Análise e Mitigação do Cross-Dataset Drift em Sistemas de Reconhecimento Facial em Tempo Real com Modelos Otimizados

Turma U – Equipe 2
Alexandre C. A. Beiruth, João G. Cunha, Leandro C. Vieira, Luana T. H. Cordeiro
{ alexandre.beiruth, joao.cunha, cardoso.vieira, luana.halicki }@pucpr.edu.br

I. Descrição do Projeto

O Reconhecimento de Expressões Faciais (REF) consolidou-se como uma área de pesquisa fundamental na interseção entre visão computacional e interação humano-computador, com aplicações que vão desde sistemas de recomendação até diagnósticos de saúde mental. Modelos de *deep learning*, como ResNet50 e EfficientNet, e modelos de *Vision Transformers*, como EfficientVit, +– alcançaram alta acurácia em datasets de referência, como FER-2013, RAF-DB e DFEW –+. No entanto, o desempenho desses modelos frequentemente se degrada ao serem transpostos de ambientes controlados de laboratório para aplicações do mundo real, um desafio conhecido como *Cross-Dataset Drift* (CDD). Este fenômeno ocorre devido às variações não representadas nos dados de treinamento, como iluminação diversa, oclusões parciais e a vasta heterogeneidade dos indivíduos, sendo um obstáculo crítico para a implementação de sistemas de REF robustos e confiáveis.

O problema central deste trabalho investiga a quantificação do impacto do CDD quando modelos de REF são aplicados a um domínio de tempo real, utilizando dados capturados continuamente por uma webcam. A principal pergunta de pesquisa que norteia este projeto é: "Qual a degradação de desempenho de um modelo treinado nos datasets RAF-DB, FER-2013 e DFEW quando aplicado a um novo domínio de inferência em tempo real?". A relevância desta investigação reside na necessidade de desenvolver soluções que não apenas sejam precisas, mas também computacionalmente eficientes e capazes de generalizar para condições de uso não vistas, viabilizando sua aplicação prática em dispositivos com recursos limitados.

Os objetivos principais do projeto são: (1) Avaliar e comparar o desempenho de arquiteturas de *deep learning* (ResNet50 e EfficientNet) e *Vision Transformers* (EfficientVit) em termos de métricas de desempenho e custo computacional para a tarefa de REF em tempo real; (2) Desenvolver uma arquitetura de aplicação completa, desde a captura e pré-processamento de faces até a inferência; (3) Medir quantitativamente a perda de performance causada pelo CDD ao transitar dos datasets de validação para o domínio em tempo real; e (4) Investigar o uso de técnicas de otimização pós-treinamento, como a quantização, para mitigar a sobrecarga computacional sem comprometer significativamente as métricas obtidas. A Figura 1 representa todo o contexto da problemática que será implementada neste artigo.

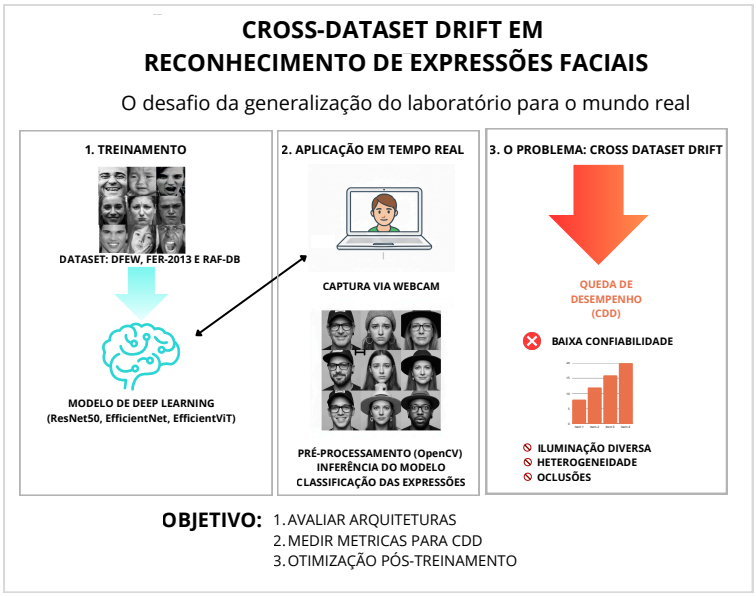

Figura 1. Fluxo do desafio, tema, métricas e objetivos.

Para alcançar tais objetivos, a abordagem proposta envolve o treinamento dos modelos nos datasets de referência e, em seguida, a implementação de um pipeline de inferência em tempo real. Este pipeline utilizará a biblioteca OpenCV para detecção e alinhamento facial, aplicando um pré-processamento padronizado antes da classificação da expressão. A análise comparativa entre os modelos e o impacto da otimização com motores de inferência como ONNX Runtime ou TensorRT permitirá selecionar a solução mais equilibrada. Como contribuição adicional, este estudo visa expor as limitações de modelos unimodais (baseados apenas em imagens) para a análise de sentimentos, sugerindo o desenvolvimento de modelos multimodais como um caminho promissor para futuras pesquisas.

II. Materiais e Métodos

Para este estudo, serão utilizadas três bases de dados: FER-2013 (35.887 imagens de 48×48 pixels em tons de cinza, coletadas automaticamente e com variações desafiadoras), RAF-DB (∼5.000 imagens de ambientes não controlados, alta diversidade) e DFEW (11.697 clipes de vídeo curtos extraídos de filmes, contendo expressões faciais em contextos dinâmicos e não controlados), abrangendo desde cenários padronizados até condições reais com ruídos, variações visuais e sequências temporais.

As imagens passarão por um *pipeline* de pré-processamento que inclui redimensionamento para 254x254 pixels, conversão para tons de cinza quando aplicável, normalização de pixels na faixa [0, 1], alinhamento facial e aplicação de *data augmentation* (rotações, espelhamentos horizontais e ajustes de brilho/contraste) para aumentar a variabilidade e reduzir o risco de *overfitting*.

Figura 2. Exemplos de imagens de três conjuntos de dados de reconhecimento de emoções, dispostos por linhas. De cima para baixo: RAF-DB, CK+ e FER2013. As emoções estão organizadas por colunas, da esquerda para a direita: felicidade, tristeza e raiva. MUDAR ESSA FIGURAAAAAAA

As arquiteturas utilizadas serão *Deep Learning* baseadas em *Convolutional Neural Networks* (CNNs): ResNet-50 (rede profunda com blocos residuais) e EfficientNet (arquitetura escalável e eficiente), e *Vision Transformers*: EfficientVit (eficiência de memória e velocidade de inferência, sem comprometer a acurácia). Todas serão treinadas via *fine-tuning* com pesos pré-treinados e ajustadas para sete classes universais de expressão facial (*anger, disgust, fear, happy, neutral, sadness, surprise*), sendo avaliadas em validação intra-base e *cross-dataset*.

O protocolo experimental segue as etapas: aquisição e organização dos *datasets*, pré-processamento padronizado, treinamento supervisionado das CNNs e avaliação *cross-dataset*. As métricas consideradas serão acurácia, F1-score, tempo de inferência e uso de memória, incluindo a análise do impacto do *Cross-Dataset Drift* (CDD). A etapa de *FER* em tempo real será implementada futuramente para complementar os resultados e verificar a viabilidade dos modelos em aplicações práticas.

Os experimentos serão realizados em três configurações de hardware com Ubuntu 24.04:

- Desktop 1: Intel Core i7-8700K, GPU NVIDIA GTX 1070 Ti, com 8 GB de VRAM;
- Desktop 2: Intel Core i5-12400F, GPU NVIDIA RTX 3060, com 12 GB de VRAM;
- Notebook: Intel Core i7-11800H, GPU NVIDIA RTX 3050, com 4 GB de VRAM.

A implementação será feita em Python, utilizando PyTorch, Scikit-learn, OpenCV, NumPy, Pandas, Matplotlib e Seaborn, no Visual Studio Code, com controle de versão via GitHub e conteinerização em Docker. Os resultados e registros de execução serão salvos em arquivos CSV.

III. Etapas do Projeto e Marcos Físicos

O desenvolvimento do projeto foi organizado em etapas, de forma a assegurar um progresso contínuo e mensurável até a entrega final. Cada fase contempla atividades específicas, com objetivos e marcos que permitirão verificar sua conclusão.

A primeira etapa corresponde ao levantamento do estado da arte, abrangendo pesquisa bibliográfica para aprofundamento de modelos de *Deep Learning* e *Vision Transformers* já usados, além da implementação de novos modelos, técnicas de inferência em tempo real e métodos para mitigação do *Cross-Dataset Drift* (CDD). Espera-se a consolidação de um conjunto de referências relevantes e o embasamento teórico necessário para orientar as próximas etapas.

Em seguida, será realizado o processamento dos conjuntos de dados, incluindo análise exploratória, normalização, redimensionamento, conversão para escala de cinza e correção do balanceamento das classes. O marco desta etapa será a disponibilização das bases prontas para uso nos experimentos, com registro do processo e parâmetros adotados.

A etapa seguinte concentra-se na experimentação, com a configuração e treinamento das arquiteturas *EfficientVit*, *ResNet-50* e *EfficientNet*. Serão aplicadas técnicas de regularização e conduzidas otimizações para execução em tempo real. O marco dessa fase será a obtenção de modelos treinados e validados, com métricas preliminares registradas.

Posteriormente, seguirá com a análise de resultados, contemplando avaliação *intra-dataset* e *cross-dataset*, comparação de desempenho entre modelos e identificação de limitações. Subsequentemente, serão consolidadas as métricas finais e produzidas interpretações que embasem a discussão no manuscrito.

Por fim, ocorrerá a redação e revisão do trabalho, envolvendo a organização do conteúdo, inclusão de figuras e tabelas, formatação final e revisão textual. O marco conclusivo será a submissão da versão final do manuscrito.

Tabela I
RESUMO DE MATERIAIS E MÉTODOS (EXEMPLO 1).

Materiais	**Descrição**	**Observações**
Conjuntos de Dados	DFEW [**?**], FER2013 [**?**], RAF-DB [**?**]	Abrangem condições reais e desafiadoras, viabilizando análise de robustez frente ao CDD. **(incluir referências)**.
Modelos	EfficientVit [**?**], ResNet-50 [**?**], EfficientNet [**?**]	Arquiteturas de *Deep Learning* e *Vision Transformers* escolhidas pelo equilíbrio entre acurácia e eficiência, visando uso em tempo real e otimização futura. **(incluir referências)**.
Técnicas	Redimensionamento para 254x254 pixels, Conversão para tons de cinza, normalização de pixels, linhamento facial, Data Augmentation	Estratégias de pré-processamento e aumento de dados para padronizar entradas e mitigar sobreajuste.
Linguagens / Bibliotecas	Python, PyTorch, Scikit-learn, OpenCV, NumPy, Pandas, Matplotlib, Seaborn	Principais ferramentas de visão computacional e aprendizado de máquina, garantindo flexibilidade e suporte.
Hardware / Recursos	Desktop GPU GTX 1070 Ti com 8 GB de VRAM; Desktop GPU RTX 3060 com 12 GB de VRAM; Notebook GPU RTX 3050 com 4 GB de VRAM	Configurações com GPUs de médio desempenho, adequadas para treinamento e testes propostos.

IV. CRONOGRAMA

O cronograma foi organizado de modo a permitir que as atividades avancem de forma progressiva e, quando possível, em paralelo, otimizando o uso do tempo e dos recursos disponíveis. Nas primeiras semanas, o foco recairá sobre a preparação do ambiente de trabalho e a base conceitual, com a criação do repositório voltado para abordagens em *Deep Learning* e *Vision Transformers*, a pesquisa sobre técnicas de processamento em tempo real e a revisão exploratória dos dados, seguida pelo pré-processamento das bases e pela correção do balanceamento das classes. A partir da terceira semana, iniciará a fase de experimentação, contemplando a configuração e o treinamento dos modelos *EfficientVit*, *ResNet-50* e *EfficientNet*, a aplicação de técnicas de regularização e a otimização para execução em tempo real. Essa fase se estenderá pelas semanas seguintes, sendo acompanhada de avaliações de eficiência computacional, comparação de resultados *intra-dataset* e *cross-dataset* de modo a identificar gargalos e limitações dos modelos. Na reta final, as métricas consolidadas e a discussão dos resultados serão incorporadas à redação do manuscrito, que passará por revisões gerais de conteúdo e forma, ajustes de figuras, tabelas e formatação, culminando na submissão da versão final.

A Tabela II sintetiza essa organização, estruturando as atividades em um intervalo de duas semanas, descrevendo quais serão as principais tarefas executadas. As etapas iniciais concentram-se na pesquisa e no processamento dos dados, seguidas por períodos dedicados à experimentação e análise de resultados, culminando nas semanas finais com a finalização e revisão do manuscrito.

Tabela II
MODELO DE CRONOGRAMA DO PROJETO (SEMANAS CONTADAS A PARTIR DA ENTREGA DESTE PLANEJAMENTO ATÉ A ENTREGA FINAL EM 09/11).

Atividade	**Sem. 1–2**	**Sem. 3–4**	**Sem. 5–6**	**Sem. 7–8**	**Sem. 9–11 (até 09/11)**
Criar novo repositório com *Deep Learning*	X				
Pesquisar artigos sobre *Real Time*	X	X			
Revisão da análise exploratória	X	X			
Pré-processamento das bases		X			
Corrigir balanceamento dos datasets		X			
Iniciar experimentação com os modelos		X			
Correção e ajustes dos modelos		X	X		
Treinar os modelos definidos			X		
Executar validação cruzada			X		
Aplicar técnicas de regularização			X	X	
Otimizar modelos para tempo real			X	X	
Avaliar eficiência computacional			X	X	
Comparar resultados intra-dataset e cross-dataset			X	X	
Identificar gargalos e limitações do modelo.			X	X	
Consolidar métricas finais				X	X
Preparar discussão dos resultados				X	X
Revisar todos os experimentos					X
Revisão geral do texto do artigo					X
Ajustar formatação, figuras e tabelas para submissão					X
Entregar versão final					X

REFERÊNCIAS