	Casos de Sucesso	50
5.3.2	Casos de Falha	50
5.4	Síntese dos Resultados	52
6	AVALIAÇÃO DE IMPLEMENTAÇÃO EM CENÁRIO REAL	58
6.1	Panorama de Perdas no Varejo Alimentar Brasileiro	58
6.2	Dimensionamento da Infraestrutura de Captação	59
6.3	Dimensionamento da Demanda de Inferência e Custos	60
6.4	Estimativa de Redução de Perdas e Retorno Sobre o Investimento	62
7	CONCLUSÕES	64
	REFERÊNCIAS	66
	APÊNDICES	70
	APÊNDICE A – REPOSITÓRIOS DO PROJETO	71

1 Introdução

O setor varejista global enfrenta um desafio crônico que diminui suas margens de lucro e afeta a sustentabilidade de suas operações: as perdas de inventário por furto. O alto fluxo de indivíduos em ambientes comerciais de livre acesso cria um ecossistema suscetível a esse tipo de conduta. Em escala global, a National Retail Federation reportou um aumento de 93% nos incidentes de furtos em 2024 em relação aos níveis de 2019, resultando em perdas financeiras anuais que ultrapassam dezenas de bilhões de dólares e forçando as empresas a repensarem ativamente suas estratégias de segurança (National Retail Federation; Loss Prevention Research Council, 2024). No contexto nacional, ações ilícitas geram impactos de tal magnitude nas receitas das empresas do setor que em 2023 foi observado um aumento de 27%, em relação ao ano anterior, no número de empresas que possuem áreas dedicadas à prevenção de perdas, totalizando 95% de todas as empresas analisadas (Associação Brasileira de Prevenção de Perdas; KPMG, 2024).

As áreas dedicadas à prevenção de perdas atuam no monitoramento e mitigação de diferentes fontes de prejuízo, incluindo quebras operacionais, furtos externos, furtos internos, erros de inventário e fraudes. Em particular, os furtos configuram-se como a segunda maior causa de perdas no varejo, correspondendo a aproximadamente 31% dos R\$34,9 bilhões em prejuízos registrados no ano analisado (Associação Brasileira de Prevenção de Perdas; KPMG, 2024), justificando investimentos maciços em segurança. Nesse sentido, agentes de vigilância constituem um componente essencial na proteção do setor varejista, sendo que uma parcela significativa dos investimentos em prevenção de perdas é direcionada a esses funcionários (GROSSBERGER, 2021).

Entretanto, métodos tradicionais de vigilância e proteção de ativos têm se mostrado insuficientes. A dependência quase exclusiva de Circuitos Fechados de Televisão (CFTV) operados de forma manual exige que os agentes de monitoramento mantenham atenção constante e ininterrupta a dezenas de monitores simultaneamente, o que pode resultar em cegueira por desatenção (SIMONS; CHABRIS, 1999). Estudos empíricos conduzidos especificamente em ambientes de monitoramento demonstram que operadores humanos, independentemente de possuírem treinamento prévio para tarefa, chegam a falhar na detecção de até 66% dos eventos críticos que ocorrem claramente em seus campos de visão (NÄSHOLM; ROHLFING; SAUER, 2014). Na prática operacional, essa limitação inerente ao fator humano se traduz em taxas significativamente reduzidas de detecção de comportamentos ilícitos em tempo real. É exatamente neste ponto que a visão computacional e o aprendizado de máquina assumem o protagonismo, oferecendo soluções automatizadas capazes de monitorar o espaço físico, reconhecer padrões complexos com precisão e sinalizar comportamentos anômalos antes que o suposto infrator abandone o

estabelecimento.

Sob a perspectiva da inovação, transformar a infraestrutura legada e passiva de CFTV em uma rede de sensores ativos, impulsionados por Inteligência Artificial (IA), pode representar um salto imenso na eficiência do monitoramento. Contudo, a transição da simples aquisição de imagens bidimensionais para a compreensão semântica profunda do comportamento de indivíduos por meio da análise do vídeo configura um problema extremamente complexo na área de visão computacional. Ao contrário do reconhecimento clássico de imagens estáticas, que depende da extração de características espaciais puras, como formas geométricas e detecção de bordas, a análise comportamental em vídeo exige a modelagem indispensável do espectro temporal. Abordagens puramente espaciais mostram-se insuficientes nesse contexto, sobretudo na distinção de ações querais¹, uma vez que não capturam a evolução temporal das interações na cena (HUANG et al., 2018). Consequentemente, comportamentos suspeitos de furto no varejo não podem ser adequadamente caracterizados a partir da análise isolada de quadros individuais.

A literatura voltada para a análise de vídeos, especialmente na detecção automatizada de anomalias, tem sido objeto de uma vigorosa e incessante exploração ao longo das últimas décadas. Com a disponibilidade de um poder computacional cada vez maior utilizando múltiplas GPUs, as metodologias de análise de vídeo transitaram de abordagens baseadas em heurísticas manuais para arquiteturas de aprendizado profundo ricas em representação semântica. Em particular, o problema da detecção de furtos tem sido solucionado por meio de diferentes métodos. Abordagens clássicas utilizam subtração de fundo para segmentar objetos em movimento do fundo estático da cena, seguidos de operações morfológicas para extrair *bounding boxes* e analisar variações espaciais ao longo do tempo (RAHANGDALE; KOKATE, 2016; CORDEIRO et al., 2020). Entretanto, essas abordagens mostraram baixa robustez em ambientes reais devido à sensibilidade a mudanças de iluminação, oclusões e à incapacidade de capturar o contexto semântico.

Já os métodos baseados em IA/aprendizado profundo passaram a empregar, dentre outras metodologias, Redes Neurais Convolucionais 3D (RNCs 3D), capazes de modelar simultaneamente informações espaciais e temporais. Nesse tipo de modelo de IA, se aprende a atribuir maiores pontuações de anomalia aos trechos mais suspeitos mesmo sem anotações precisas do instante em que o evento ocorre (SULTANI; CHEN; SHAH, 2018). No entanto, o treinamento inicial de RNCs 3D é extremamente custoso, exigindo a criação de bancos de dados relativamente extensos, como o UCF-Crime (SULTANI; CHEN; SHAH, 2018), que consiste em longos vídeos de vigilância não editados que abrangem 13 tipos de ocorrências do mundo real, incluindo furto em lojas, com mais de 128 horas.

A fim de contornar essa explosão de parâmetros e dados exigida pelas RNCs 3D, surgiram as Redes Neurais Híbridas, nas quais o processo analítico de aprendizado é de-

¹ Movimentos temporalmente opostos, como abrir versus fechar uma porta.

composto em duas etapas sequenciais e distintas: redes convolucionais bidimensionais tradicionais pré-treinadas para extração de características espaciais dos quadros combinadas com redes recorrentes responsáveis por modelar as dependências temporais das sequências de vídeo (MUNEER et al., 2023). Embora apresentem bom equilíbrio entre desempenho e custo computacional, esses métodos podem perder informações espaciais detalhadas, pois a dinâmica temporal é analisada apenas após a compressão das características visuais extraídas pela RNC.

Paralelamente, surgiram abordagens orientadas à preservação de privacidade, nas quais algoritmos de estimativa de pose extraem apenas os pontos-chave do corpo humano, convertendo os movimentos dos indivíduos em representações esqueléticas anonimizadas. A dinâmica dessas articulações é então modelada por Redes Convolucionais de Grafos Espaço-Temporais (RASHVAND et al., 2025), capazes de analisar padrões biomecânicos de movimento para identificar comportamentos suspeitos. Como desvantagem dessa abordagem, tem-se uma redução da eficácia na presença de oclusões, o que frequentemente ocorre em ambientes de varejo.

Por fim, a vertente que tem dominado as pesquisas contemporâneas explora a maturação e evolução dos Transformadores de Visão (*Vision Transformers*), que adapta a tecnologia originalmente desenvolvida para decodificar linguagem natural para o escopo temporal do vídeo, nos quais sequências de quadros são divididas em *patches* visuais e processadas por mecanismos de auto-atenção capazes de modelar relações espaciais e temporais de longo alcance (BERTASIUS; WANG; TORRESANI, 2021). Estas arquiteturas utilizam atenção fatorada no espaço e no tempo para capturar dependências complexas entre regiões da cena ao longo da sequência, alcançando desempenho competitivo em *benchmarks* de reconhecimento de ações, o que se dá devido a ajustes arquiteturais finos e a um maior treinamento, embora ainda dependam de grandes volumes de dados para treinamento eficaz (PRABHAKAR; SHIVASHANKAR; MOHANA, 2024).

A literatura apresentada evidencia a importância do monitoramento automático do ambiente de varejo e que ainda não existe uma solução única e definitiva para o problema da detecção automática de comportamentos suspeitos. Diferentes abordagens apresentam vantagens e limitações relacionadas à robustez, custo computacional e de treinamento e capacidade de modelar adequadamente as dinâmicas espaço-temporais presentes em vídeos de vigilância. Dessa forma, neste trabalho comparar-se-ão os desempenhos de arquiteturas baseadas em RNCs 3D e Transformadores de Visão na tarefa de detecção de possíveis furtos em vídeos de vigilância, verificando qual apresenta maior eficácia, sensibilidade e viabilidade técnica para a tarefa, no banco de dados utilizado.

Para a condução das comparações, estruturou-se um conjunto de dados próprio a partir de uma agregação de bases de vídeos reais e sintéticos publicamente disponíveis em repositórios online, englobando as coleções DCSASS, Shoplifting Dataset 2.0 e Shoplifting

- MNIT (HERVAS, 2019; HISHAM, 2024; ANSARI; SINGH, 2023). Como medida de precaução ética e legal, os rostos presentes em vídeos foram artificialmente distorcidos, impossibilitando o reconhecimento e assegurando a total preservação da privacidade dos indivíduos.

Além disso, adaptaram-se ambos os modelos por meio de *transfer learning*, com a substituição da camada final de classificação por uma cabeça binária treinada para distinguir entre as classes “*Shoptlifting*” (possível furto) e “Normal” em vídeos reais de vigilância varejista. Os modelos são treinados sobre o conjunto de dados específico da tarefa, considerando diferentes estratégias de ajuste dos parâmetros, incluindo o treinamento apenas da camada classificadora e o *fine-tuning* completo da rede. Para então, finalmente, serem avaliados em um conjunto de teste em comum. A comparação é feita com base no *F2-Score*, uma variação da medida F que atribui um peso maior à revocação em comparação à precisão (POWERS, 2011), pois, no cenário varejista, falsos negativos são mais críticos que falsos positivos, considerando um avaliador intermediário humano.

O presente trabalho divide-se em sete capítulos, além desta introdução. Na Fundamentação Teórica, apresentam-se os conceitos de Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo, avançando para classificação binária e modelagem espaço-temporal em imagens e vídeos. Em seguida, são descritas as arquiteturas I3D e TimeSFormer, bem como as técnicas de treinamento, funções de perda e métricas de avaliação.

No Banco de Dados, descreve-se o processo de construção do conjunto de dados utilizado neste trabalho, incluindo a agregação de diferentes bases públicas, as etapas de curadoria, filtragem e segmentação dos vídeos, bem como a caracterização final do conjunto em termos de distribuição de classes e duração. Adicionalmente, são apresentados os procedimentos de pré-processamento, as técnicas de aumento de dados empregadas e a estratégia de divisão em subconjuntos de treinamento, validação e teste, visando garantir a representatividade e a confiabilidade das avaliações realizadas.

Nos Resultados do Treinamento, são apresentados e discutidos os desempenhos de 16 configurações experimentais de *fine-tuning* para as arquiteturas I3D e TimeSFormer. O capítulo demonstra empiricamente que o ajuste profundo de todos os pesos da rede é estritamente essencial para a convergência do aprendizado nesta tarefa. A análise compara o comportamento dos modelos nos conjuntos de validação e teste, evidenciando um claro compromisso entre custo computacional e capacidade de generalização. Enquanto o modelo I3D destaca-se pelo baixo tempo de inferência, o TimeSFormer alcança o melhor desempenho preditivo global, obtendo 100% de precisão no conjunto de teste e demonstrando controle superior na redução de falsos positivos em cenários inéditos.

Na Avaliação em Vídeos Reais, apresenta-se a aplicação dos melhores modelos de cada arquitetura em um cenário prático de monitoramento contínuo de fluxo de vídeo,

utilizando uma abordagem de inferência por janela deslizante. O desempenho prático é avaliado sobre um conjunto inédito de sete vídeos reais de ocorrências variadas no varejo. A análise revela que o TimeSFormer demonstra maior seletividade e capacidade de sincronização com o exato momento da ocultação, embora sofra com latência em alguns casos. Em contrapartida, o modelo I3D destaca-se na detecção de ações abruptas, mas apresenta uma forte tendência à saturação de probabilidade e falsos positivos gerados pela manipulação legítima de produtos. Por fim, evidencia-se que ambas as arquiteturas compartilham uma limitação crítica: a falha na detecção diante de oclusões severas e movimentos corporais sutis.

Na Avaliação de Implementação em Cenário Real, apresenta-se um estudo de viabilidade técnica e financeira para a implantação do sistema proposto em um ambiente varejista padrão. O capítulo detalha o dimensionamento da infraestrutura de captação de imagens e a modelagem da demanda computacional para inferência contínua, transpondo o custo de processamento do modelo TimeSFormer para requisitos reais de *hardware* (GPUs). Por fim, realiza-se uma projeção econômica que contrapõe os custos estimados de aquisição dos equipamentos à perspectiva de redução de perdas financeiras geradas por furtos, demonstrando o potencial de retorno sobre o investimento e a viabilidade prática da solução.

Por fim, em Conclusões, sintetizam-se os achados empíricos da pesquisa, destacando que o ajuste profundo de todos os pesos da rede foi estritamente essencial para a convergência e o aprendizado eficaz das dinâmicas de furto em ambas as arquiteturas. Confirma-se a superioridade do modelo TimeSFormer na generalização para dados inéditos, alcançando 100% de precisão e controle superior na redução de alarmes falsos, em contraste com o comportamento conservador e a tendência a falsos positivos do modelo convolucional I3D. Discute-se também o compromisso técnico inerente à aplicação mercadológica, evidenciando que a maior precisão do TimeSFormer exige um investimento computacional substancialmente maior, ao passo que o I3D se destaca pela alta eficiência e baixa latência. O capítulo encerra o trabalho apontando as vulnerabilidades compartilhadas por ambos os modelos frente a oclusões severas e movimentos corporais sutis, além de propor diretrizes para trabalhos futuros, tais como a ampliação do banco de dados com filmagens locais do varejo e a integração com abordagens multimodais baseadas em estimativa de pose esquelética.

2 Fundamentação Teórica

2.1 Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo

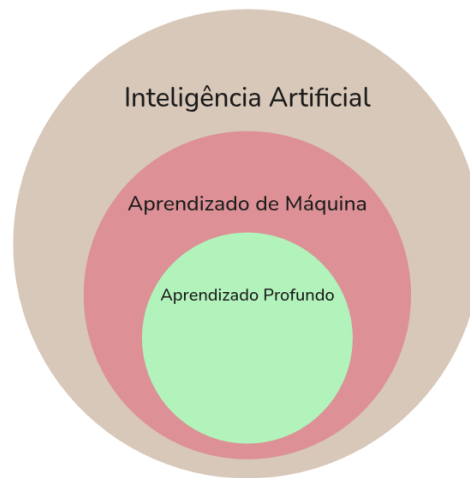
A análise automatizada de dados visuais, como imagens e vídeos, é o objeto de estudo da visão computacional: campo interdisciplinar que busca dotar os computadores da capacidade de extrair, processar e compreender informações de alto nível a partir do mundo visual (SZELISKI, 2022). Atualmente, o desenvolvimento desses sistemas fundamenta-se nos avanços recentes da IA, área da computação dedicada ao estudo e projeto agentes racionais, ou seja, sistemas capazes de perceber o ambiente, interpretar informações e tomar decisões orientadas a maximizar as chances de atingir objetivos específicos (RUSSELL; NORVIG, 2010). Historicamente, abordagens de IA baseavam-se em sistemas simbólicos e regras explicitamente programadas. Contudo, tais métodos mostram-se inadequados para problemas complexos e de alta variabilidade, como a identificação de comportamentos humanos em ambientes reais.

Nesse contexto, o Aprendizado de Máquina (*Machine Learning* – ML) consolidou-se como o paradigma dominante, permitindo que modelos computacionais aprendam padrões diretamente a partir de dados. De acordo com a definição clássica, um programa aprende a partir da experiência E , em relação a uma tarefa T e uma medida de desempenho P , se seu desempenho em T , avaliado por P , melhora com a experiência acumulada (MITCHELL, 1997). No escopo deste trabalho, a tarefa corresponde à classificação de sequências de vídeo, a experiência é representada por conjuntos de dados contendo exemplos de comportamentos normais e de furto, e a medida de desempenho refere-se às métricas utilizadas para avaliar a qualidade das predições do modelo.

Como ilustrado na Figura 1, dentro do ML destaca-se o Aprendizado Profundo (*Deep Learning* – DL), que utiliza redes neurais artificiais com múltiplas camadas para aprender representações hierárquicas dos dados. Diferentemente de abordagens tradicionais, que dependem da extração manual de características, modelos de DL realizam esse processo de forma automática, aprendendo desde padrões de baixo nível até representações semânticas mais abstratas (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa capacidade torna o DL particularmente adequado para o processamento de dados de alta dimensionalidade, como imagens e vídeos, permitindo a modelagem de padrões complexos associados ao comportamento humano.

Dessa forma, o DL estabelece a base teórica para o desenvolvimento de sistemas de visão computacional capazes de analisar sequências de vídeo de forma automática, sendo

Figura 1 – Relação hierárquica entre Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo.



Fonte - Elaborado pelo autor.

a base metodológica deste trabalho.

2.2 Aprendizado Supervisionado e Classificação Binária

O treinamento de modelos de DL para tarefas de visão computacional é, em grande parte, realizado sob o paradigma de Aprendizado Supervisionado (*Supervised Learning* – SL). Nessa abordagem o modelo é treinado a partir de um conjunto de dados rotulado, no qual cada instância de entrada x está associada a um rótulo de saída conhecido y , frequentemente denominado *ground truth*. O objetivo do processo de aprendizado é estimar uma função de mapeamento $f: X \rightarrow Y$ capaz de generalizar adequadamente para dados não observados, minimizando o erro entre as predições do modelo e os rótulos reais (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012).

Dentre as diversas tarefas que podem ser realizadas com SL, a classificação destaca-se como uma das mais recorrentes. Na classificação o objetivo é atribuir cada entrada a uma dentre um conjunto finito de classes discretas (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). No contexto deste trabalho, a tarefa abordada consiste em um problema de classificação binária, no qual o espaço de saída é composto por duas classes mutuamente exclusivas: “*Shoptlifting*” e “Normal”. Formalmente, a classificação binária pode ser modelada como a predição de uma variável aleatória $y \in \{0, 1\}$, sendo que, por convenção, a classe positiva ($y = 1$) representa o evento de interesse e a classe negativa ($y = 0$) representa eventos normais. A saída do modelo é tipicamente interpretada como uma probabilidade associada à classe positiva, permitindo a definição de um limiar de decisão para a classificação final (CHEN, 2021).

Um aspecto fundamental do tipo de problema abordado no presente trabalho é o desbalanceamento de classes: eventos de interesse, como furtos, ocorrem com frequência significativamente menor em comparação a comportamentos normais. Como consequência, um modelo treinado de forma ingênua pode apresentar viés em direção à classe majoritária (i.e. “Normal”), resultando em previsões aparentemente precisas em termos de acurácia, mas pouco eficazes na detecção do evento raro (CASTRO; BRAGA, 2011). Desta forma, torna-se necessário adotar estratégias que mitiguem os efeitos do desbalanceamento, incluindo o uso de funções de perda apropriadas e métricas de avaliação mais informativas, capazes de refletir o desempenho do modelo em ambas as classes (GRANDINI; BAGLI; VISANI, 2020). Esses aspectos são fundamentais para a correta avaliação de sistemas de detecção de eventos raros e serão discutidos em maior detalhe nas seções subsequentes.

2.3 Análise de Imagens

No contexto do DL, imagens são representadas como tensores multidimensionais, nos quais cada elemento corresponde à intensidade de um pixel em determinado canal de cor, tipicamente no espaço RGB. A partir dessa representação estruturada, modelos computacionais são capazes de extrair padrões relevantes por meio de transformações sucessivas aplicadas aos dados de entrada (SZELISKI, 2022).

Dentre as abordagens desenvolvidas para tarefas de classificação, destacam-se duas famílias principais de modelos: as RNCs e os modelos baseados em Transformadores de visão. Essas abordagens diferem fundamentalmente na forma como capturam dependências espaciais e semânticas nas imagens, constituindo paradigmas complementares na área de visão computacional.

2.3.1 Redes Neurais Convolucionais

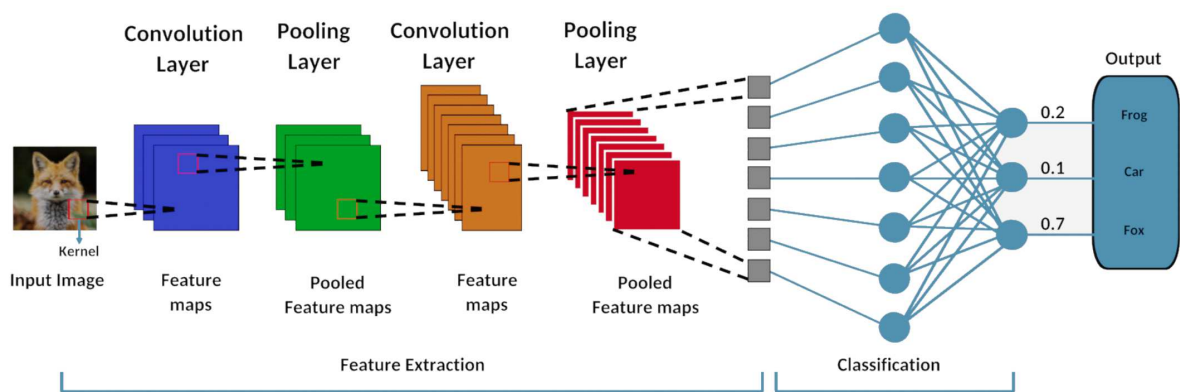
As RNCs foram projetadas para explorar a estrutura espacial das imagens, incorporando propriedades como localidade e invariância à translação. Diferentemente de redes totalmente conectadas, nas quais cada neurônio se conecta a todos os elementos da entrada, as RNCs utilizam filtros convolucionais que operam localmente sobre regiões da imagem, reduzindo significativamente o número de parâmetros e preservando a organização espacial dos dados (SZELISKI, 2022).

A operação central dessas redes é a convolução, na qual um conjunto de filtros aprendíveis é aplicado de forma deslizando sobre a imagem, gerando mapas de características. Esses mapas capturam padrões locais, como bordas, texturas e formas, que são progressivamente combinados em representações mais abstratas ao longo das camadas mais profundas da rede. Como resultado, as RNCs aprendem uma hierarquia de carac-

terísticas, partindo de informações de baixo nível até representações semânticas de alto nível (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Figura 2 ilustra visualmente essa dinâmica, na qual a imagem de entrada é submetida a sucessivas camadas de convolução e *pooling*, responsáveis por extrair os mapas de características (*features*) de forma hierárquica. Ao término dessa fase de extração, essas informações espaciais são unidas e repassadas para a etapa de classificação, onde a rede consolida os dados para prever a saída final.

Figura 2 – Arquitetura típica de uma RNC e a extração progressiva de mapas de características.



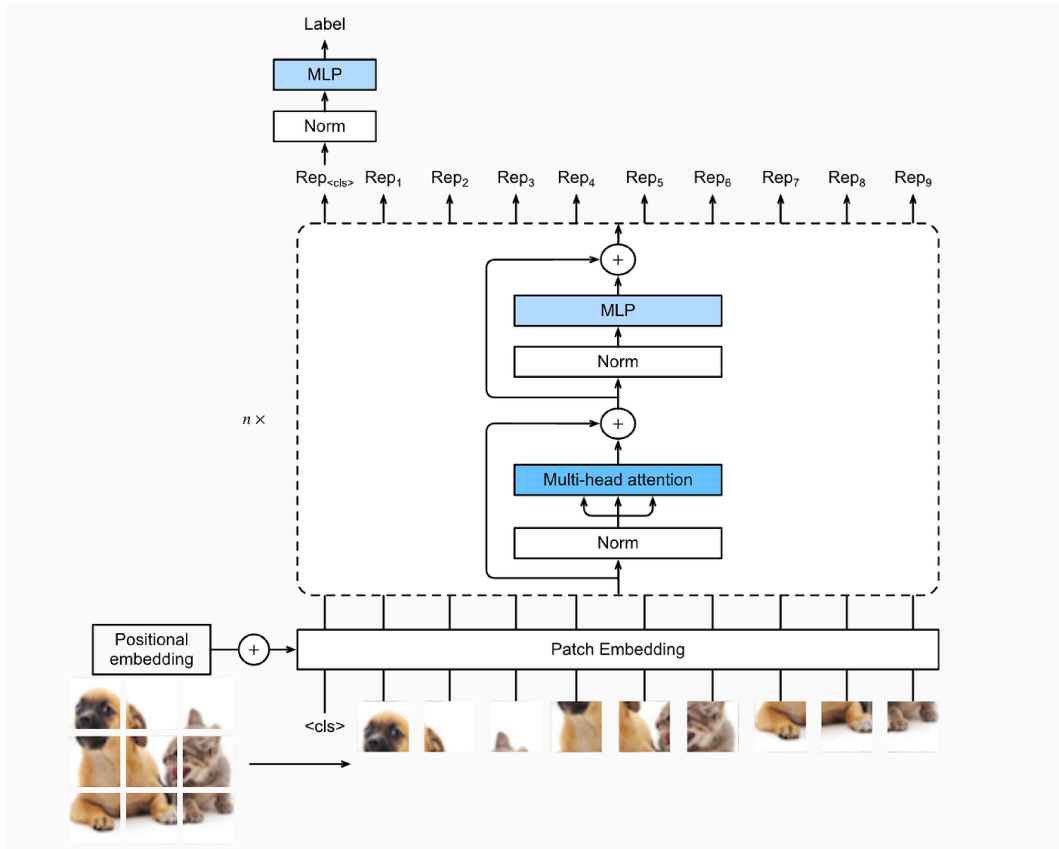
Fonte - Schmucker (2024).

2.3.2 Transformadores de Visão

Os transformadores de visão são baseados em mecanismos de atenção adotando uma abordagem global para a análise de imagens. Os *Transformers* adaptados para tarefas de visão computacional, como o *Vision Transformer* (ViT) (PRABHAKAR; SHIVASHANKAR; MOHANA, 2024), dividem a imagem em pequenas regiões chamadas *patches*, que são tratadas como elementos de uma sequência. Cada *patch* é projetado em um espaço vetorial, permitindo que o modelo processe a imagem de forma análoga a uma sequência de *tokens*. Para preservar a informação espacial, são adicionados *embeddings* posicionais que codificam a localização original de cada fragmento na imagem (VASWANI et al., 2017).

Na Figura 3, observa-se o fluxo de processamento. A imagem de entrada é fragmentada e alinhada em diversos *patches*. Esses fragmentos são combinados com as informações de localização (*Positional embedding*). O modelo então analisará as relações entre essas partes para gerar a classificação final, através de camadas de mecanismos de atenção (*Multi-head attention*) e normalizações.

Figura 3 – Processo de divisão de uma imagem em *patches* e adição de *embeddings* posicionais na arquitetura *Vision Transformer*.



Fonte - Zhang et al. (2023).

O principal mecanismo desses modelos é a atenção (*self-attention*), que permite ao modelo calcular dinamicamente a relevância entre diferentes regiões da imagem (MAURÍCIO; DOMINGUES; BERNARDINO, 2023). Dessa forma, cada parte da imagem pode interagir diretamente com todas as demais, possibilitando a captura de dependências globais desde as primeiras camadas da rede. Essa característica representa uma vantagem significativa em tarefas que exigem compreensão contextual ampla, como é o caso do monitoramento de lojas de varejo a partir de imagens de CFTV (TAKAHASHI et al., 2024).

2.4 Análise de Vídeos

A análise de vídeos difere fundamentalmente da análise de imagens estáticas por incorporar a dimensão temporal como elemento essencial para a compreensão dos dados. Dessa forma, a representação de dados de entrada em modelos de aprendizado profundo é estendida para tensores multidimensionais que incluem a dimensão temporal. Isto é, um vídeo pode ser modelado como uma sequência ordenada de quadros, na qual cada quadro

contribui para a construção de um padrão dinâmico ao longo do tempo (COSTA, 2019).

Para solução dessa diferença fundamental e modelagem espaço-temporal dos vídeos, duas abordagens são utilizadas majoritariamente: as RNCs 3D e os modelos de vídeo baseados em mecanismos de atenção. Assim como no 2D, as abordagens diferem principalmente na forma como capturam relações espaciais e temporais.

As RNCs 3D estendem diretamente o conceito de convolução bidimensional para o domínio espaço-temporal. Nessa abordagem, os filtros convolucionais passam a operar simultaneamente nas dimensões de altura, largura e tempo, permitindo que o modelo extraia características dinâmicas a partir de volumes de dados (TRAN et al., 2018). Dessa forma, padrões de movimento são aprendidos de forma integrada à informação espacial presente nos quadros. Essas redes preservam o viés indutivo das convoluções, como a localidade e a invariância à translação, agora estendidas ao domínio temporal. Isso permite que essas arquiteturas capturem de forma eficiente padrões locais de movimento ao longo de sequências curtas de quadros. No entanto, a modelagem de dependências temporais de longo alcance depende do empilhamento de múltiplas camadas, o que pode aumentar significativamente a complexidade do modelo e dificultar o treinamento. Além disso, a introdução da dimensão temporal nos filtros resulta em um crescimento expressivo no número de parâmetros, elevando o custo computacional de inferência e de treino (HUANG; GUO; GAO, 2020).

Os modelos de vídeo baseados em mecanismos de atenção estendem a arquitetura dos *Transformers* para incorporar explicitamente a dimensão temporal na modelagem dos dados. Para vídeos, a sequência de entrada passa a incluir múltiplos quadros, exigindo a modelagem conjunta de relações espaciais e temporais (BERTASIUS; WANG; TORRESANI, 2021). Seria possível adaptar os modelos 2D aplicando diretamente o mecanismo de *self-attention* sobre todos os *patches* de todos os quadros simultaneamente. No entanto, essa estratégia apresenta custo computacional elevado, uma vez que a complexidade da atenção cresce quadraticamente com o número de elementos da sequência. Para contornar essa limitação, arquiteturas modernas propõem a fatoração da atenção em componentes espaciais e temporais, permitindo modelar essas dimensões de forma separada e mais eficiente. Nesse contexto, a atenção temporal permite capturar relações entre diferentes instantes do vídeo, identificando padrões dinâmicos ao longo do tempo, enquanto a atenção espacial foca na interação entre regiões dentro de um mesmo quadro (ARNAB et al., 2021).

2.5 Arquiteturas para Análise de Vídeos

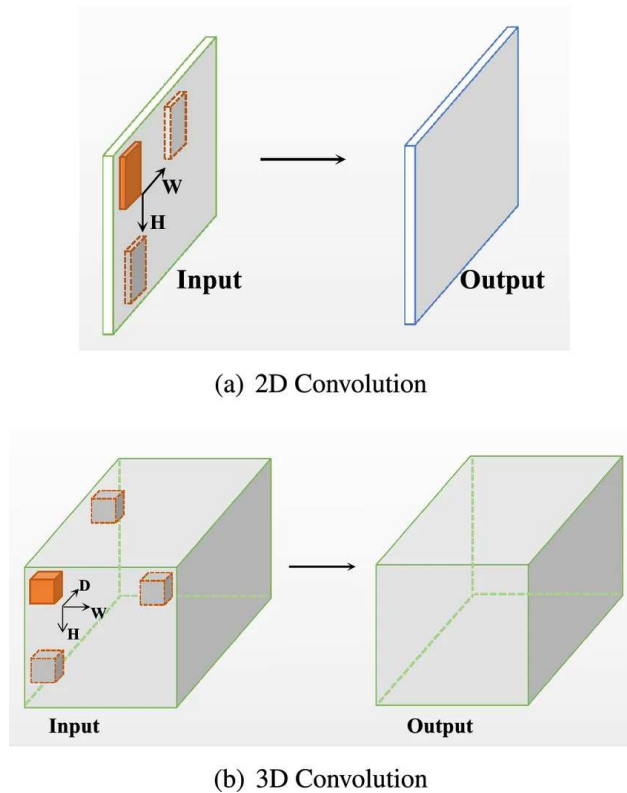
O uso de arquiteturas consolidadas na literatura e amplamente adotadas em aplicações práticas permite o aproveitamento de pesos pré-treinados em grandes bases de dados,

reduzindo o custo computacional e a necessidade de grandes volumes de dados rotulados. Esse aspecto é particularmente relevante em cenários aplicados, nos quais a disponibilidade de dados específicos pode ser limitada. Nesse contexto, este trabalho adota duas arquiteturas já consolidadas para a análise de vídeos: o I3D (*Inflated 3D ConvNet*) e o TimeSFormer. A escolha dessas arquiteturas permite a comparação entre abordagens com diferentes vieses indutivos e capacidades de modelagem, conforme discutido nas seções anteriores.

2.5.1 Inflated 3D ConvNet

O modelo I3D estende arquiteturas convolucionais 2D para o domínio espaço-temporal por meio da chamada inflação de filtros. Nesse processo, filtros bidimensionais previamente treinados em tarefas de classificação de imagens são expandidos para três dimensões, permitindo sua aplicação direta sobre sequências de vídeo. A Figura 4 apresenta a convolução 2D, que atua apenas sobre as dimensões espaciais de altura e largura, e a convolução 3D, que incorpora a dimensão de profundidade, ou seja, o tempo no contexto dos vídeos, possibilitando a extração simultânea de características espaciais e temporais.

Figura 4 – Comparação entre a operação de convolução padrão (2D) e a convolução espaço-temporal (3D) aplicada a volumes de vídeo.



Fonte - Leng et al. (2023).

Uma vantagem importante do I3D é a possibilidade de inicialização a partir de

pesos pré-treinados em grandes bases de imagens, o que facilita o treinamento e melhora a capacidade de generalização. Além disso, sua estrutura baseada em convoluções 3D permite capturar eficientemente padrões locais espaço-temporais (CARREIRA; ZISSERMAN, 2017).

Por outro lado, assim como outras arquiteturas baseadas em convoluções tridimensionais, o I3D apresenta limitações na modelagem de dependências de longo alcance, além de um custo computacional elevado decorrente do aumento do número de parâmetros.

2.5.2 TimeSFormer

O TimeSFormer adapta a arquitetura dos *Transformers* para a análise de vídeos, explorando mecanismos de *self-attention* para modelar relações espaço-temporais, sendo capaz de estabelecer interações diretas entre diferentes regiões e instantes do vídeo.

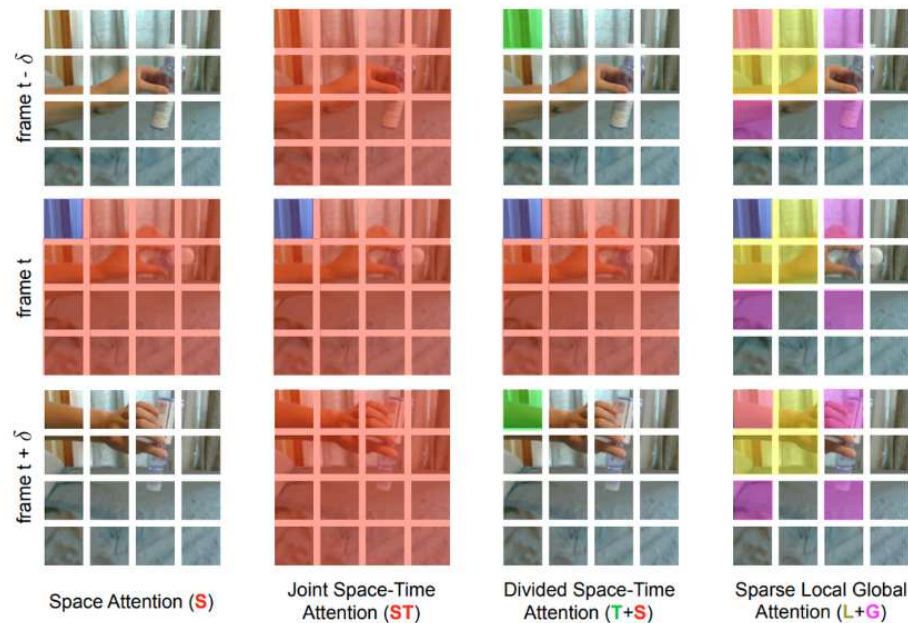
Assim como outros modelos baseados em *Transformers*, o TimeSFormer representa os dados de entrada como sequências de *patches* extraídos dos quadros do vídeo, incorporando informações posicionais para preservar a estrutura espaço-temporal. No entanto, diferentemente dos modelos comuns do mesmo tipo, aqui a atenção é aplicada em diferentes graus ao longo do tempo e do espaço, isto é: ao invés de utilizar todos os *patches* no quadro atual, anterior e posterior para classificação do quadro atual como na *Joint Space-Time Attention* da Figura 5, o TimeSFormer utiliza um subconjunto deste, como na abordagem *Divided Space-Time Attention*, permitindo reduzir o custo computacional sem detrimento significativo da performance (BERTASIUS; WANG; TORRESANI, 2021).

2.6 Transfer Learning e fine-tuning

O treinamento de modelos de aprendizado profundo para tarefas de visão computacional, especialmente no domínio de vídeos, demanda grandes volumes de dados rotulados e elevado custo computacional. Para mitigar esse problema, técnicas de *Transfer Learning* surgem como uma abordagem eficaz, consistindo na reutilização do conhecimento adquirido em grandes domínios de origem para acelerar a convergência e melhorar a generalização em novas tarefas (PAN; YANG, 2009). Nestas técnicas assume-se que representações genéricas, como bordas em imagens ou noções básicas de movimento, são transferíveis para problemas específicos, o que é vital quando os conjuntos de dados alvo são limitados (YOSINSKI et al., 2014).

Tanto o I3D quanto o TimeSFormer, adotados neste trabalho, possuem versões com pesos pré-treinados em grandes conjuntos de dados de vídeo, o que permite sua adaptação para tarefas específicas com menor quantidade de dados. Para o I3D, os pesos iniciais surgem à partir da inflação de filtros 2D originalmente treinados no conjunto de dados ImageNet para a dimensão 3D, seguidos de um pré-treinamento no banco de dados

Figura 5 – Visualização da fatoração do mecanismo de atenção em componentes temporais e espaciais na arquitetura TimeSFormer.



Fonte - Adaptado de Bertasius, Wang e Torresani (2021).

Kinetics-400 (K400). O K400 é um banco de dados abrangente composto por centenas de milhares de vídeos focados em ações humanas. Essa abordagem em duas etapas garante que o I3D pré-treinado herde a forte capacidade de extração de características espaciais do ImageNet e, simultaneamente, aprenda a modelar dinâmicas temporais no domínio de vídeos (DENG et al., 2009; KAY et al., 2017).

Já o TimeSFormer possui pesos pré-treinados no K400 e no Something-Something V2 (SSv2). A escolha de disponibilizar pesos para esses dois bancos de dados baseia-se nas diferenças cruciais entre eles. O K400 possui um forte viés espacial, onde a presença de objetos e o cenário muitas vezes bastam para inferir a ação. Em contraste, o SSv2 foca em interações abstratas e direcionais, onde a compreensão da ordem e evolução temporal é obrigatória para o reconhecimento correto (KAY et al., 2017; GOYAL et al., 2017).

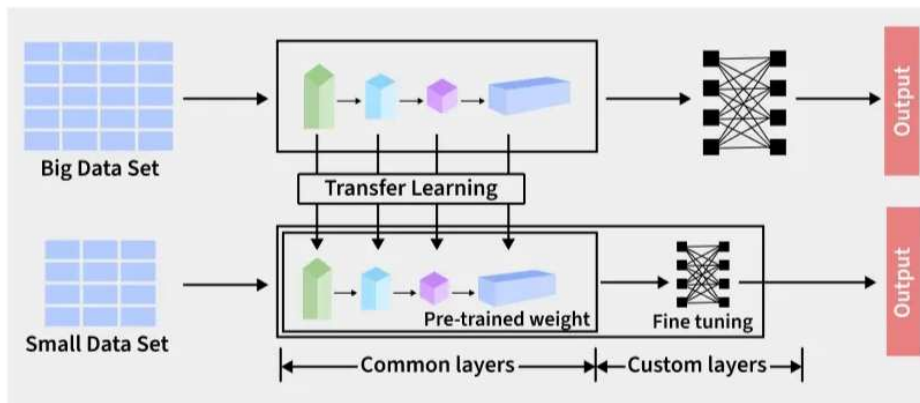
A disponibilidade desses pesos pré-treinados é particularmente relevante em cenários aplicados, nos quais os conjuntos de dados específicos são significativamente menores. No entanto, nessas situações o uso direto dos modelos pré-treinados não é suficiente, uma vez que as distribuições de dados e os padrões de interesse diferem daqueles presentes nos conjuntos de origem. Dessa forma, torna-se necessário adaptar os modelos ao novo domínio por meio do processo de *fine-tuning*, que consiste no ajuste dos parâmetros de um modelo previamente treinado utilizando um novo conjunto de dados, com o objetivo de especializar as representações aprendidas para a tarefa alvo. Diferentemente do treinamento do zero, essa abordagem parte de um estado inicial já informativo, permitindo uma

convergência mais rápida e, frequentemente, melhor desempenho final (TAJBAKHSI et al., 2016).

Esse processo pode ser realizado de diferentes maneiras, dependendo do grau de adaptação desejado. Em uma abordagem mais conservadora, as camadas iniciais do modelo são mantidas congeladas, enquanto apenas as camadas finais são treinadas. Alternativamente, é possível realizar o ajuste completo de todos os parâmetros da rede, permitindo uma adaptação mais profunda ao novo domínio, ao custo de maior risco de *overfitting* (YOSINSKI et al., 2014).

A Figura 6 sintetiza visualmente a união desses conceitos. Como demonstrado, o modelo adquire conhecimento inicial a partir de um grande volume de dados, estabelecendo pesos pré-treinados em suas camadas em comum. Ao ser transferido para uma nova tarefa com dados mais escassos, as camadas que diferem das originais passam pelo *fine-tuning*, permitindo adaptar as representações do modelo para gerar a saída específica sem treinamento do início.

Figura 6 – Fluxo conceitual do processo de *Transfer Learning* e *fine-tuning* adaptando pesos de um domínio de origem para uma tarefa específica.



Fonte - Yosinski et al. (2014).

2.7 Métricas e Função de Perda

O treinamento de modelos de aprendizado profundo para classificação de vídeos requer a definição simultânea de dois componentes fundamentais: a função de perda, responsável por guiar o processo de otimização dos parâmetros da rede, e as métricas de avaliação, utilizadas para quantificar o desempenho do modelo de forma interpretável e comparável (GRANDINI; BAGLI; VISANI, 2020). Em uma classificação binária sob forte desbalanceamento de classes tanto a escolha da função de perda quanto das métricas de avaliação é de extrema importância a fim de evitar vieses em direção à classe majoritária.

A função de perda atua como um mecanismo de retroalimentação do modelo, mensurando o erro entre as predições realizadas e os rótulos reais, permitindo a atualização dos pesos por meio do algoritmo de retropropagação (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para o modelo I3D, cuja saída consiste em uma única probabilidade associada à ocorrência da classe positiva, é empregada uma função de perda apropriada para problemas binários, a *Binary Cross-Entropy* (BCE), baseada na comparação direta entre a probabilidade predita e o rótulo verdadeiro. Essa escolha é coerente com a modelagem probabilística do problema e favorece a penalização mais intensa de predições incorretas realizadas com alta confiança. Por sua vez, no modelo TimeSFormer, cuja saída é estruturada como um vetor de probabilidades para duas classes, utiliza-se uma função de perda de natureza categórica, a *Cross-Entropy Loss* (CEL), que considera simultaneamente as probabilidades atribuídas a cada classe possível. Embora as formulações matemáticas dessas funções difiram, ambas compartilham o mesmo objetivo fundamental: maximizar a probabilidade da classe correta enquanto minimizam a da classe incorreta (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Considerando o desbalanceamento inerente ao problema, são adotadas estratégias de ponderação das classes durante o cálculo da perda, nas quais a penalização atribuída aos erros é proporcional ao grau de sub-representação da classe, sendo mais severa quanto mais minoritária ela for. Essa abordagem é essencial para evitar que o modelo aprenda soluções triviais, como a predição constante da classe majoritária, o que resultaria em desempenho aparente elevado, porém sem utilidade prática (CASTRO; BRAGA, 2011).

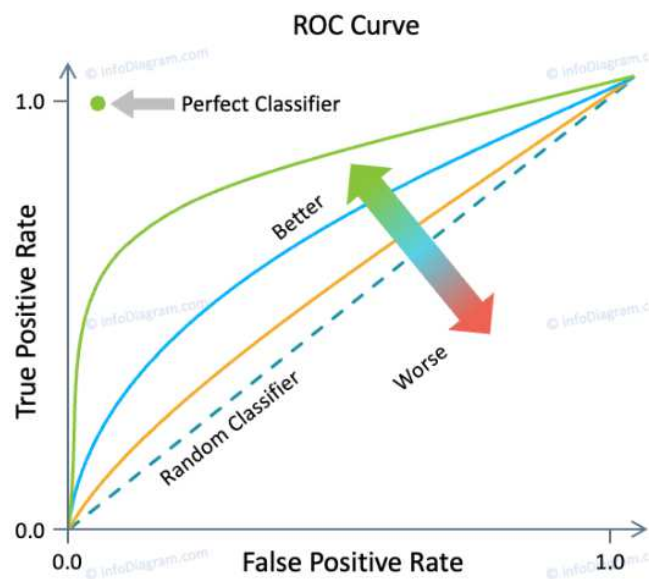
No que diz respeito à avaliação de desempenho, métricas derivadas da matriz de confusão, como precisão e revocação, permitem analisar diferentes aspectos do comportamento do modelo, como sua capacidade de evitar falsos positivos ou de identificar corretamente eventos de interesse. A partir dessas duas medidas, calcula-se o F1-Score, que consiste na média harmônica entre a precisão e a revocação, sendo assim uma métrica amplamente utilizada para resumir o equilíbrio do classificador em um único valor. No entanto, em cenários desbalanceados, a acurácia, revocação ou F1-Score isoladamente podem ser medidas enganosas, não refletindo adequadamente o desempenho sobre a classe minoritária.

Por este motivo, adota-se como principal métrica de avaliação a área sob a curva ROC (*Area Under the Receiver Operating Characteristic Curve* - AUC-ROC). A curva ROC ilustra o desempenho do classificador ao plotar a taxa de verdadeiros positivos contra a taxa de falsos positivos ao longo de diferentes limiares de decisão. A AUC-ROC resume essa curva em um único valor, que quantifica a capacidade do modelo de discriminar entre as classes variando de 0 a 1. Deste modo, quanto mais próximo de 1 o AUC-ROC, melhor o desempenho do modelo, sendo 1 correspondente a um classificador perfeito,

enquanto um valor de 0,5 indica uma classificação aleatória, como mostra a Figura 7. O uso dessa métrica mostra-se especialmente adequado por sua independência de um limiar fixo e por capturar com clareza a relação de compromisso entre acertos e falsos alarmes (GRANDINI; BAGLI; VISANI, 2020).

Adicionalmente, após o treinamento, os melhores modelos são avaliados em um mesmo conjunto de teste, previamente separado do processo de otimização, permitindo uma comparação justa entre diferentes arquiteturas e estratégias de treinamento. Essa abordagem assegura que a escolha da melhor configuração não esteja enviesada por dados utilizados durante o ajuste dos parâmetros, promovendo maior confiabilidade na análise dos resultados. Nesta etapa, devido ao fato de falsos negativos serem mais críticos para o cenário de detecção de furtos, o F2-Score, que atribui mais peso à revocação, é preferível como métrica de avaliação final.

Figura 7 – Exemplo de Curvas ROC, a Área Sob a Curva (AUC) é a métrica empregada para avaliar cenários de classificação desbalanceados.



Fonte - Adaptado de infoDiagram (2024).

Por fim, aspectos relacionados à estabilidade do treinamento também desempenham papel relevante no desempenho final dos modelos. O I3D incorpora mecanismos de normalização baseados em estatísticas do *batch*, em contrapartida com o TimeSFormer, que utiliza estratégias alternativas de normalização, mais adequadas à sua estrutura baseada em mecanismos de atenção (CARREIRA; ZISSERMAN, 2017; BERTASIUS; WANG; TORRESANI, 2021). Essas diferenças estruturais impactam diretamente a dinâmica de treinamento e devem ser consideradas na definição dos hiperparâmetros como a taxa de aprendizado inicial.

3 Descrição do Banco de Dados

O desempenho de modelos de aprendizado profundo em tarefas de classificação de vídeos está diretamente relacionado à qualidade, diversidade e representatividade do conjunto de dados utilizado no treinamento. No contexto da detecção de possíveis furtos no varejo, a disponibilidade de bases de dados públicas é limitada, especialmente quando se busca um equilíbrio entre realismo, anotação adequada e volume de dados. Dessa forma, optou-se pela construção de um banco de dados próprio a partir da agregação e adaptação de múltiplas fontes disponíveis na literatura.

3.1 Construção do conjunto de dados

O conjunto de dados utilizado neste trabalho foi construído a partir da combinação de três bases públicas: DCSASS, Shoplifting Dataset 2.0 e Shoplifting Dataset MNNIT (HERVAS, 2019; HISHAM, 2024; ANSARI; SINGH, 2023). Esses conjuntos apresentam características distintas, incluindo variações no ambiente, no tipo de anotação, na natureza dos eventos registrados e, para os casos de comportamento ilícito, nas estratégias de ocultação dos itens furtados.

Inicialmente, foram considerados os vídeos originais de cada fonte. O conjunto MNNIT é composto por 182 vídeos, sendo 90 classificados como “Normal” (i.e. ação normal de compra) e 92 como “*Shoplifting*” (i.e. vídeo associado a um furto). Trata-se de um banco de dados sintético, construído com a participação de múltiplos atores em um ambiente controlado de laboratório, no qual produtos são dispostos de forma a simular um cenário de varejo. Para a classe *Shoplifting*, observa-se que o padrão predominante de ocultação consiste em transportar os itens para mochilas, como na Figura 8, havendo, contudo, variação consistente quanto às características dos objetos, tais como tamanho, forma, cor, tipo e quantidade. Por outro lado, a classe “Normal” contempla interações típicas de consumidores, como observação, manuseio, retirada e devolução de um ou múltiplos itens, sendo estes iguais aos utilizados nos casos de furto. Ressalta-se ainda que as mochilas, utilizada no contexto de “*Shoplifting*” como meio de ocultação, encontram-se presentes em algumas amostras da classe “Normal”, como forma de reforço positivo à semântica da ocultação, e não ao meio.

O Shoplifting Dataset 2.0 tem 276 vídeos, distribuídos entre 85 “Normal”, 82 da classe “*See and Let*” (Pegar e Deixar) e 109 “*Shoplifting*”. A classe “*See and Let*”, que representa situações em que o indivíduo manipula o produto, mas o devolve ao local de origem, foi incorporada à classe “Normal”, por não representar comportamento ilícito. Adicionalmente, a classe “*Buy*” (com 66 vídeos), presente neste banco de dados, foi des-

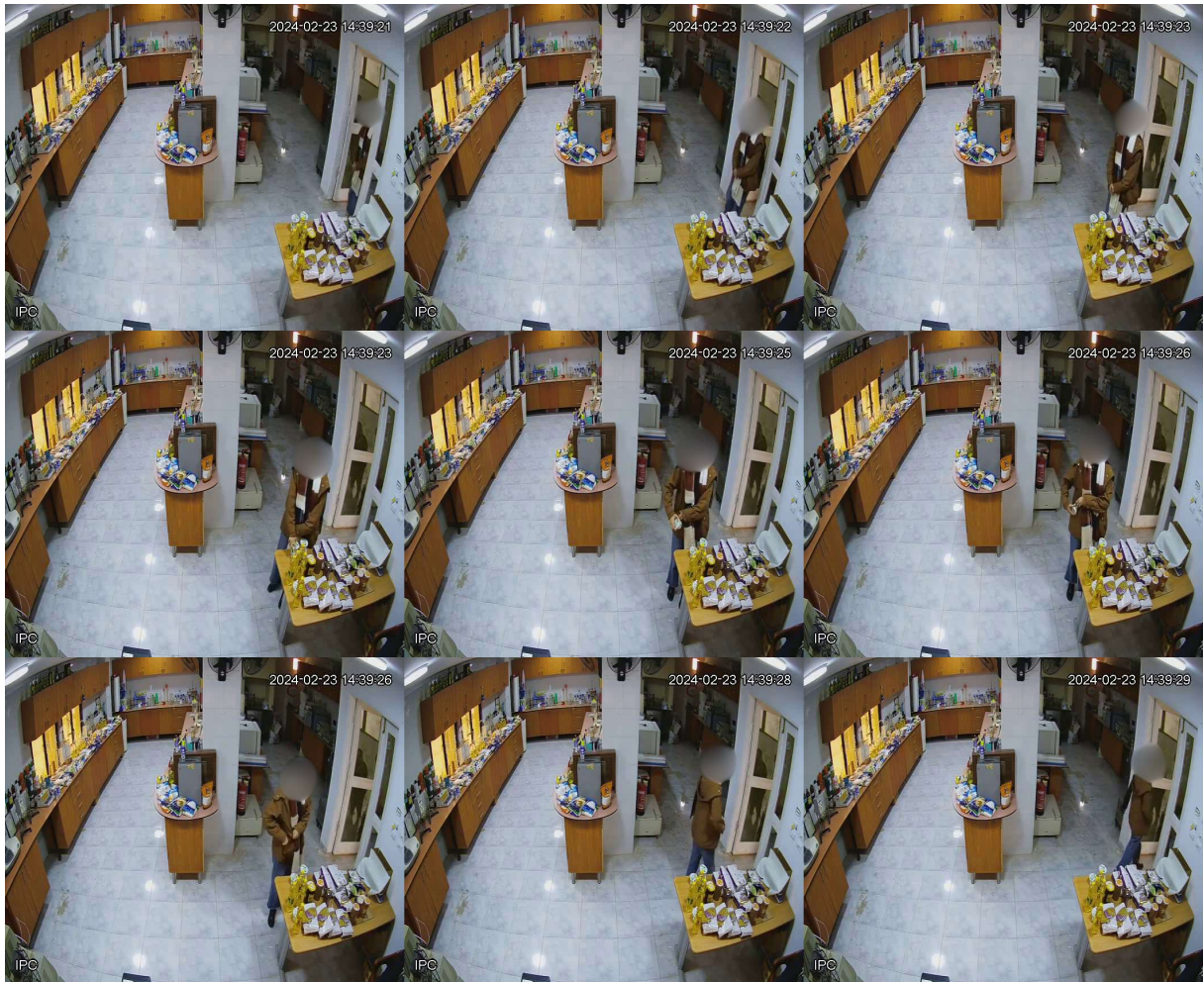
Figura 8 – Exemplo de amostra da classe “*Shoplifting*” do banco de dados MNNIT.

Fonte - Do autor.

cartada. Tal decisão fundamenta-se no fato de que o conjunto em questão é sintético, sendo composto por encenações realizadas em um ambiente doméstico adaptado para simular o varejo, como na Figura 9. Nesse contexto, não há a representação explícita de elementos essenciais ao processo de compra real, como um ponto de pagamento. Dessa forma, as ações rotuladas como “*Buy*” tornam-se semanticamente ambíguas e potencialmente indistinguíveis de comportamentos de furto, comprometendo a consistência das anotações e a confiabilidade do treinamento supervisionado.

O banco de dados DCSASS, por sua vez, apresenta 28 vídeos longos contendo múltiplos eventos. Diferentemente dos demais, nesses vídeos o comportamento de furto não ocorre durante toda a duração, exigindo um processamento adicional. Assim, cada vídeo foi segmentado em múltiplos trechos menores, chamados aqui de blocos, com base nas anotações temporais disponíveis, gerando novos exemplos rotulados como “*Normal*” ou “*Shoplifting*” conforme o rótulo dos trechos que constituem o bloco. Este processo é necessário pois, diferentemente dos anteriores, todas as amostras desse banco de dados são de ambientes e ocorrências reais, para ambas as classes. Desta forma, este conjunto é o que mais colaborou com variabilidade de ângulo de visão, ambiente, método de ocul-

Figura 9 – Exemplo de amostra da classe “*Shoplifting*” do banco de dados Shoplifting Dataset 2.0.



Fonte - Do autor.

tação, item de interesse, como demonstra a Figura 10 com quadros dos diferentes casos. Adicionalmente, foi realizada uma etapa de curadoria dos dados, na qual foram removidos vídeos com baixa variabilidade temporal, isto é, sequências com pouca ou nenhuma mudança significativa entre os quadros, uma vez que tais amostras não contribuem para o aprendizado de padrões dinâmicos. Como resultado, do DCSASS, foram obtidos 57 vídeos da classe “Normal” e 34 da classe “*Shoplifting*”.

Após o processo de agregação, filtragem e segmentação, o banco de dados final para o desenvolvimento do presente trabalho é composto por 549 vídeos, sendo 314 pertencentes à classe “Normal” e 235 à classe “*Shoplifting*”. Em termos de duração, o banco totaliza aproximadamente 8031 segundos de vídeo, dos quais 5047 segundos correspondem a comportamentos normais e 2984 segundos a eventos de furto.

Figura 10 – Exemplos de ambientes do banco de dados DCSASS.



Fonte - Do autor.

3.2 Representação e Pré-processamento

Os modelos utilizados neste trabalho operam sobre sequências de quadros extraídos de vídeos, exigindo padronização das entradas em termos de resolução espacial e taxa de quadros por segundo (FPS – *Frames Per Second*) por amostra. Para cada vídeo, foi realizado um processo de reamostragem temporal, no qual o FPS original é ajustado para atender aos requisitos específicos de cada arquitetura. Em seguida, são extraídos segmentos compostos por um número fixo de quadros, correspondente ao tamanho de entrada esperado por para cada modelo em seus respectivos testes.

Cada amostra, i.e. uma sequência de quadros reamostrados de um vídeo, é tratada de forma independente sendo atribuída a um único rótulo (“Normal” ou “*Shoplifting*”). Durante a fase de treinamento, os quadros são selecionados de forma variável dentro de intervalos uniformemente distribuídos ao longo do vídeo, aumentando a diversidade das amostras. Já nas etapas de validação e teste, a amostragem é realizada de forma determinística e uniforme, garantindo reprodutibilidade na avaliação. Além disso, são aplicadas transformações padrão de pré-processamento, incluindo redimensionamento espacial e normalização, de acordo com os requisitos das arquiteturas utilizadas.

3.3 Aumento de Banco de Dados

Com o objetivo de aumentar a capacidade de generalização dos modelos e mitigar os efeitos do tamanho limitado do conjunto de dados, foram empregadas técnicas de aumento de banco de dados durante o treinamento. Essa abordagem consiste em criar sinteticamente novas amostras para o modelo ao aplicar deformações e transformações nas imagens do conjunto de treinamento original, ensinando a rede a ser invariante a essas modificações visuais (GOODFELLOW; BENGIO; COURVILLE, 2016).

As transformações aplicadas foram variações de cor, que alteram parâmetros como brilho, contraste, saturação e matiz, e espelhamento horizontal, que introduz invariância espacial em relação à orientação da cena, com os seguintes parâmetros:

- **Variação aleatória de cor (*Color Jitter*):** Foram adotadas escalas de variação máxima de $\pm 20\%$ para o brilho, contraste e saturação e $\pm 10\%$ para o matiz. Essas alterações foram aplicadas de forma homogênea e simultânea a todos os quadros de uma mesma amostra temporal, mantendo a consistência temporal e evitando o efeito indesejado de cintilação.
- **Espelhamento horizontal (*Horizontal Flip*):** Configurada com uma probabilidade de ocorrência de 50% para cada amostra processada durante o treinamento.

A Figura 11 demonstra uma sequência de quadros de um mesmo vídeo antes e depois da aplicação das transformações. Tais transformações foram selecionadas com base em sua plausibilidade no contexto real de aquisição dos dados, refletindo variações efetivamente observáveis em sistemas de câmeras, ao mesmo tempo em que evitam alterações não realistas, como o espelhamento vertical. Essas técnicas foram aplicadas de forma consistente em ambos os modelos avaliados, possuindo os mesmos parâmetros.

Figura 11 – Exemplo de transformação de sequência de quadros original (acima) após a aplicação das técnicas adotadas para aumento de banco de dados (abaixo).



Fonte - Do autor.

É importante salientar que o aumento de banco de dados foi utilizado exclusivamente durante a fase de treinamento, não sendo aplicado nos conjuntos de validação e teste, a fim de preservar a integridade da avaliação e permitir uma comparação justa entre diferentes configurações.

3.4 Divisão do Conjunto de Dados

Como de praxe no treinamento de modelos no aprendizado supervisionado, o banco de dados final foi dividido em três subconjuntos distintos: treinamento, validação e teste, seguindo a proporção de 70%, 15% e 15%, respectivamente (CHOLLET, 2021). Como resultado, o subconjunto de treinamento é composto por 314 amostras, já o de validação 82 e o de teste 83. Essa divisão foi realizada de forma a garantir que cada subconjunto fosse representativo da distribuição global das classes, ou seja, de forma estratificada, seguindo

a distribuição 57,3% Normal e 42,7% *Shoplifting* do conjunto completo. A separação foi conduzida ao nível de vídeo, assegurando que amostras derivadas de um mesmo vídeo não estivessem presentes em múltiplos subconjuntos, evitando assim vazamento de informação entre as fases de treinamento e avaliação.

Observa-se que o conjunto de dados agregado apresenta um leve desbalanceamento entre as classes, com aproximadamente 1,34 vezes mais vídeos da classe “Normal” em relação à classe “*Shoplifting*”. Para mitigar os efeitos desse desbalanceamento, foram utilizadas funções de perda ponderadas durante o treinamento (JOHNSON; KHOSHGOFTAAR, 2019).

4 Resultados do Treinamento

Este capítulo apresenta a avaliação experimental das arquiteturas I3D e TimeSFormer desenvolvidas para a identificação de detecção de furto em vídeos. Através da análise de dezesseis configurações experimentais distintas, investiga-se o impacto de estratégias de adaptação de pesos, uso de modalidades múltiplas de entrada e resoluções temporais, resultando na seleção e avaliação dos modelos mais adequados para implementação prática.

4.1 Configuração dos Experimentos

A fim de gerar resultados comparativos, foram avaliadas 16 configurações experimentais, combinando um dos modelos avaliados (I3D ou TimeSFormer), modalidade de entrada (imagem RGB ou RGB + imagem de fluxo óptico¹), estratégia de *fine-tuning* (retreinamento do modelo completo ou apenas das camadas finais, chamadas de “cabeça” do modelo), pré-treinamento e quantidade de quadros, conforme apresentado na Tabela 1. Cabe destacar que, em todas as configurações, o modelo I3D foi treinado utilizando sempre a entrada fixa de 64 quadros. Essa restrição foi aplicada devido à dependência rígida da arquitetura I3D em relação ao tamanho de seus filtros temporais. Portanto, fixou-se o I3D em sua janela padrão para garantir o aproveitamento adequado da rede original, variando-se a quantidade de quadros no modelo TimeSFormer, cuja arquitetura baseada em atenção é mais flexível, como forma de aproximar o volume de processamento temporal entre os dois modelos. Todos os experimentos compartilham o mesmo banco de dados, descrito no capítulo anterior, carregado através de semente de aleatoriedade fixa para garantir a reprodutibilidade das partições de treino, validação e teste em ambas as arquiteturas.

Tabela 1 – Variações de experimentos realizados. Os símbolos “✓” indicam os experimentos conduzidos, enquanto “–” indica configurações que não foram treinadas.

Modelo	fine-tuning	Modalidade	Número de quadros:		8 quadros		32 quadros		64 quadros	
			Dados de pré-treinamento:	K400	SSv2	K400	SSv2	K400	SSv2	
I3D	Completo	RGB		–	–	–	–	✓	–	
		RGB + Fluxo Óptico		–	–	–	–	✓	–	
	Somente cabeça	RGB		–	–	–	–	✓	–	
		RGB + Fluxo Óptico		–	–	–	–	✓	–	
TimeSFormer	Completo	RGB		✓	✓	✓	✓	✓	✓	
	Somente cabeça	RGB		✓	✓	✓	✓	✓	✓	

¹ Imagem que representa o movimento aparente de objetos entre quadros consecutivos, codificando direção e magnitude do deslocamento dos pixels.

O desbalanceamento entre classes é tratado por meio de pesos na função de perda: no I3D, o parâmetro *pos_weight* da BCE *loss* recebe a razão $N_{Normal}/N_{Shoptlifting}$. No TimeSFormer, o mesmo peso é passado à CEL *loss* por meio de um *Trainer* customizado.

A Tabela 2 resume os hiperparâmetros compartilhados entre os experimentos. O tamanho do lote é definido automaticamente por busca binária, respeitando o limite de memória da GPU com margem de segurança de 80%.

Tabela 2 – Parâmetros comuns de treinamento.

Parâmetro	Valor
Semente aleatória	42
Épocas máximas	70
Tamanho do lote	auto (máx. para GPU)
Acúmulo de gradiente	1
Resolução da Imagem de Entrada	224×224

4.2 Preparação do Modelo I3D

Para adequação à tarefa de classificação binária, a cabeça original do I3D, projetada para 400 classes, é substituída por uma nova camada de convolução 3D com saída escalar ($1024 \rightarrow 1$), seguida de uma ativação sigmoide na função de perda BCE. Essa substituição ocorre após o carregamento dos pesos pré-treinados, de forma que a nova camada é inicializada aleatoriamente mantendo inalterados os demais pesos pré-treinados do *backbone*.

Cada amostra é representada por 64 quadros no formato RGB retirados em sequência de um mesmo vídeo. Além disso, também avaliou-se uma variação com técnica de fusão tardia, utilizando dois modelos independentes: um processando os quadros RGB e o outro processando os dados de fluxo óptico. A predição final da classe do vídeo foi obtida calculando-se a média das probabilidades geradas pela saída de cada um dos modelos durante a inferência.

A taxa de aprendizado do otimizador (*Adaptive Moment Estimation* – Adam) foi ajustada para $\eta = 1 \times 10^{-3}$, sem agendador de decaimento. Não há mecanismo de parada antecipada, todos os experimentos executam as 70 épocas integralmente, sendo os pesos salvos ou descartados de acordo com AUC-ROC no conjunto de validação.

4.3 Preparação do Modelo TimeSFormer

No TimeSFormer, a adaptação para a tarefa binária também ocorre por meio da substituição da cabeça classificadora. O modelo é carregado de forma a permitir que

a biblioteca reinicialize automaticamente a camada classificadora quando o número de classes difere da arquitetura original ($400 \rightarrow 2$), enquanto todos os demais pesos do *backbone* são preservados.

O TimeSFormer é treinado por meio da abstração **Trainer** da biblioteca *Transformers*. O *Trainer* é configurado para avaliar o modelo ao final de cada época e aplicar parada antecipada com paciência de 7 épocas, ou seja, o treinamento é interrompido automaticamente se a AUC-ROC de validação não melhorar por 7 épocas consecutivas. Esse mecanismo reduz o tempo de treinamento e mitiga o risco de sobreajuste, mas implica que diferentes experimentos podem ter números distintos de épocas efetivas.

4.3.1 Interpolação dos *Time Embeddings*

Os parâmetros originais do modelo pré-treinado do TimeSFormer foram obtidos utilizando amostras com 8 quadros em sequência como entrada. No entanto, como forma de tornar a comparação entre TimeSFormer e I3D (que trabalha com 64 quadros na entrada) mais adequada, exploraram configurações com entradas compostas por 8, 32 e 64 quadros em sequência. Quando o número de quadros difere do original, os *time embeddings* pré-treinados ($1 \times 8 \times 768$) são interpolados linearmente para a nova resolução temporal, por exemplo ($1 \times 32 \times 768$), preservando parte do conhecimento temporal aprendido durante o pré-treinamento.

Essa interpolação tem uma implicação direta na estratégia de congelamento: nos experimentos com *fine-tuning* somente da cabeça, os *time embeddings* interpolados são explicitamente descongelados, pois seus valores, embora aproximados dos pesos originais pela interpolação, precisam ser ajustados ao novo número de quadros. Na prática, isso significa que os experimentos “somente cabeça” com 32 e 64 quadros possuem significativamente mais parâmetros treináveis do que os com 8 quadros.

4.3.2 Taxa de Aprendizado e Ausência de *Batch Normalization*

A taxa de aprendizado do otimizador do TimeSFormer (*Adam with Decoupled Weight Decay* – AdamW) é fixada em $\eta = 1 \times 10^{-5}$, duas ordens de grandeza inferior à do I3D. Essa diferença se justifica por dois fatores arquiteturais. Primeiro, o TimeSFormer utiliza *Layer Normalization* em vez de *Batch Normalization*. Enquanto a *Batch Norm* normaliza as ativações com base nas estatísticas do mini-lote e possui parâmetros de escala e deslocamento que ajudam a estabilizar o treinamento com taxas de aprendizado mais altas, a *Layer Norm* realiza a normalização por amostra individual. Consequentemente, essa característica torna os gradientes do TimeSFormer mais sensíveis a atualizações agressivas de pesos. Segundo, o TimeSFormer é um modelo consideravelmente maior, e modelos com mais parâmetros pré-treinados em larga escala tendem a ser mais suscetíveis à degradação

de conhecimento quando submetidos a taxas de aprendizado elevadas.

4.4 Comparação da Dimensão dos Treinamentos

Além das diferenças de configuração experimental, os modelos também variam significativamente em capacidade representacional e quantidade de parâmetros efetivamente ajustados durante o treinamento. Neste trabalho, utiliza-se a notação 8f, 32f e 64f para indicar o número de quadros de entrada considerados por cada modelo, isto é, o tamanho da janela temporal processada. A Tabela 3 apresenta a comparação entre número total de parâmetros, quantidade treinável e proporção atualizada em cada estratégia de *fine-tuning*. Esses valores ajudam a interpretar eventuais diferenças de desempenho entre treinamentos completos e ajustes restritos à cabeça classificadora.

Tabela 3 – Número total e treinável de parâmetros.

Modelo	Ajuste	Total	Treináveis
I3D (64f)	Completo	12,3 M	12,3 M
	Somente cabeça		1,0 k
TimeSFormer (8f)	Completo	121,3 M	121,3 M
	Somente cabeça		1,5 k
TimeSFormer (32f)	Completo	121,3 M	121,3 M
	Somente cabeça		26,1 k
TimeSFormer (64f)	Completo	121,3 M	121,3 M
	Somente cabeça		50,7 k

Observa-se que o TimeSFormer possui aproximadamente dez vezes mais parâmetros totais que o I3D. Entretanto, na estratégia de ajuste somente da cabeça, a quantidade de pesos treináveis permanece reduzida, crescendo progressivamente com o aumento do número de quadros devido à adaptação dos *time embeddings*.

4.5 Dinâmica de Treinamento e Convergência

A análise das curvas de perda durante o treinamento revela a dinâmica de aprendizado e a capacidade de adaptação de cada arquitetura. A Figura 12 apresenta o comportamento da função de perda ao longo das épocas para os experimentos do I3D e do TimeSFormer.

No I3D (Figura 12a), observa-se que os modelos submetidos ao *fine-tuning* completo ("Completo" nas imagens) convergem rapidamente, estabilizando-se em menos de 10 épocas. Por outro lado, as configurações com ajuste restrito ("Cabeça" nas imagens) apenas à cabeça classificadora estacionam em patamares elevados de erro, próximos a 0,55,

o que indica uma capacidade muito limitada de adaptação aos dados do domínio alvo quando os extratores de características espaciais e temporais são mantidos congelados.

Padrão semelhante é verificado no TimeSFormer (Figura 12b). Os experimentos com descongelamento total do *backbone* apresentam rápida convergência, o que acionou o mecanismo de parada antecipada entre a 10^a e a 27^a época. Em contrapartida, as variações de apenas cabeça descongelada permanecem estabelecidas em alto erro mesmo após a execução das 70 épocas máximas estipuladas, reforçando que a complexidade da tarefa de detecção de furto exige a atualização dos pesos profundos da rede.

4.6 Evolução das Métricas na Validação

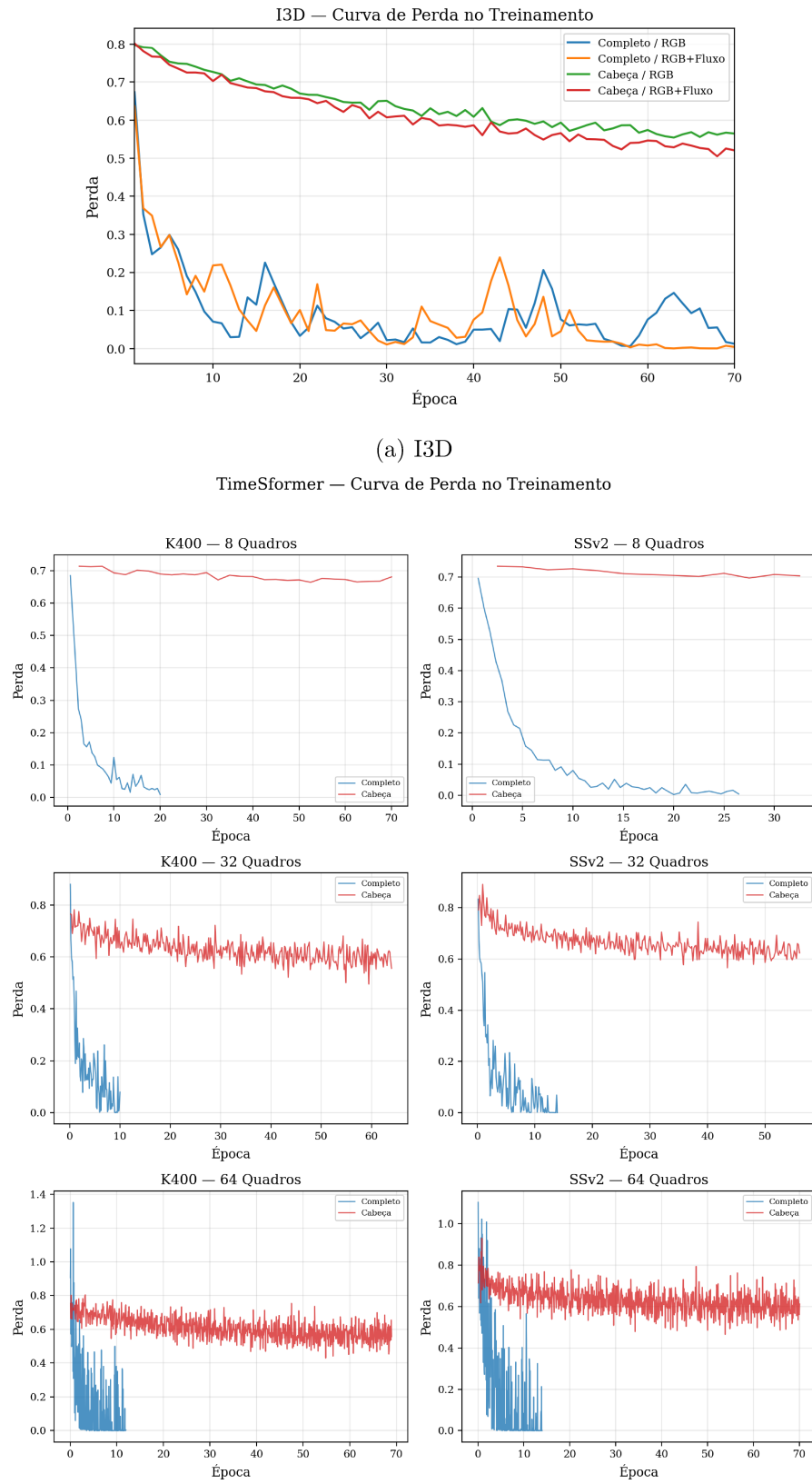
O desempenho no conjunto de validação confirma as severas limitações impostas pelo treinamento somente da cabeça do modelo em comparação à atualização de todos os parâmetros. A Figura 13 ilustra a evolução da Acurácia, F1-Score e AUC-ROC para o modelo I3D. É evidente que as configurações de *fine-tuning* completo convergem rapidamente logo nas primeiras épocas, superando a marca de 0,94 de AUC-ROC.

Um ponto de destaque é o comportamento do fluxo óptico: no cenário de *fine-tuning* completo, a adição do fluxo não proporcionou ganhos significativos em relação à modalidade puramente RGB, sugerindo que a rede espacial atualizada consegue extrair os indícios de movimento necessários. No entanto, quando apenas a cabeça de classificação é treinada, a adição do fluxo óptico eleva visivelmente o desempenho, indicando que a explicitação do movimento compensa parcialmente a incapacidade da rede congelada de adaptar suas representações espaciais para a nova tarefa.

Para o TimeSFormer, as métricas de validação na Figura 14 revelam uma disparidade ainda mais drástica entre as estratégias de treinamento. O treinamento exclusivo da cabeça de classificação falha em produzir resultados significativos, estagnando em valores baixos de Acurácia e F1-Score, demonstrando que o mecanismo de auto-atenção espaço-temporal requer forte adaptação ao novo domínio-alvo.

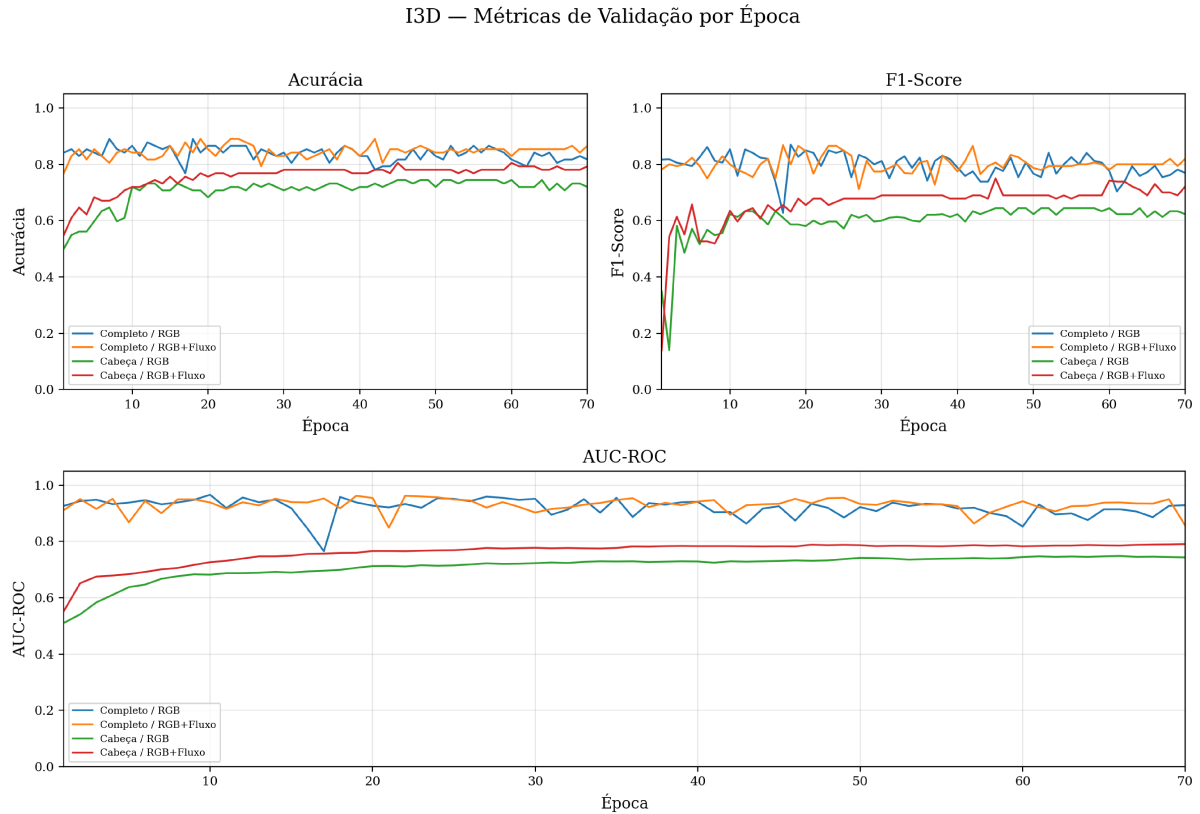
Em contrapartida, os modelos com *fine-tuning* completo evidenciam grande robustez e eficiência, com os seis experimentos atingindo valores de AUC-ROC superiores a 0,91. Além da alta performance, nota-se que as configurações completas atingem a estabilidade de desempenho e acionam critérios de parada antecipada muito cedo, independentemente do banco de dados de pré-treino ou da resolução temporal de entrada (8, 32 ou 64 quadros). O aumento na quantidade de quadros não demonstrou alterar significativamente o teto máximo das métricas alcançadas, mas manteve a rápida convergência dos modelos.

Figura 12 – Curvas de perda de treinamento para os experimentos do I3D (a) e TimeSFormer (b) nas modalidades "Completo" e "Cabeça".



Fonte - Do autor.

Figura 13 – Evolução das métricas de validação para o I3D.



Fonte - Do autor.

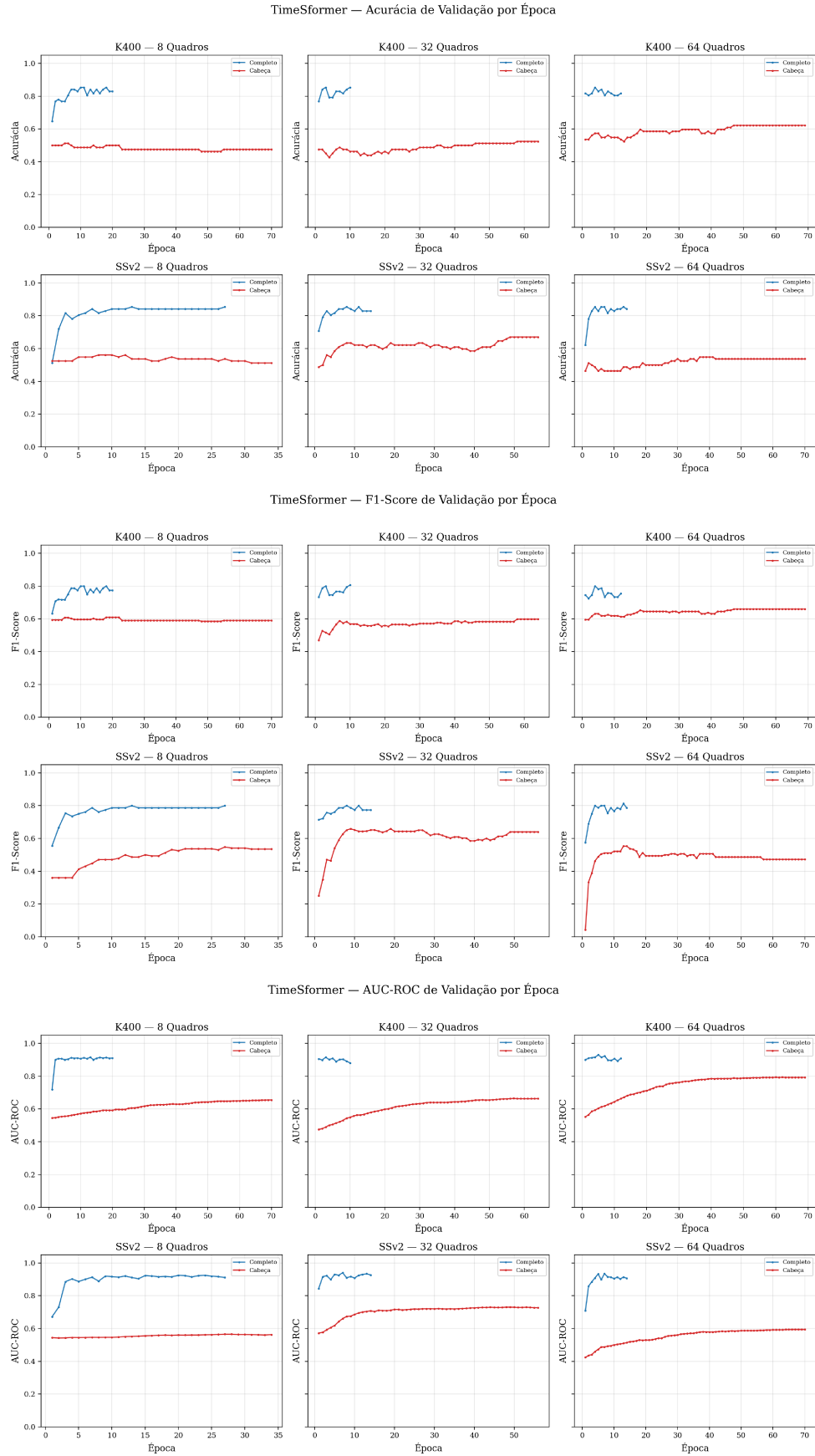
4.7 Avaliação Global e Seleção de Modelos

A fim de estabelecer o melhor representante de cada arquitetura, a Figura 15 apresenta os resultados de AUC-ROC de validação para cada experimento. O modelo I3D na configuração Completa/RGB liderou esta etapa com AUC-ROC de 0,966, enquanto o TimeSFormer atingiu seu melhor desempenho na configuração pré-treinado no SSv2, com 32 quadros e ajuste completo, que obteve 0,940.

4.8 Desempenho no Conjunto de Teste

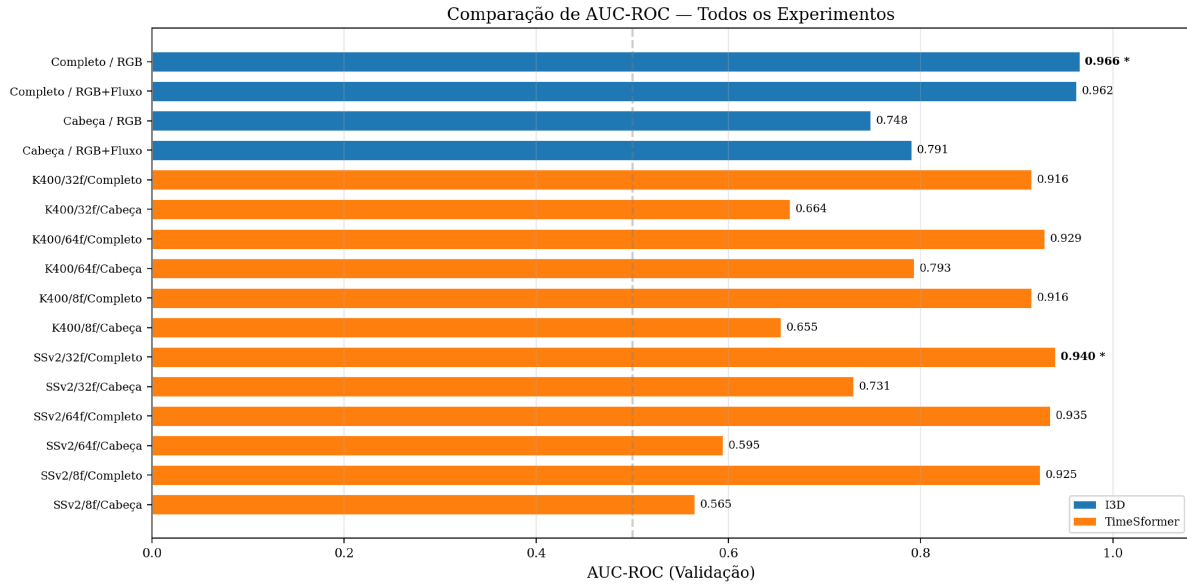
A avaliação definitiva foi conduzida através da inferência com o melhor modelo de cada arquitetura no conjunto de teste composto por 83 amostras inéditas (47 da classe *Normal* e 36 da classe *Shoptlifting*). Os resultados reportados na Tabela 4 demonstram uma inversão no ranqueamento de desempenho em relação à fase de validação. Embora a avaliação do domínio priorize a revocação por meio do F2-Score, destaca-se que o modelo TimeSFormer (SSv2 / 32 quadros / Completo) também alcançou uma precisão de 100%, evidenciando maior especificidade e um controle sobre os falsos positivos. Em contraste, o I3D (Completo / RGB) apresentou precisão de 72,3%, indicando um viés mais conservador

Figura 14 – Evolução das métricas de validação para o TimeSFormer.



Fonte - Do autor.

Figura 15 – Comparação geral de AUC-ROC entre todas as 16 configurações experimentais.



Fonte - Do autor.

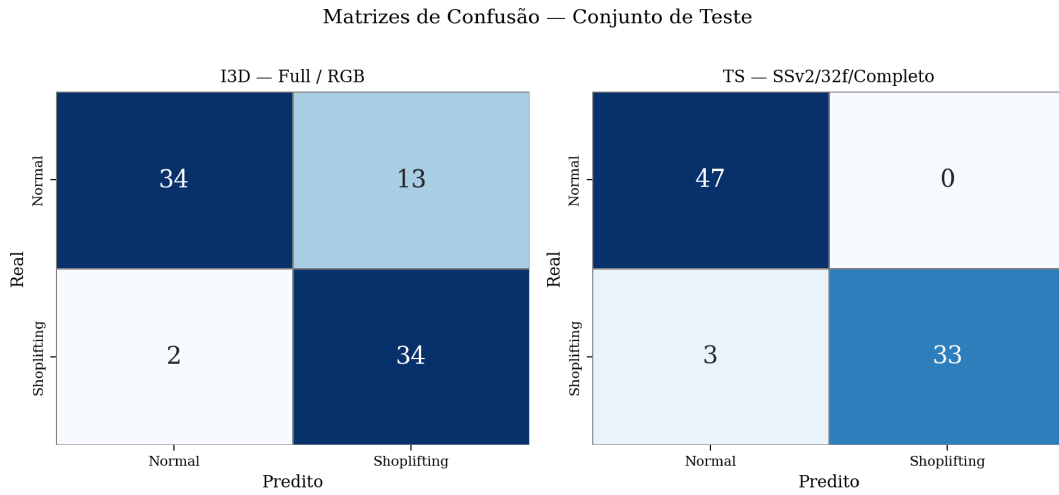
na tomada de decisão, com maior tendência a classificar amostras normais como casos anômalos. Já a diferença em revocação entre os modelos foi consideravelmente menor, de apenas 2,7%, o que levou o TimeSFormer ao melhor resultado geral na avaliação final.

Tabela 4 – Métricas no conjunto de teste dos melhores modelos (Precisão, Revocação e F2-Score).

Modelo	Experimento	Precisão	Revocação	F2-Score
I3D	Completo / RGB	0,723	0,944	0,890
TimeSFormer	SSv2/32f/Completo	1,000	0,917	0,932

As matrizes de confusão na Figura 16 detalham quantitativamente as classificações incorretas. A discrepância sugere que o I3D, além de conservador, pode ter sofrido um leve sobreajuste no conjunto de validação, enquanto o TimeSFormer demonstrou uma capacidade superior de generalização para dados totalmente desconhecidos. Esse comportamento deve-se, muito provavelmente, à superioridade arquitetural do mecanismo de auto-atenção na captura de dependências espaço-temporais globais e à robustez das representações obtidas durante o pré-treinamento, fatores que mitigam o risco de sobreajuste mesmo em um modelo com maior número de parâmetros.

Figura 16 – Matrizes de confusão do melhor I3D e do melhor TimeSFormer no conjunto de teste.



Fonte - Do autor.

4.9 Eficiência

Além do desempenho preditivo, os experimentos foram avaliados sob o viés da viabilidade prática considerando o tempo de inferência. A Tabela 5 apresenta a taxa de transferência (*throughput*) e o tempo de execução (*runtime*) para uma configuração realista de implementação com uma GTX 1660 Super, utilizando lote unitário e entradas sintéticas aleatórias, de modo a evitar otimizações específicas de *cache* ou carregamento de dados.

Tabela 5 – *Throughput* de inferência dos melhores modelos I3D e TimeSFormer.

Experimento	Amostras/s	ms/amostra
I3D - Completo / RGB (64 quadros)	12,57	79,52
TimeSFormer - SSv2/32f/Completo (32 quadros)	1,35	738,53

Em um cenário real de monitoramento, o processamento tende a ocorrer sobre fluxos contínuos de vídeo por meio de janelas temporais deslizantes, nas quais sequências consecutivas de quadros são periodicamente encaminhadas ao modelo. Dessa forma, a análise não depende de clipes previamente segmentados, permitindo que eventos mais longos sejam observados em múltiplas inferências sucessivas ao longo do tempo.

Nessa perspectiva, o modelo I3D apresentou um tempo de 79,52 ms por janela de 64 quadros. O baixo tempo de processamento oferece ampla margem para reduzir o passo entre janelas consecutivas, aumentar a frequência de atualização das predições e escalar a solução para múltiplas câmeras em uma mesma GPU. Tais características tornam a arquitetura particularmente atrativa em ambientes com restrições computacionais ou com

demanda por menor latência operacional. O TimeSFormer, por sua vez, alcançou 738,53 ms por janela de 32 quadros. Embora apresente maior custo computacional, o tempo de inferência permanece compatível com aplicações em tempo real para fluxos individuais, especialmente quando associado a estratégias adequadas de amostragem temporal e infraestrutura dimensionada para essa finalidade.

Em termos práticos, os resultados indicam que ambas as arquiteturas são tecnicamente viáveis para sistemas de monitoramento contínuo, porém com compromissos distintos entre desempenho computacional e capacidade de implantação. O I3D destaca-se pela eficiência e escalabilidade, enquanto o TimeSFormer configura uma alternativa promissora em contextos nos quais maior disponibilidade de recursos computacionais permita explorar seu potencial de modelagem espaço-temporal.

4.10 Síntese dos Resultados

A condução e análise empírica dos 16 experimentos permite consolidar as seguintes constatações a respeito do reconhecimento de ações de furto nas condições avaliadas:

1. **O ajuste profundo da rede foi essencial:** Apenas o treinamento com atualização integral dos pesos permitiu que os modelos aprendessem adequadamente a tarefa proposta. Em ambos os casos, limitar o treinamento à camada classificadora impediu convergência satisfatória.
2. **A contribuição do fluxo óptico no I3D foi condicionada à estratégia de treino:** Quando houve a atualização integral dos pesos, o uso exclusivo de quadros RGB mostrou-se suficiente, tornando a inclusão do fluxo redundante. No entanto, ao congelar o extrator de características, fornecer o fluxo óptico foi fundamental para elevar o desempenho, compensando a incapacidade da rede rígida de extrair novos padrões de movimento apenas pela aparência visual.
3. **O TimeSFormer demonstrou robustez:** Mesmo sob variações no número de quadros e no conjunto de pré-treino utilizado, o modelo sustentou métricas elevadas de AUC-ROC ao longo dos experimentos com treinamento completo.
4. **Nos dados inéditos, o Transformer se destacou:** No conjunto de teste, o TimeSFormer alcançou a melhor capacidade de generalização, obtendo 100% de precisão e o maior F2-Score, enquanto o I3D apresentou comportamento mais conservador, com maior ocorrência de falsos positivos.
5. **Precisão adicional exige maior investimento computacional:** Observou-se uma troca direta entre custo e desempenho: o I3D se destacou pela rapidez e efici-

ência de inferência, ao passo que o TimeSFormer entregou resultados superiores em precisão e redução de alarmes falsos, porém com maior tempo de processamento.

5 Avaliação em Vídeos Reais

Os experimentos apresentados no capítulo anterior foram realizados exclusivamente com amostras do conjunto de dados agregado. Embora esse procedimento permita o treinamento e a comparação controlada entre arquiteturas, as condições típicas de monitoramento reais não são reproduzidas, pois - na prática - o modelo processa um fluxo contínuo de vídeo com duração indefinida. É importante ressaltar que esse tipo de vídeo real não foi utilizado na etapa de treinamento, o que configura o cenário atual como um caso extremo de avaliação para a capacidade de generalização dos modelos treinados. Nesse contexto, a avaliação demanda uma estratégia de inferência diferente da utilizada no treinamento, em que cada entrada corresponde a um clipe de duração fixa com rótulo único. Para lidar com essa diferença, adota-se uma abordagem baseada em janela deslizante, com agregação temporal por quadro, descrita a seguir.

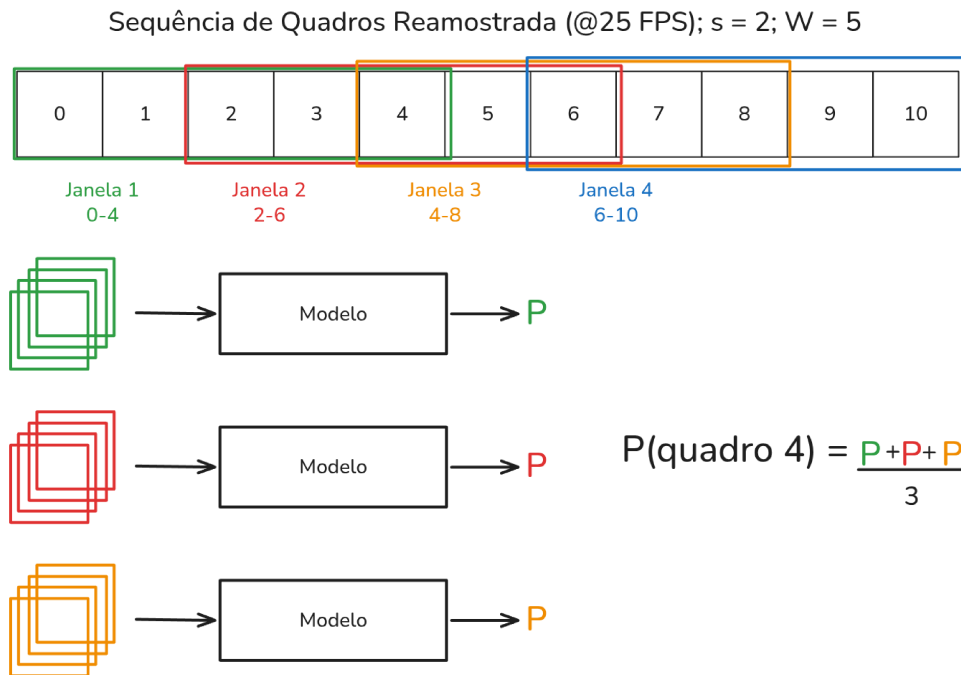
5.1 Metodologia de Inferência

Os vídeos de monitoramento real são primeiramente reamostrados com uma taxa uniforme de 25 quadros por segundo, taxa adotada durante o pré-processamento do conjunto de treinamento. Cada quadro é então redimensionado com preservação da proporção, o maior lado é escalado para 224 pixels e recortado centralmente para a resolução de entrada padrão de ambos os modelos 224×224 pixels.

Sobre a sequência de quadros reamostrada aplica-se uma janela deslizante de tamanho fixo e passo $s = 8$ quadros. O tamanho da janela é mantido idêntico ao usado durante o treinamento de cada modelo: $W_{\text{I3D}} = 64$ quadros e $W_{\text{TS}} = 32$ quadros. A cada posição de janela, a subsequência extraída é encaminhada ao respectivo modelo, que produz uma probabilidade escalar $p \in [0, 1]$ associada à classe *Shoplifting*.

Em razão do passo $s < W$, janelas consecutivas se sobrepõem de forma que um quadro de índice f pertence a mais de uma janela. Em vez de reportar previsões por janela, que introduziriam descontinuidades temporais, opta-se por agregar as previsões ao nível do quadro. A probabilidade atribuída ao quadro f é a média aritmética das previsões de todas as janelas que o contêm, como na Figura 17, o resultado é uma curva de probabilidade contínua e de resolução temporal unitária, alinhada quadro a quadro com o vídeo original reamostrado. Um quadro é classificado como suspeito quando $\hat{p}(f)$ excede o limiar de decisão calibrado no conjunto de validação para cada modelo. Para o modelo I3D, utilizou-se o limiar que otimiza o desempenho global no conjunto de validação, resultando em $\tau_{\text{I3D}} = 0,7449$. Para o TimeSFormer, o limiar corresponde à fronteira natural do *softmax* binário, $\tau_{\text{TS}} = 0,50$.

Figura 17 – Exemplo de aplicação da abordagem de janela deslizante sobre vídeo real.



Fonte - Do autor.

5.2 Conjunto de Vídeos Reais

Foram coletados sete vídeos reais de furtos em estabelecimentos varejistas, sem qualquer uso nas etapas de treinamento ou avaliação anteriores. Esses vídeos, obtidos de fontes públicas na internet, contemplam diferentes tipos de ocorrência em ambientes distintos. As principais características dos vídeos utilizados nesse capítulo encontram-se na Tabela 6.

Tabela 6 – Características dos vídeos reais de furto utilizados na avaliação.

Nome	Duração total (s)	Ambiente	Método de ocultação
furto-celular	8	Salão de beleza	Sem ocultação
furto-pc	29	Loja de roupas	Mochila
furto-roupa	33	Loja de Roupas	Bolsa
furto-supermercado	16	Supermercado	Bolsa
furto-shopping	18	Shopping	Bolso
furto-tenis1	9	Loja de Calçados	Bolsa
furto-tenis2	7	Loja de Calçados	Bolsa

5.3 Resultados

A análise das curvas de probabilidade geradas pela inferência em janela deslizante revela comportamentos distintos entre as duas arquiteturas. Enquanto o modelo

I3D apresenta uma curva frequentemente saturada em patamares elevados, o TimeSFormer demonstra maior seletividade, reagindo de forma mais sensível aos eventos de interesse. Para ilustrar essas dinâmicas e justificar as classificações, os gráficos contínuos foram analisados em conjunto com os quadros originais dos vídeos, permitindo a verificação de como transformações visuais específicas na cena impactaram a confiança de cada modelo.

5.3.1 Casos de Sucesso

Os casos de sucesso foram definidos pela capacidade do modelo em elevar a probabilidade de furto acima do limiar estabelecido sincronizadamente com a ação real de ocultação ou subtração.

O modelo TimeSFormer obteve sucesso notável nos vídeos *furto-supermercado*, *furto-tenis1* e *furto-tenis2*. Conforme ilustrado nas Figura 18, Figura 19 e Figura 20, a curva de probabilidade do modelo mantém-se em níveis basais durante a fase de aproximação e interação normal do cliente com o produto. O pico da curva, que cruza o limiar de detecção, coincide com os quadros em que ocorre a ocultação efetiva do item, retornando à normalidade logo após a conclusão do movimento.

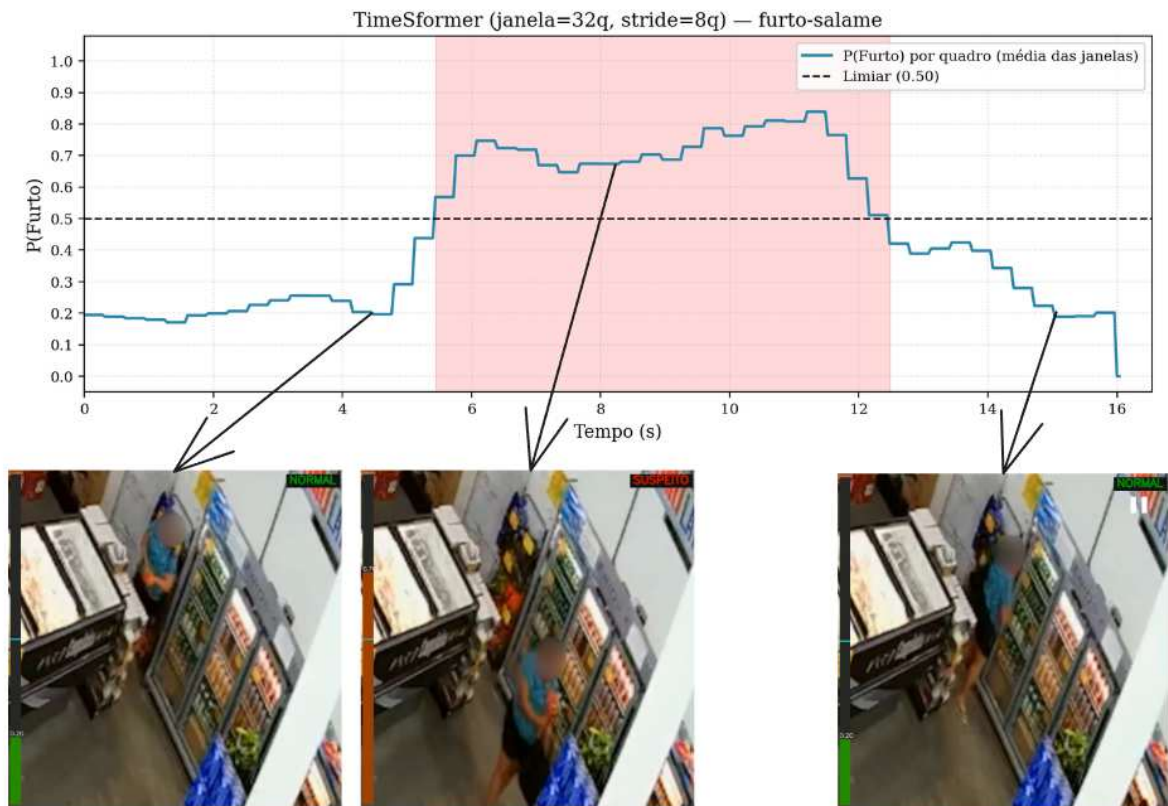
Por outro lado, o modelo I3D demonstrou um sucesso expressivo no vídeo *furto-celular*. Neste cenário, a arquitetura apresentou alta precisão temporal, conforme demonstrado na Figura 21. A avaliação quadro a quadro revela que o modelo classifica a sequência como suspeita estritamente no instante em que o indivíduo retira o aparelho da bancada de forma abrupta. Adicionalmente, no vídeo *furto-pc*, o I3D obteve um sucesso parcial: embora tenha conseguido identificar um pico claro de probabilidade durante o ato do furto, o modelo manteve uma probabilidade residual persistentemente alta em trechos do vídeo onde nenhuma ação anômala ocorria, demonstrado na Figura 22.

5.3.2 Casos de Falha

As falhas observadas evidenciam as limitações de generalização dos modelos frente a oclusões severas e ações de baixa magnitude visual. Os vídeos *furto-roupa* e *furto-shopping* representaram falhas totais para ambas as arquiteturas, com as probabilidades mantendo-se significativamente abaixo dos limiares de alarme durante todo o clipe. A análise visual desses casos sugere que o método de ocultação, como nas Figura 23 e Figura 24, inibiu de forma contundente a extração das características espaço-temporais necessárias para a discriminação do ilícito.

O I3D exibiu uma clara tendência a falsos positivos e à total falta de precisão temporal nos vídeos *furto-tenis1*, *furto-tenis2* e *furto-supermercado*. Nestes cenários, a curva de probabilidade permaneceu saturada em níveis próximos a 1,0 durante praticamente

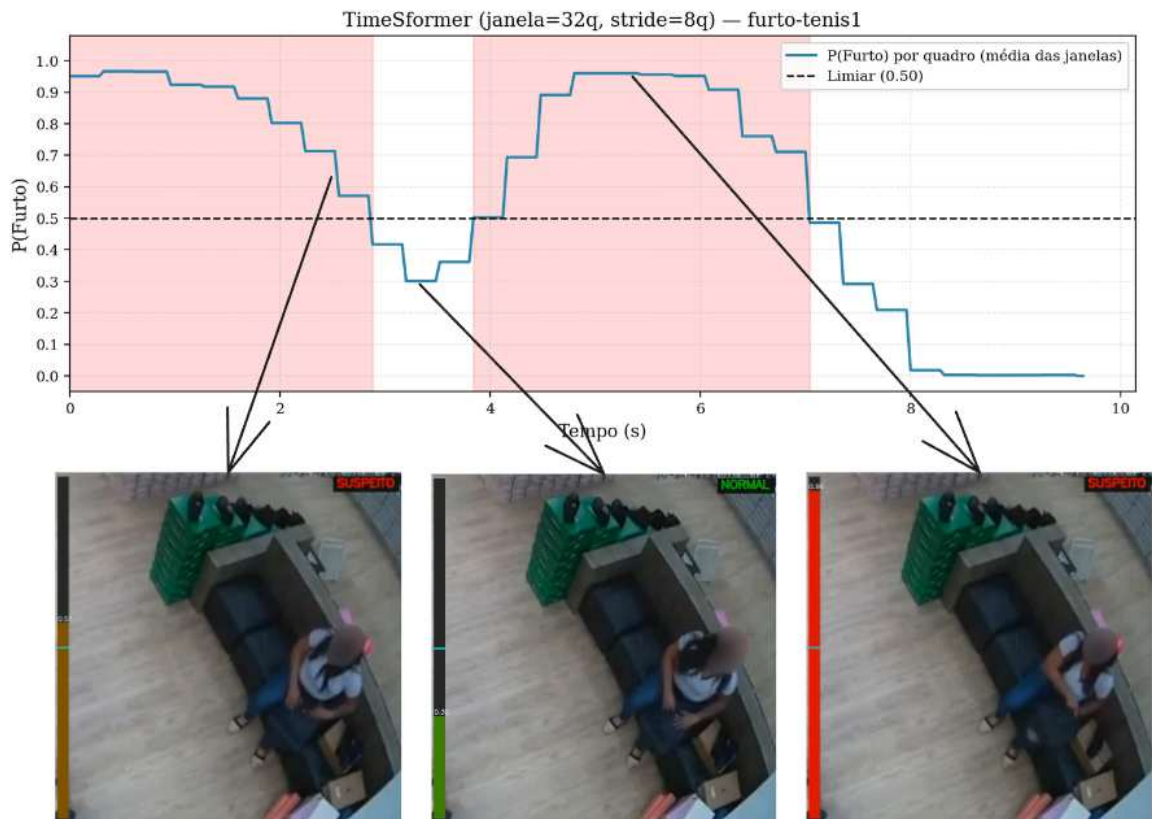
Figura 18 – Curva de probabilidade do modelo TimeSFormer para o vídeo *furto-supermercado*.



Fonte - Do autor.

toda a gravação. A análise dos quadros correspondentes, exemplificada na Figura 25, mostra que a simples presença do indivíduo observando ou segurando o produto já mantém a rede em estado de alerta máximo. Essa saturação evidencia a incapacidade do modelo convolucional de distinguir entre a manipulação legítima e a ocultação intencional. No caso particular de *furto-supermercado*, o modelo chegou a inverter a lógica, atribuindo maior confiança de furto a seções de comportamento normal do que ao momento exato da ocultação, demonstrando na Figura 26.

Em contrapartida, o TimeSFormer apresentou falhas mais relacionadas à latência e à interpretação equívoca do contexto global. No vídeo *furto-celular*, a detecção do evento ocorreu com um atraso considerável, a observação dos quadros críticos na Figura 27 mostra que a probabilidade permaneceu baixa durante o momento da subtração, superando o limiar apenas após a ação ter sido concluída. Por fim, no vídeo *furto-pc*, o modelo interpretou incorretamente uma seção de comportamento normal e não reagiu durante o instante real do furto, na Figura 28. Esse comportamento indica que, em certos arranjos visuais, o mecanismo de auto-atenção global pode se fixar em elementos contextuais irrelevantes da cena, perdendo a sensibilidade para a dinâmica local da ação do indivíduo.

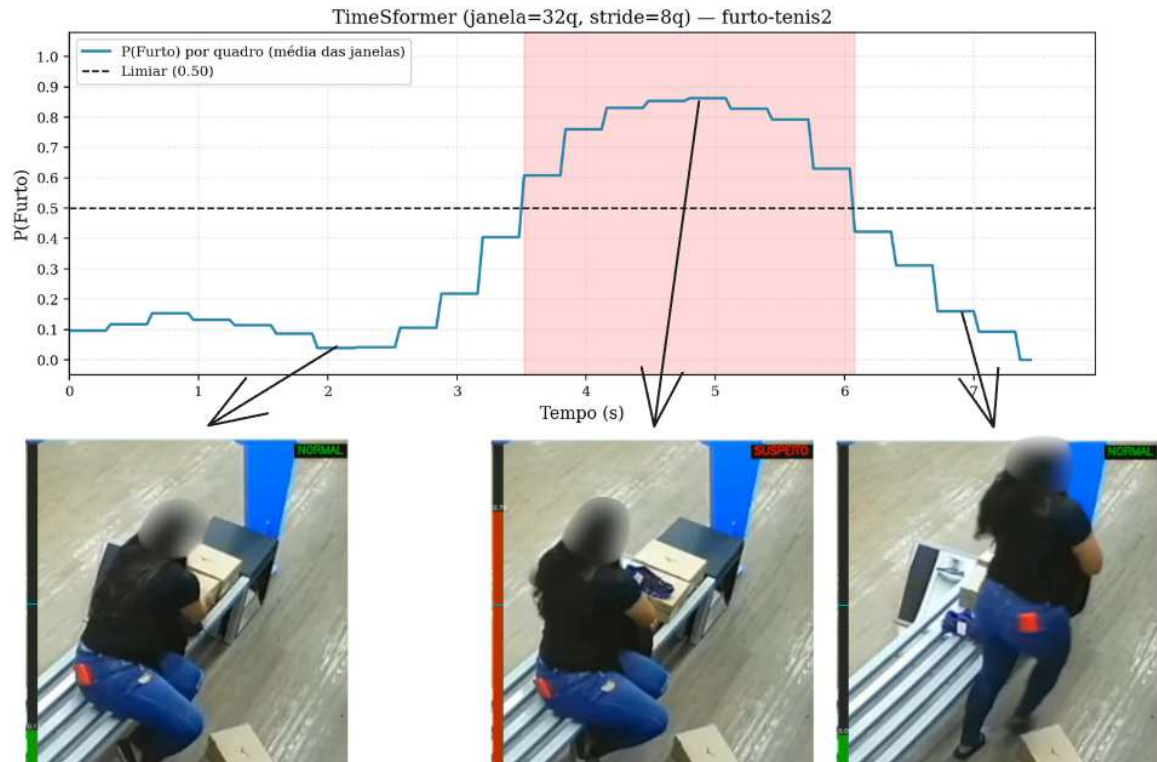
Figura 19 – Curva de probabilidade do modelo TimeSFormer para o vídeo *furto-tenis1*.

Fonte - Do autor.

5.4 Síntese dos Resultados

A avaliação em vídeos reais contínuos revelou comportamentos distintos entre as arquiteturas ao enfrentar as condições típicas de um sistema de monitoramento. A análise preditiva aplicada aos sete vídeos reais consolida as seguintes constatações:

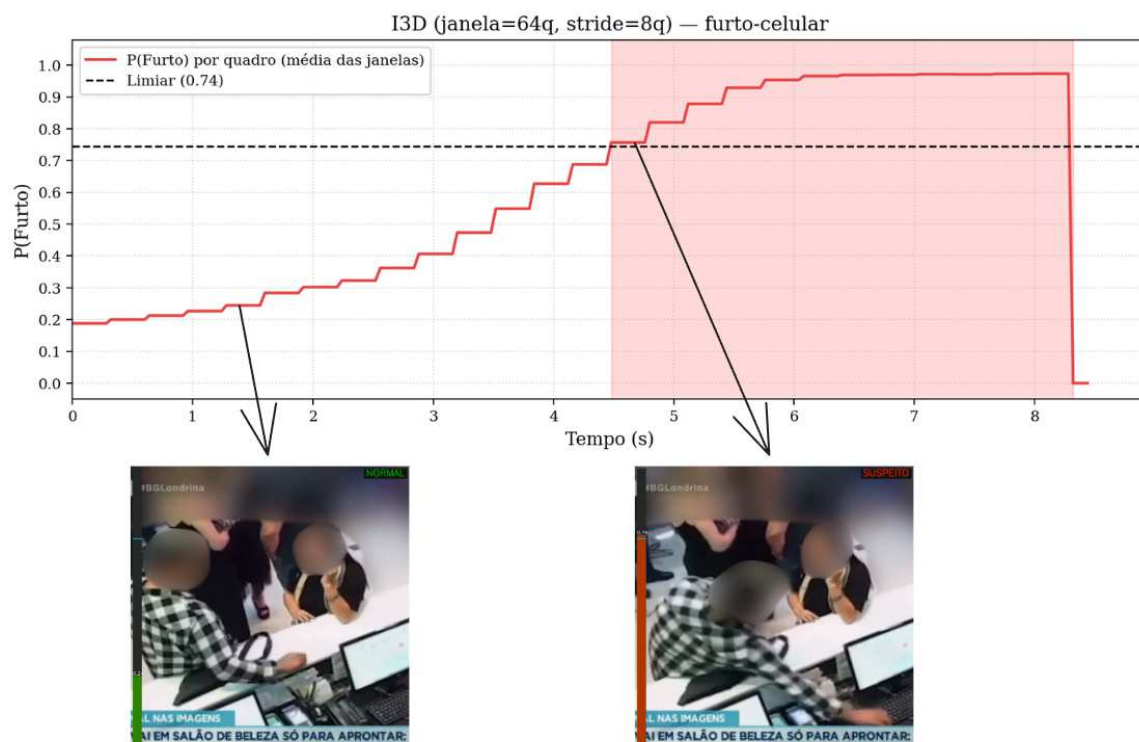
- **Seletividade e Robustez do TimeSFormer:** O modelo baseado no mecanismo de atenção demonstrou maior seletividade em suas predições. A arquitetura obteve sucesso notável ao elevar a probabilidade de anomalia de maneira sincronizada com as ações efetivas de ocultação dos itens, retornando à normalidade em seguida. As limitações encontradas neste modelo estiveram relacionadas predominantemente à latência, resultando em atraso considerável na detecção de alguns eventos. Ademais, o mecanismo de auto-atenção global mostrou-se suscetível a focar em elementos contextuais irrelevantes, perdendo a sensibilidade para as dinâmicas locais da ação do infrator.
- **Sensibilidade a Movimentos e Saturação do I3D:** A arquitetura convolucional apresentou sucesso expressivo e alta precisão temporal na detecção de subtrações ca-

Figura 20 – Curva de probabilidade do modelo TimeSFormer para o vídeo *furto-tenis2*.

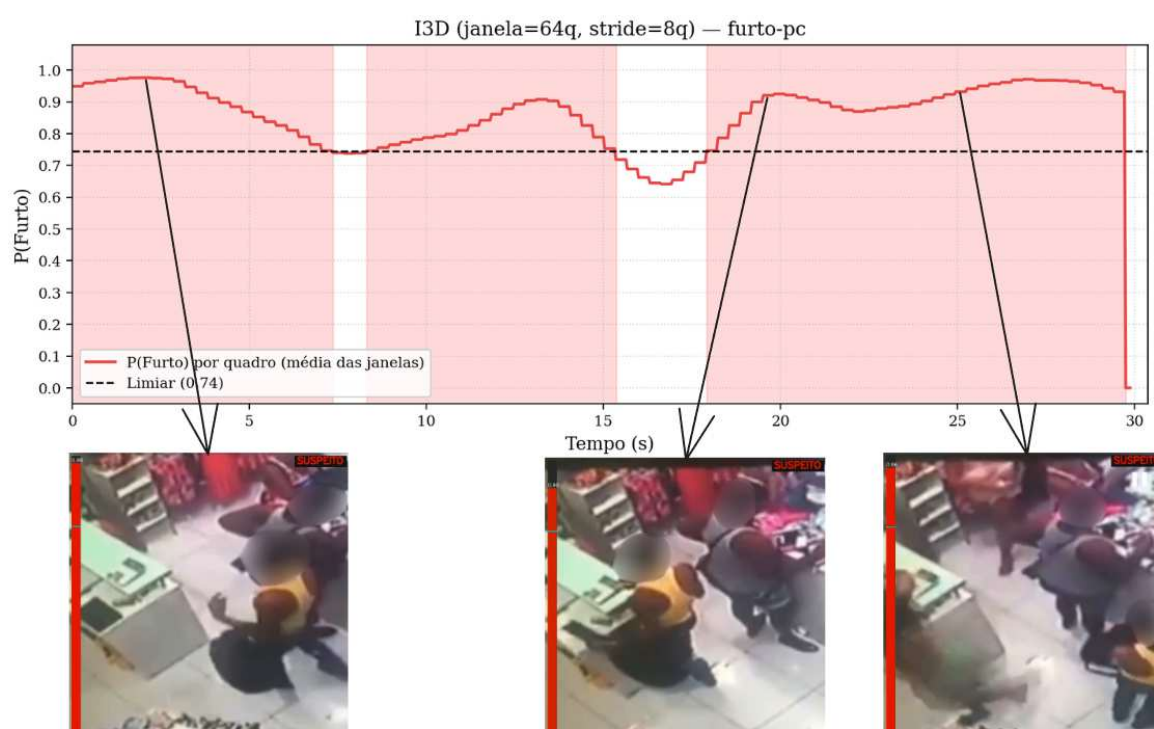
Fonte - Do autor.

racterizadas por movimentos abruptos. Contudo, o I3D exibiu uma clara tendência a falsos positivos e à total falta de precisão temporal na detecção de outras modalidades de furto. As curvas de probabilidade desse modelo saturaram frequentemente em patamares elevados devido à simples presença de um indivíduo manipulando os produtos, evidenciando a dificuldade da rede de distinguir entre interações legítimas e atos intencionais de ocultação.

- **Vulnerabilidade Compartilhada a Oclusões Severas:** As duas arquiteturas falharam na detecção em cenários marcados por oclusões severas e ações corporais de baixa magnitude visual. Quando a ação era demasiadamente ocluída, os modelos eram inibidos de extrair as características espaço-temporais indispensáveis, mantendo as probabilidades de alarme significativamente abaixo do necessário para discriminação do ato.

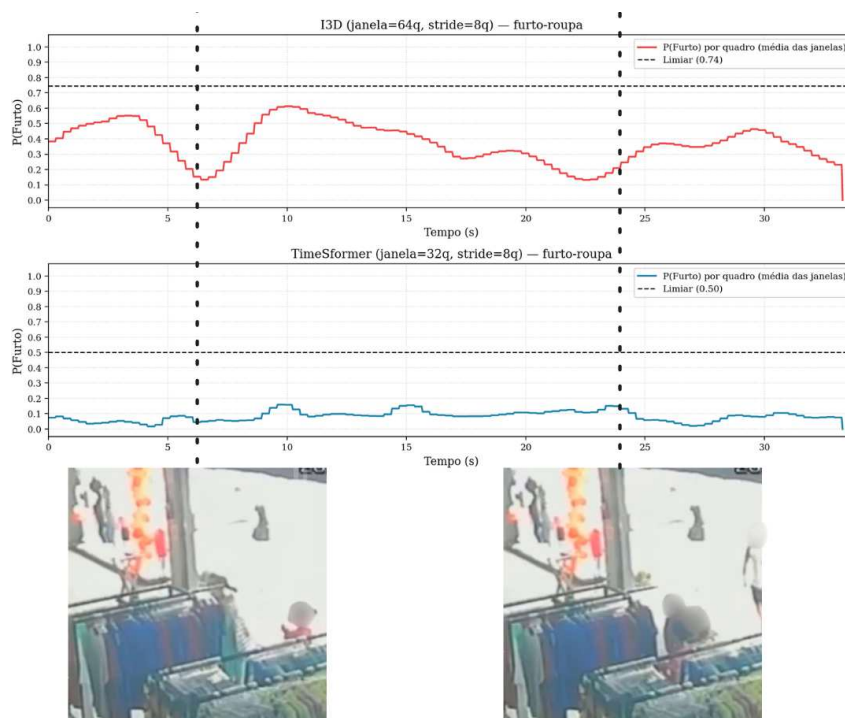
Figura 21 – Curva de probabilidade do modelo I3D para o vídeo *furto-celular*.

Fonte - Do autor.

Figura 22 – Curva de probabilidade do modelo I3D para o vídeo *furto-pc*.

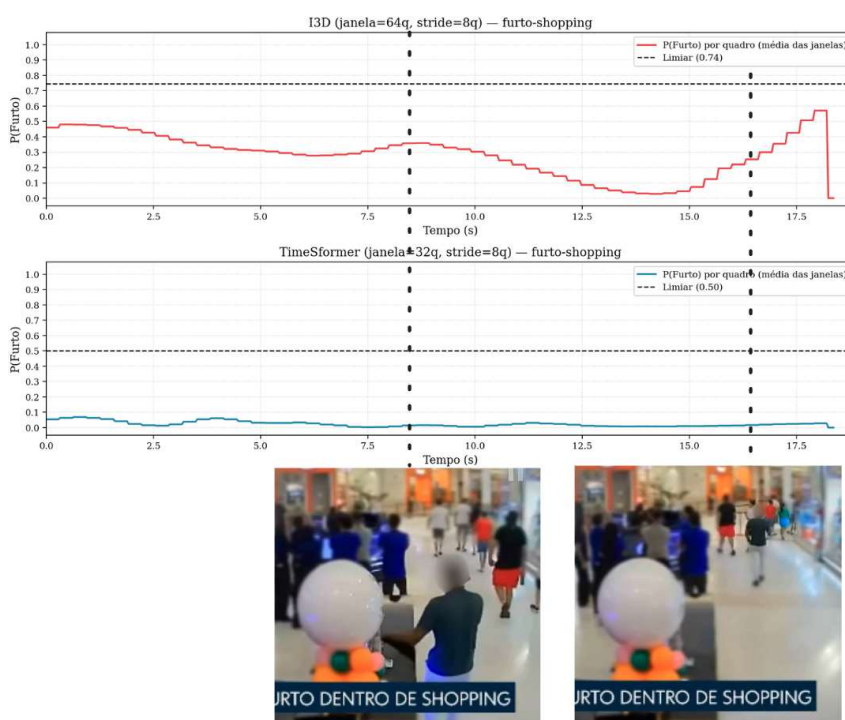
Fonte - Do autor.

Figura 23 – Curva de probabilidade dos modelo I3D e TimeSFormer para o vídeo *furto-roupa*.

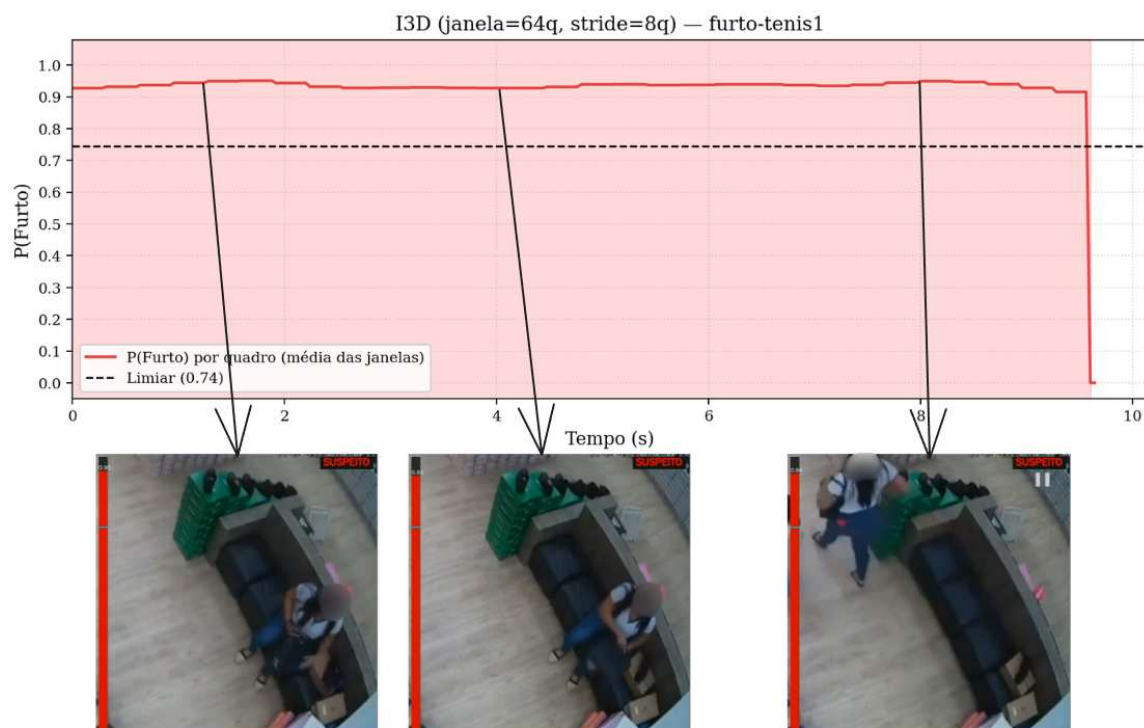


Fonte - Do autor.

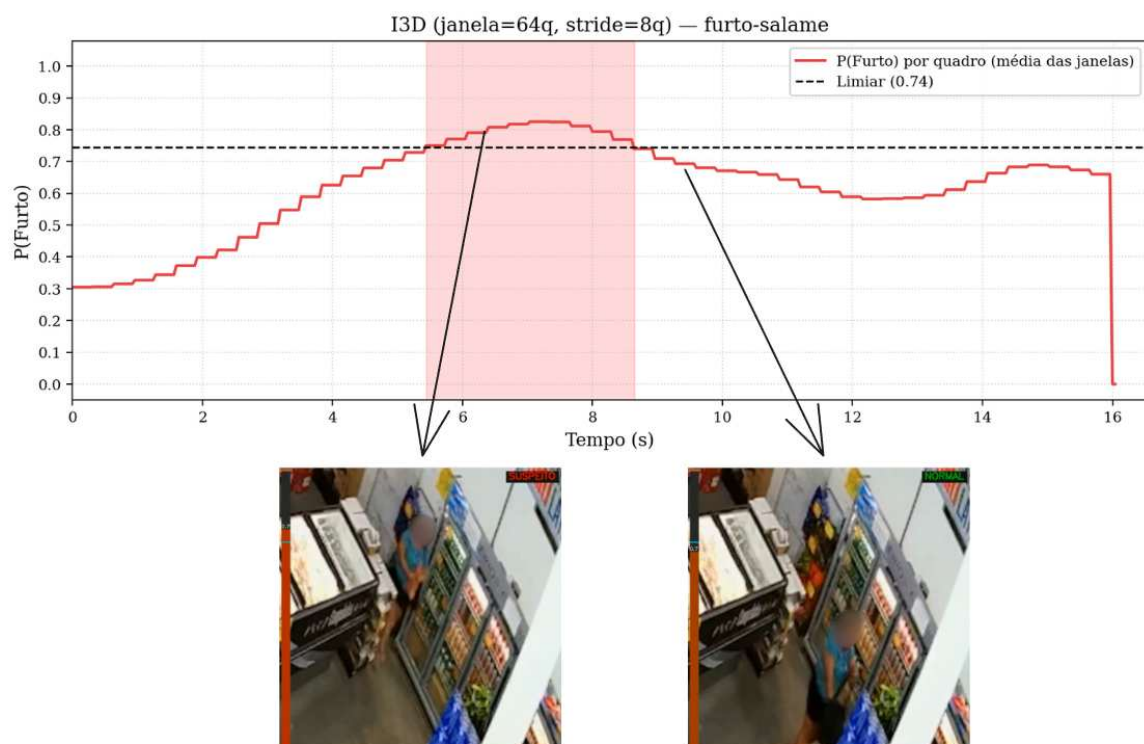
Figura 24 – Curva de probabilidade dos modelo I3D e TimeSFormer para o vídeo *furto-shopping*.



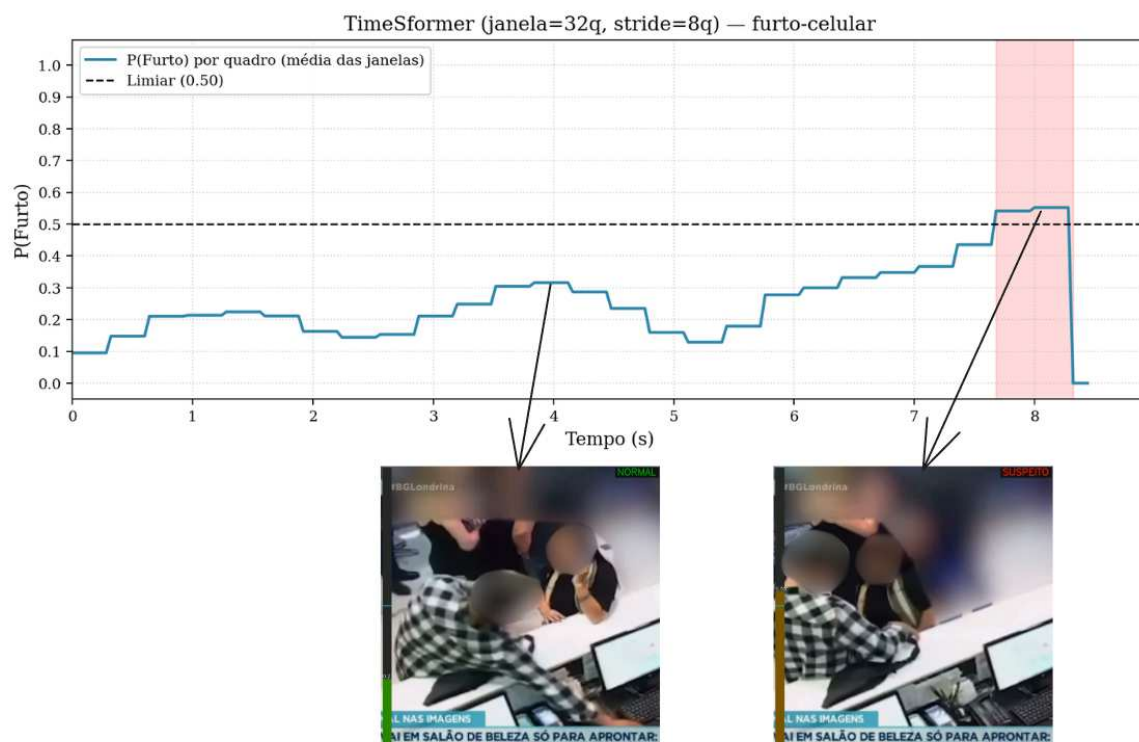
Fonte - Do autor.

Figura 25 – Curva de probabilidade do modelo I3D para o vídeo *furto-tenis1*.

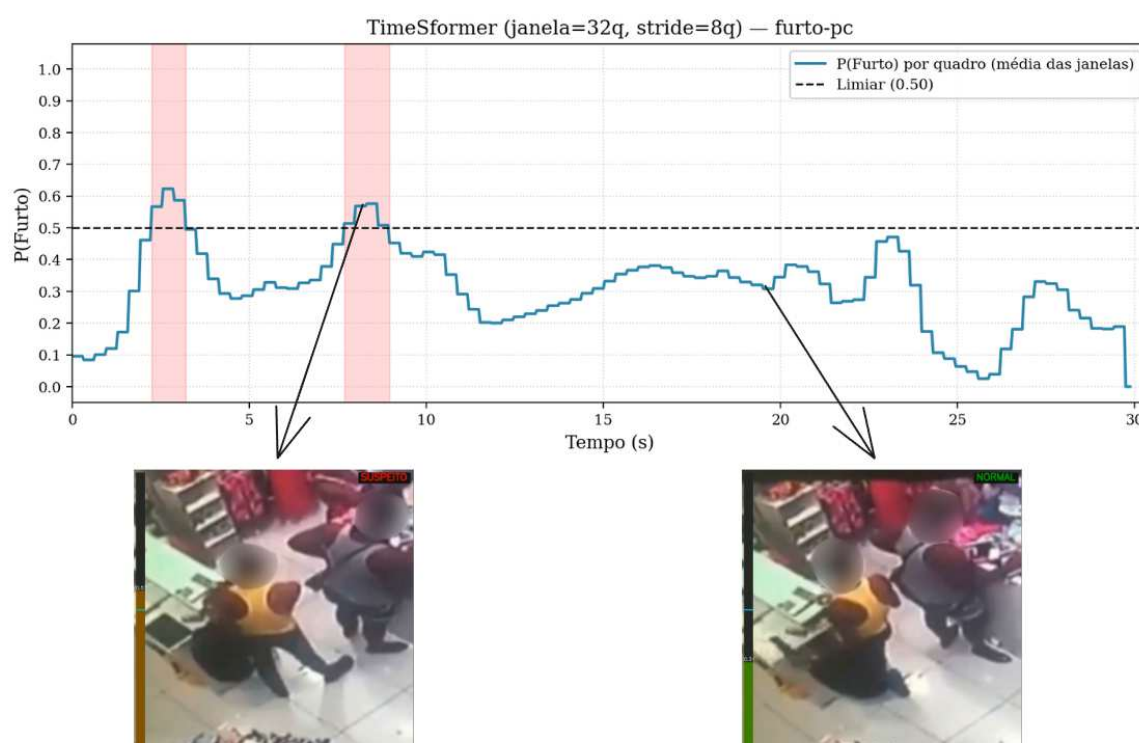
Fonte - Do autor.

Figura 26 – Curva de probabilidade do modelo I3D para o vídeo *furto-supermercado*.

Fonte - Do autor.

Figura 27 – Curva de probabilidade do modelo TimeSFormer para o vídeo *furto-celular*.

Fonte - Do autor.

Figura 28 – Curva de probabilidade do modelo TimeSFormer para o vídeo *furto-pc*.

Fonte - Do autor.

6 Avaliação de Implementação em Cenário Real

Para estruturar esta avaliação de viabilidade com rigor técnico e financeiro, a análise é decomposta em etapas fundamentais. Inicialmente, realiza-se o levantamento do panorama financeiro atual e do histórico de perdas no varejo alimentar brasileiro, baseando-se em indicativos recentes do setor. Sob a caracterização física de um supermercado de médio porte padronizado, executa-se o dimensionamento da rede de câmeras IP necessária para a cobertura espacial abrangente do ambiente. Posteriormente, elabora-se a modelagem matemática da carga computacional contínua gerada pela inferência, transpondo os resultados experimentais para um dimensionamento de servidores dotados de GPUs. Por fim, realiza-se a projeção percentual e monetária de perdas evitáveis contraposta ao custo total de aquisição e operação do hardware, obtendo a estimativa do período de recuperação do capital.

6.1 Panorama de Perdas no Varejo Alimentar Brasileiro

Considerando um supermercado de médio porte com faturamento anual próximo de R\$ 90 milhões, observa-se um volume médio de perdas de aproximadamente R\$ 57.875 por mês, equivalente a cerca de R\$ 694 mil por ano, ou 0,76% do faturamento anual (AIRES; SANTOS; FERNANDES, 2026).

Embora as perdas de inventário englobem diversas categorias, como avarias, vencimentos, erros operacionais e divergências de estoque, os furtos internos e externos figuram entre as principais causas registradas pelo setor supermercadista, sendo responsáveis por aproximadamente 20% a 35% do total de perdas (Associação Brasileira de Supermercados, 2019).

Aplicando essa proporção ao cenário considerado, estima-se que as perdas anuais associadas a furtos situem-se entre R\$ 139 mil e R\$ 243 mil. Em termos relativos ao faturamento do estabelecimento, esse montante corresponde a aproximadamente 0,15% a 0,27% da receita anual.

Essa faixa de valores será utilizada nas próximas seções como referência para estimar o montante financeiro potencialmente recuperável por meio da redução de ocorrências de furto.

6.2 Dimensionamento da Infraestrutura de Captação

Para estimar os requisitos computacionais de uma solução de detecção automática de furtos baseada em vídeo, é necessário inicialmente estabelecer uma infraestrutura de captação compatível com a cobertura integral do ambiente monitorado. Nesta análise, considera-se um supermercado de médio porte com área de vendas aproximada de 1000 m².

Diferentemente de aplicações convencionais de vigilância patrimonial, sistemas voltados à análise comportamental dependem da observação contínua das interações entre clientes e produtos ao longo dos corredores de circulação. Em ambientes supermercadistas, a presença de gôndolas e prateleiras com alturas elevadas reduz significativamente o campo de visão efetivo das câmeras e cria regiões suscetíveis a oclusões. Dessa forma, a cobertura do ambiente demanda não apenas a observação dos acessos principais, mas também o monitoramento dos corredores de venda e das áreas de maior concentração de mercadorias.

Para fins de estimativa, adota-se uma densidade de implantação correspondente a aproximadamente uma câmera para cada 15 a 20 m² de área monitorada. Essa faixa é compatível com projetos de CFTV que buscam cobertura abrangente dos corredores e sobreposição parcial entre campos de visão adjacentes.

Considerando uma área total de 1000 m², obtém-se:

$$N_{cam} = \frac{A}{A_{cam}} \quad (6.1)$$

onde A representa a área monitorada e A_{cam} a área efetivamente coberta por cada câmera. Assumindo uma cobertura média de aproximadamente 16 m² por dispositivo, resulta:

$$N_{cam} = \frac{1000}{16} \approx 62,5 \quad (6.2)$$

Arredondando para um valor inteiro e considerando áreas adicionais como entradas, saídas, região de caixas e setores de maior risco operacional, adota-se como referência uma infraestrutura composta por aproximadamente 64 câmeras IP.

Uma vez estabelecida a infraestrutura de captação, o problema pode ser reinterpretado sob a perspectiva computacional. Cada uma das 64 câmeras IP instaladas produz um fluxo de vídeo contínuo e independente, potencialmente sujeito à análise automatizada por algoritmos de detecção de furtos.

Dessa forma, a carga computacional total do sistema não depende apenas das características individuais dos modelos de inteligência artificial avaliados neste trabalho,

mas também da quantidade de fluxos simultâneos que devem ser processados. Em outras palavras, a infraestrutura de inferência deve ser capaz de executar continuamente o processamento de aproximadamente 64 canais de vídeo em paralelo.

Consequentemente, o dimensionamento computacional pode ser obtido a partir da capacidade de processamento por fluxo observada experimentalmente para o modelo TimeSFormer, extrapolando-se esse resultado para o conjunto completo de câmeras necessário para cobertura do ambiente monitorado. A próxima seção apresenta essa modelagem e a conversão da demanda de inferência em requisitos de hardware.

6.3 Dimensionamento da Demanda de Inferência e Custos

Uma vez estabelecida a necessidade de monitoramento de aproximadamente 64 câmeras IP para cobertura integral de um supermercado de médio porte, torna-se possível converter a demanda operacional do sistema em requisitos concretos de processamento computacional. Conforme discutido na seção 4.8, o melhor modelo obtido neste trabalho foi o TimeSFormer pré-treinado no banco de dados SSv2 com entrada de 32 quadros e ajuste completo dos parâmetros, alcançando o maior F2-Score no conjunto de teste, considerando que o objetivo desta análise consiste em dimensionar a infraestrutura necessária para o melhor modelo obtido, adota-se esse TimeSFormer como referência.

Os experimentos de eficiência, na seção 4.9, demonstraram que o TimeSFormer executa aproximadamente 1,35 inferências por segundo em uma GPU GTX 1660 Super, correspondendo a um tempo médio de 738,53 ms para processar uma única janela de vídeo composta por 32 quadros.

A GTX 1660 Super utilizada nos experimentos possui desempenho teórico próximo de 5 TFLOPS em operações FP32. Dessa forma, pode-se estimar a demanda computacional associada a uma única inferência do modelo através da relação entre o desempenho disponível e o tempo efetivamente consumido durante a execução:

$$C_{inf} = P \cdot t \quad (6.3)$$

onde P representa a capacidade computacional da GPU (em TFLOPS) e t o tempo médio de inferência (em segundos).

Substituindo os valores experimentais:

$$C_{inf} = 5 \times 0,73853 \quad (6.4)$$

$$C_{inf} \approx 3,69 \text{ TFLOP} \quad (6.5)$$

Portanto, cada janela processada pelo TimeSFormer demanda aproximadamente 3,69 TFLOP de processamento.

Assumindo o cenário conservador em que as 64 câmeras instaladas estejam simultaneamente produzindo atividade relevante e necessitem ser analisadas continuamente, a carga computacional total torna-se:

$$C_{total} = 64 \times 3,69 \quad (6.6)$$

$$C_{total} \approx 236 \text{ TFLOP} \quad (6.7)$$

Este valor representa a capacidade computacional necessária para sustentar o processamento simultâneo dos fluxos de vídeo no mesmo intervalo temporal observado experimentalmente.

Atualmente, aceleradores de inferência voltados à inteligência artificial disponíveis comercialmente apresentam desempenho significativamente superior ao hardware utilizado durante os experimentos. Como referência, uma GPU NVIDIA RTX 5090 disponibiliza aproximadamente 105 TFLOPS em precisão FP32, enquanto uma RTX 5080 disponibiliza cerca de 56 TFLOPS.

Dessa forma, o processamento completo da loja poderia ser atendido por diferentes configurações equivalentes de hardware, tais como:

- 3 GPUs RTX 5090 ($3 \times 105 \approx 315$ TFLOPS);
- 5 GPUs RTX 5080 ($5 \times 56 \approx 280$ TFLOPS);
- infraestrutura distribuída composta por múltiplos servidores menores.

Sob a perspectiva econômica, considerando valores praticados no mercado brasileiro durante 2026, uma RTX 5090 possui custo aproximado entre R\$ 18.000 e R\$ 22.000. Assim, uma infraestrutura baseada em três placas totalizaria investimento próximo de R\$ 60.000, desconsiderando os demais componentes do servidor.

Incluindo processador, memória, armazenamento, sistema de alimentação redundante e margem para expansão futura, o custo total de implantação da infraestrutura de inferência situa-se tipicamente entre R\$ 70.000 e R\$ 90.000 para uma unidade varejista de aproximadamente 1000 m².

É importante destacar que esta estimativa representa um cenário deliberadamente conservador, no qual todas as câmeras são processadas continuamente e de forma independente. Na prática, mecanismos de otimização podem reduzir substancialmente a demanda

computacional. Entre eles destacam-se o processamento em lote, a redução da frequência de inferência em áreas de baixo risco, a utilização de detectores de movimento para ativação seletiva do modelo e a execução distribuída em múltiplos períodos temporais. Consequentemente, o custo real de implantação tende a ser inferior ao valor estimado nesta análise.

6.4 Estimativa de Redução de Perdas e Retorno Sobre o Investimento

A etapa final desta análise consiste em converter os resultados de desempenho obtidos experimentalmente em uma estimativa econômica para um cenário real de operação.

Conforme discutido no Capítulo 1, estudos sobre monitoramento por CFTV indicam que operadores humanos podem falhar na identificação de até 66% dos eventos críticos presentes em seu campo de visão. Em outras palavras, mesmo em ambientes monitorados, apenas aproximadamente 34% das ocorrências potencialmente observáveis tendem a ser efetivamente identificadas em tempo real.

Por sua vez, o melhor modelo obtido neste trabalho, o TimeSFormer pré-treinado no banco de dados SSv2 com entrada de 32 quadros e ajuste completo dos parâmetros, alcançou revocação de 91,7% no conjunto de teste. Como a revocação representa a proporção de furtos corretamente identificados pelo sistema, este valor pode ser interpretado como a capacidade potencial de detecção automática do modelo.

Assumindo que a taxa de falha humana e a taxa de revocação do modelo sejam representativas do ambiente operacional, a diferença entre ambos corresponde ao percentual adicional de ocorrências passíveis de identificação:

$$\Delta D = 91,7\% - 34,0\% \quad (6.8)$$

$$\Delta D = 57,7\% \quad (6.9)$$

Assim, o sistema proposto possui potencial para identificar 57,7% mais ocorrências de furto do que o monitoramento exclusivamente humano.

Na seção 6.1, foi estimado que um supermercado de médio porte apresenta perdas anuais associadas a furtos entre R\$139 mil e R\$243 mil. Considerando que a redução de perdas seja proporcional ao aumento da capacidade de detecção, obtém-se uma estimativa conservadora do valor recuperável:

$$G_{min} = 0,577 \times 139.000 \quad (6.10)$$

$$G_{min} \approx R\$80.200/ano \quad (6.11)$$

e

$$G_{max} = 0,577 \times 243.000 \quad (6.12)$$

$$G_{max} \approx R\$140.200/ano \quad (6.13)$$

Portanto, a implantação do sistema possui potencial para reduzir perdas financeiras em aproximadamente R\$80 mil a R\$140 mil por ano para uma unidade varejista de porte médio.

Na seção 6.3, estimou-se um investimento de implantação entre R\$70 mil e R\$90 mil para a infraestrutura de inferência necessária ao processamento das câmeras do estabelecimento. Dessa forma, o período de recuperação do capital investido pode ser estimado por:

$$Payback = \frac{Investimento}{Ganho\ Anual} \quad (6.14)$$

resultando em:

$$Payback_{conservador} = \frac{90.000}{80.200} \approx 1,12 \text{ anos} \quad (6.15)$$

e

$$Payback_{otimista} = \frac{70.000}{140.200} \approx 0,50 \text{ anos} \quad (6.16)$$

Portanto, mesmo considerando hipóteses conservadoras para a infraestrutura computacional e para a efetividade operacional do sistema, observa-se que o investimento apresenta potencial de retorno em um horizonte entre **seis e quatorze meses**. Além do benefício financeiro direto associado à redução de perdas, a solução proposta ainda oferece vantagens adicionais não quantificadas nesta análise, como aumento da capacidade de monitoramento, padronização da vigilância, redução da fadiga operacional e geração automática de alertas em tempo real para as equipes de prevenção de perdas.

7 Conclusões

O presente trabalho abordou um problema crônico para o setor varejista global: a necessidade de identificação automatizada de comportamentos suspeitos e furtos, reduzindo a dependência quase exclusiva e limitante da vigilância humana em CFTVs. Neste contexto, o objetivo principal da pesquisa foi executado comparando empiricamente o desempenho de duas arquiteturas distintas de aprendizado profundo na tarefa de modelagem e detecção de anomalias espaço-temporais em vídeos: a RNC 3D (I3D) e o Transformador de Visão (TimeSFormer). Para viabilizar este estudo, o trabalho apresentou a contribuição metodológica de curadoria e de construção de um banco de dados próprio, agregando, filtrando e segmentando amostras das bases públicas DCSASS, Shoplifting Dataset 2.0 e Shoplifting MNNIT.

Através da formulação de dezesseis configurações experimentais, a pesquisa avaliou o impacto de diferentes estratégias de ajuste de pesos, uso de múltiplas modalidades de entrada e diferentes resoluções temporais. A síntese dos resultados obtidos durante o treinamento respondeu ao problema de pesquisa, constatando-se primeiramente que o ajuste profundo de todos os pesos da rede é estritamente essencial para a convergência e o aprendizado eficaz das dinâmicas de furto em ambas as arquiteturas em bancos de dados reduzidos como o construído. Experimentos que limitaram o treinamento apenas à camada classificadora final falharam expressivamente em adaptar as representações visuais ao novo domínio, estacionando em altos patamares de erro mesmo após dezenas de épocas de treinamento. Além disso, demonstrou-se que a inclusão de imagens de fluxo óptico no modelo I3D atua como um mecanismo compensatório relevante apenas quando o extrator de características original é mantido congelado, tornando-se redundante quando a RNC é ajustada por completo. A avaliação definitiva sobre o conjunto de teste confirmou a superioridade dos mecanismos de atenção global para a compreensão semântica da cena, evidenciando uma importante inversão no ranqueamento de desempenho em relação à fase de validação. A arquitetura TimeSFormer destacou-se por sua notável capacidade de generalização para dados inéditos, alcançando 100% de precisão, o que denota um controle superior na redução de alarmes falsos, além de obter o melhor F2-Score geral (0,932). Por sua vez, o modelo convolucional I3D exibiu um comportamento preditivo marcadamente conservador, com uma precisão consideravelmente menor (72,3%), refletindo uma forte tendência em classificar interações normais como ocorrências anômalas.

Essa diferença de comportamentos foi reiterada e aprofundada na etapa de inferência baseada em janela deslizante, que simulou o monitoramento de vídeos reais e contínuos. Nesse cenário extremo, o TimeSFormer provou ser mais seletivo, obtendo êxito ao elevar as probabilidades de anomalia de maneira sincronizada ao exato instante da

ocultação dos produtos. Em contrapartida, embora o I3D tenha confirmado sua capacidade de detectar ações abruptas com precisão temporal, o modelo falhou criticamente ao apresentar saturações e falsos positivos contínuos gerados por simples manipulações legítimas de itens, evidenciando a dificuldade da arquitetura convolucional em distinguir a observação normal da intenção ilícita.

Sob a ótica da aplicação mercadológica, o trabalho mapeia o compromisso técnico necessário para a implementação desses sistemas na indústria varejista. Verificou-se que a obtenção da precisão adicional e da seletividade proporcionada pelo TimeSFormer exige um investimento computacional substancialmente maior, com tempo de inferência de 738,53 milissegundos por janela. Já o I3D destaca-se pela alta eficiência e baixa latência (79,52 milissegundos), configurando-se como uma alternativa viável e escalável para ambientes com severas restrições de infraestrutura, desde que sua sensibilidade aos falsos positivos possa ser contornada por lógicas de limiares mais estritas.

Apesar dos resultados, a pesquisa identifica limitações inerentes aos modelos avaliados. Ambas as arquiteturas compartilham uma vulnerabilidade crítica: a falha completa de detecção diante de oclusões severas na cena (causadas por estantes ou aglomerações) e de movimentos corporais propositalmente sutis. Nestas condições, a ocultação dos membros ou do objeto inibe fortemente a extração das características espaço-temporais essenciais, fazendo com que as probabilidades de comportamento suspeito permaneçam abaixo do limiar de alarme necessário para a discriminação do ato. Além disso, o TimeSFormer demonstrou alguns episódios de latência no alarme e, sob certos arranjos visuais, uma tendência a fixar seu mecanismo de atenção global em elementos contextuais irrelevantes, perdendo sensibilidade para as ações suspeitas.

Como sugestões para estudos futuros, recomenda-se a expansão do banco de dados com filmagens locais que capturem a grande diversidade de iluminação, densidade de clientes e arquiteturas de prateleiras características do varejo brasileiro. Na área experimental, sugere-se a realização de investigações voltadas à otimização fina do treinamento por meio da busca sistemática por melhores hiperparâmetros, com foco especial na calibração da taxa de aprendizado e no seu método de decaimento. Para conferir maior robustez estatística aos resultados e garantir a estabilidade da convergência, recomenda-se a implementação de estratégias de validação cruzada (K-fold) ou métodos de busca em grade, permitindo uma exploração mais granular do espaço de parâmetros. Em termos arquiteturais, a incorporação de abordagens multimodais, em especial a integração com Redes Convolucionais baseadas em estimativa de pose esquelética, representa um caminho promissor para manter a coerência do rastreamento corporal mesmo quando parte do indivíduo está oclusa.

Referências

- AIRES, M. de F.; SANTOS, R. G. L. dos; FERNANDES, T. de O. Controle interno e prevenção de perda: Um estudo de caso em um supermercado que implantou o setor de prevenção de perdas. **Revista REGEO**, v. 17, n. 2, p. 1–13, 2026.
- ANSARI, M. A.; SINGH, D. K. **Shoplifting Dataset (2022) - CV Laboratory MNNIT Allahabad**. Mendeley Data, 2023. Acesso em: 25 mar. 2026. Disponível em: <https://data.mendeley.com/datasets/r3yjf35hzh/1>.
- ARNAB, A.; DEHGHANI, M.; HEIGOLD, G.; SUN, C.; LUČIĆ, M.; SCHMID, C. ViViT: A video vision transformer. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. Montreal, Canada: IEEE, 2021. p. 6836–6846.
- Associação Brasileira de Prevenção de Perdas; KPMG. **Pesquisa Abrappe de Perdas no Varejo Brasileiro 2024**. São Paulo, Brasil, 2024. Disponível em: <https://kpmg.com/br/pt/insights/2024/11/pesquisa-abrappe-2024.html>.
- Associação Brasileira de Supermercados. **19ª Avaliação de Perdas no Varejo Brasileiro de Supermercados**. 2019. Acesso em: 06 jun. 2026. Disponível em: http://static.abras.com.br/pdf/apresentacao_perdas_2019.pdf.
- BERTASIUS, G.; WANG, H.; TORRESANI, L. Is Space-Time attention all you need for video understanding? In: **Proceedings of the 38th International Conference on Machine Learning (ICML)**. Virtual Event: PMLR, 2021. (Proceedings of Machine Learning Research, v. 139), p. 813–824.
- CARREIRA, J.; ZISSERMAN, A. Quo vadis, action recognition? a new model and the Kinetics dataset. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, HI, USA: IEEE, 2017. p. 6299–6308.
- CASTRO, C. L. de; BRAGA, A. d. P. Aprendizado supervisionado com conjuntos de dados desbalanceados. **SBA: Controle & Automação**, Sociedade Brasileira de Automática, Campinas, SP, Brasil, v. 22, n. 5, p. 441–466, 2011.
- CHEN, L.-P. Supervised learning for binary classification on us adult income. **Journal of Modeling and Optimization**, v. 13, n. 2, p. 80–91, 2021.
- CHOLLET, F. **Deep Learning with Python**. Shelter Island, NY, USA: Manning Publications, 2021.
- CORDEIRO, G. A.; GROCKOTZKI, G.; AZEVEDO, I. J. de; MANTOVANI, J.; SANTOS, M. H. d. S.; PONCIO, R. V.; MENDES, C. M. Detecção de roubo de computadores em laboratório usando visão computacional. In: **Anais do Computer on the Beach**. Balneário Camboriú, SC, Brasil: Universidade do Vale do Itajaí, 2020. p. 54–55.
- COSTA, G. d. B. P. d. **Representation learning of spatio-temporal features from video**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), São Carlos, SP, Brasil, 2019.

- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A large-scale hierarchical image database. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Miami, FL, USA: IEEE, 2009. p. 248–255.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016.
- GOYAL, R.; KAHOU, S. E.; MICHALSKI, V.; MATERZYNSKA, J.; WESTPHAL, S.; KIM, H.; HAENEL, V.; FRUEND, I.; YIANILOS, P.; MUELLER-FREITAG, M.; HOPPE, F.; THURAU, C.; BAX, I.; MEMISEVIC, R. The “Something Something” video database for learning and evaluating visual common sense. In: **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. Venice, Italy: IEEE, 2017. p. 5842–5850.
- GRANDINI, M.; BAGLI, E.; VISANI, G. **Metrics for Multi-Class Classification: an Overview**. 2020.
- GROSSBERGER, K. J. Retail security: Use of security guards. In: **Encyclopedia of Security and Emergency Management**. Cham, Switzerland: Springer, 2021. p. 839–846.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Second. New York, NY, USA: Springer, 2009.
- HERVAS, M. **DCSASS Dataset**. Kaggle, 2019. Acesso em: 25 mar. 2026. Disponível em: <https://www.kaggle.com/datasets/mateohervas/dcsass-dataset>.
- HISHAM, M. **Shoplifting dataset 2.0 (4 labels)**. Kaggle, 2024. Acesso em: 25 mar. 2026. Disponível em: <https://www.kaggle.com/datasets/mohamedhisham20/shoplifting-dataset-2-0-4-labels>.
- HUANG, D.-A.; RAMANATHAN, V.; MAHAJAN, D.; TORRESANI, L.; PALURI, M.; FEI-FEI, L.; NIEBLES, J. C. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, UT, USA: IEEE, 2018. p. 7366–7375.
- HUANG, Y.; GUO, Y.; GAO, C. Efficient parallel inflated 3d convolution architecture for action recognition. **IEEE Access**, IEEE, v. 8, p. 45753–45765, 2020.
- infoDiagram. **AUC ROC Curve - Classifier Model Quality Measure**. 2024. Acesso em: 01 abr. 2026. Disponível em: <https://www.infodiagram.com/slides/auc-roc-curve-classifier-model-quality-measure>.
- JOHNSON, J. M.; KHOSHGOFTAAR, T. M. Survey on deep learning with class imbalance. **Journal of big data**, Springer, v. 6, n. 1, p. 27, 2019.
- KAY, W.; CARREIRA, J.; SIMONYAN, K.; ZHANG, B.; HILLIER, C.; VIJAYANARASIMHAN, S.; VIOLA, F.; GREEN, T.; BACK, T.; NATSEV, P. et al. **The Kinetics Human Action Video Dataset**. London, UK, 2017.

- LENG, C.; DING, Q.; WU, C.; CHEN, A.; WANG, H.; WU, H. Bdnet: a method based on forward and backward convolutional networks for action recognition in videos. **The Visual Computer**, v. 40, p. 1–15, 10 2023.
- MAURÍCIO, J.; DOMINGUES, I.; BERNARDINO, J. Comparing vision transformers and convolutional neural networks for image classification: A literature review. **Applied Sciences**, MDPI, v. 13, n. 9, p. 5521, 2023.
- MITCHELL, T. M. **Machine Learning**. New York, NY, USA: McGraw-Hill, 1997.
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of Machine Learning**. Cambridge, MA, USA: MIT Press, 2012. (Adaptive Computation and Machine Learning series).
- MUNEER, I.; SADDIQUE, M.; HABIB, Z.; MOHAMED, H. G. Shoplifting detection using hybrid neural network cnn-bilsmt and development of benchmark dataset. **Applied Sciences**, MDPI, v. 13, n. 14, p. 8341, 2023.
- NÄSHOLM, E.; ROHLFING, S.; SAUER, J. D. Pirate stealth or inattentional blindness? the effects of target relevance and sustained attention on security monitoring for experienced and naïve operators. **PLoS One**, Public Library of Science San Francisco, USA, v. 9, n. 1, p. e86157, 2014.
- National Retail Federation; Loss Prevention Research Council. **The Impact of Retail Theft & Violence**. 2024.
- PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 22, n. 10, p. 1345–1359, 2009.
- POWERS, D. M. W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. **Journal of Machine Learning Technologies**, International Science Congress Association, Indore, India, v. 2, n. 1, p. 37–63, 2011.
- PRABHAKAR, C. J.; SHIVASHANKAR, G. S.; MOHANA, S. H. ViTHAR: Vision transformers for human action recognition in videos. In: **Proceedings of the 4th International Conference on Multimedia Processing, Communication & Information Technology (MPCIT)**. Shivamogga, India: IEEE, 2024. p. 358–363.
- RAHANGDALE, K.; KOKATE, M. Event detection using background subtraction for surveillance systems. **International Research Journal of Engineering and Technology**, v. 3, n. 01, p. 1300–1304, 2016.
- RASHVAND, N.; NOGHRE, G. A.; PAZHO, A. D.; YAO, S.; TABKHI, H. Exploring pose-based anomaly detection for retail security: A real-world shoplifting dataset and benchmark. In: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)**. Tucson, AZ, USA: IEEE, 2025. p. 1123–1131.
- RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3rd. ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

- SCHMUCKER, P. **Evaluating defense mechanisms against universal adversarial perturbations for convolutional neural networks**. Tese (Doutorado) — Ruhr University Bochum, 2024.
- SIMONS, D. J.; CHABRIS, C. F. Gorillas in our midst: Sustained inattention blindness for dynamic events. **perception**, SAGE Publications Sage UK: London, England, v. 28, n. 9, p. 1059–1074, 1999.
- SULTANI, W.; CHEN, C.; SHAH, M. Real-world anomaly detection in surveillance videos. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, UT, USA: IEEE, 2018. p. 6479–6488.
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. Second. Cham, Switzerland: Springer Nature, 2022.
- TAJBAKHSH, N.; SHIN, J. Y.; GURUDU, S. R.; HURST, R. T.; KENDALL, C. B.; LAWRENCE, M. B.; TAJBAKHSH, A.; LIANG, J. Convolutional neural networks for medical image analysis: Full training or fine tuning? **IEEE transactions on medical imaging**, IEEE, v. 35, n. 5, p. 1299–1312, 2016.
- TAKAHASHI, S.; SAKAGUCHI, Y.; KOUNO, N.; TAKASAWA, K.; ISHIZU, K.; AKAGI, Y.; AOYAMA, R.; TERAYA, N.; BOLATKAN, A.; SHINKAI, N. et al. Comparison of vision transformers and convolutional neural networks in medical image analysis: a systematic review. **Journal of Medical Systems**, Springer, v. 48, n. 1, p. 84, 2024.
- TRAN, D.; WANG, H.; TORRESANI, L.; RAY, J.; LECUN, Y.; PALURI, M. A closer look at spatiotemporal convolutions for action recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, UT, USA: IEEE, 2018. p. 6450–6459.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.
- YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? In: **Advances in Neural Information Processing Systems (NIPS)**. Montreal, QC, Canada: Curran Associates, Inc., 2014. v. 27, p. 3320–3328.
- ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. **Dive into Deep Learning**. Cambridge, UK: Cambridge University Press, 2023.

Apêndices

APÊNDICE A – Repositórios do Projeto

O código-fonte, os experimentos e a documentação desenvolvidos neste trabalho encontram-se publicamente disponíveis no GitHub, organizados em três repositórios distintos, cada um responsável por um escopo bem delimitado do projeto.

Organização dos Repositórios

- **shoplifting-detection**: *monorepo* principal do projeto. Contém os *scripts* centralizados de download e pré-processamento do banco de dados, os *pipelines* de treinamento e validação de ambas as arquiteturas, as configurações dos experimentos via `config.yaml` e os artefatos gerados (pesos dos modelos e registros de métricas). Inclui os subpacotes `i3d-shoplifting` e `timesformer-shoplifting` como dependências locais.
- **i3d-shoplifting**: subpacote responsável pelo *fine-tuning* do modelo I3D como classificador binário. Implementa a arquitetura I3D em PyTorch puro, os módulos de pré-processamento para extração de quadros RGB e geração de fluxo óptico TV-L1, e o laço de treinamento com `BCEWithLogitsLoss` e ponderação de classes.
- **timesformer-shoplifting**: subpacote responsável pelo *fine-tuning* do modelo TimeSFormer como classificador binário. Implementa o *dataset* PyTorch com amostragem temporal via Decord, os utilitários de padronização de vídeo com FFmpeg e o treinamento por meio do `Trainer` da biblioteca *Transformers* com `CEL` e ponderação de classes.

Gerenciamento de Dependências com UV

Todos os repositórios adotam a ferramenta **UV** para o gerenciamento do ambiente Python. O UV é um gerenciador de pacotes e ambientes virtuais de alto desempenho que resolve e instala dependências de forma significativamente mais rápida que as ferramentas tradicionais. A configuração completa do ambiente de cada repositório é declarada no arquivo `pyproject.toml`, permitindo a reprodução integral do ambiente de desenvolvimento com um único comando.