A Estrutura da Proposta

A estrutura da **Proposta do Projeto** entregue (observe o prazo no Plano de Ensino da disciplina) é:

1. Título do projeto

Desenvolvimento de um Sistema de Reconhecimento de Gestos Dinâmicos em Tempo Real Usando Temporal Transformers e Esqueletos 2D

2. Nomes dos participantes da equipe

Davi Baechtold Campos

3. Nome e assinatura do orientador

Orientador: Prof. Dr. Alceu de Souza Brito Junior

Co-orientador: Prof. Dr. Alessandro Zimmer

4. Motivação para o desenvolvimento do projeto

A interação humano-máquina (IHM) tem se tornado cada vez mais presente em nosso cotidiano, impulsionando a busca por interfaces mais naturais, intuitivas e acessíveis. Os métodos tradicionais de controle, como teclados, mouses e telas sensíveis ao toque, embora eficientes, ainda apresentam limitações em determinados contextos, como em ambientes automotivos, onde a atenção do motorista deve ser mantida na estrada, ou no controle de sistemas robóticos complexos, que demandam uma interação mais fluida.

Nesse cenário, o reconhecimento de gestos surge como uma alternativa promissora, permitindo que os usuários controlem dispositivos e interajam com sistemas digitais por meio de movimentos corporais. No entanto, o desenvolvimento de sistemas de reconhecimento de gestos robustos e em tempo real, capazes de interpretar uma variedade de movimentos dinâmicos a partir de um fluxo de vídeo de uma câmera comum (monocular), ainda é um desafio significativo. A extração de características relevantes e a modelagem das dependências temporais dos gestos são etapas cruciais que exigem abordagens sofisticadas.

Este projeto é motivado pela necessidade de explorar e avançar nas técnicas de reconhecimento de gestos, utilizando tecnologias de visão computacional e aprendizado profundo para criar um sistema eficiente e de baixo custo. A aplicação de modelos de *deep learning*, como os *Transformers*, que têm demonstrado grande sucesso em tarefas de modelagem de sequências, pode oferecer uma solução robusta para a interpretação de gestos dinâmicos a partir de sequências de esqueletos 2D. O desenvolvimento de tal sistema tem o potencial de impactar positivamente áreas como a indústria automotiva, a robótica e a criação de interfaces mais inclusivas e acessíveis.

5. Objetivo

O objetivo principal deste trabalho é desenvolver e avaliar um sistema de reconhecimento de gestos dinâmicos em tempo real, baseado em sequências de esqueletos 2D extraídas de um vídeo monocular, utilizando um modelo de aprendizado profundo do tipo *Temporal Transformer*.

Para alcançar este objetivo principal, os seguintes objetivos específicos foram definidos:

- **Desenvolver um pipeline para extração de dados:** Implementar um sistema para extrair os pontos-chave (landmarks) do esqueleto humano a partir de um fluxo de vídeo em tempo real, utilizando a biblioteca MediaPipe.
- **Revisão da Literatura e Levantamento de Dados:** Realizar uma revisão do estado da arte em métodos de reconhecimento de gestos e levantar bases de dados existentes para o treinamento e avaliação do modelo.
- **Construir e treinar um modelo de classificação:** Desenvolver um modelo de *Transformer* capaz de aprender as características temporais de sequências de gestos e classificá-las em diferentes categorias.
- **Avaliar o desempenho do modelo:** Realizar uma avaliação quantitativa do modelo treinado, utilizando métricas como acurácia e F1-score, em um conjunto de dados de gestos.
- **Criar uma demonstração em tempo real:** Desenvolver uma aplicação que utilize o modelo treinado para classificar gestos capturados por uma webcam em tempo real, demonstrando a viabilidade da solução.
- **Analisar a aplicação em casos de uso específicos:** Explorar a aplicação do sistema em, pelo menos, dois casos de uso: controle de interface em um ambiente automotivo simulado e controle de um braço robótico.

6. Descrição sucinta das teorias e técnicas utilizadas

O sistema proposto será desenvolvido utilizando uma combinação de técnicas de visão computacional e aprendizado profundo. O processo pode ser dividido nas seguintes etapas:

1. **Extração de Esqueletos 2D:** A primeira etapa consiste em extrair os pontos-chave (landmarks) do corpo e das mãos a partir do vídeo da webcam. Para isso, será utilizada a biblioteca **MediaPipe**, que oferece modelos de *deep learning* pré-treinados e otimizados para a detecção de pose em tempo real. Especificamente, será empregado o modelo **BlazePose**, que foi treinado em bases de dados de larga escala, como a **COCO (Common Objects in Context)**, além de datasets proprietários que garantem alta acurácia em uma vasta gama de movimentos. Para a detecção das mãos, o MediaPipe utiliza um pipeline composto por um detector de palmas (BlazePalm) e um modelo de landmarks, ambos treinados em extensas bases de dados internas de imagens de mãos. A utilização desses modelos pré-treinados elimina a necessidade de coletar e treinar um modelo de visão computacional do zero, permitindo focar na etapa de classificação temporal dos gestos. O resultado dessa etapa é uma sequência de vetores de coordenadas 2D que representam a pose do usuário em cada quadro do vídeo.
2. **Pré-processamento e Normalização:** As sequências de coordenadas extraídas serão pré-processadas e normalizadas para garantir que o modelo seja robusto a variações de escala, translação e rotação. Isso é fundamental para que o sistema funcione independentemente da posição do usuário em relação à câmera.
3. **Modelagem com *Temporal Transformer*:** O coração do sistema é um modelo de aprendizado profundo baseado na arquitetura **Transformer**. Os *Transformers* são conhecidos por seu mecanismo de **auto-atenção** (*self-attention*), que permite ao modelo ponderar a importância de diferentes elementos em uma sequência. No contexto de reconhecimento de gestos, isso significa que o modelo pode aprender quais poses em uma sequência de movimentos são mais relevantes para a classificação de um gesto. A natureza "temporal" do modelo refere-se à sua capacidade de processar a sequência de poses ao longo do tempo. Além disso, uma camada de **Codificação Posicional** (*Positional Encoding*) será utilizada para fornecer ao modelo informações sobre a ordem dos quadros na sequência.
4. **Treinamento e Classificação:** O modelo será treinado em um conjunto de dados de gestos, onde cada gesto é representado por uma sequência de esqueletos 2D. Após o treinamento, o modelo será capaz de receber uma nova sequência de gestos e classificá-la em uma das categorias aprendidas.

As principais tecnologias utilizadas serão a linguagem de programação **Python**, a biblioteca de aprendizado profundo **PyTorch** para a implementação do modelo *Transformer*, e as bibliotecas **OpenCV** e **MediaPipe** para o processamento de vídeo e extração de esqueletos.

Tabela 1: Conjuntos de dados de língua de sinais e gestos.

Tabela 2 – Conjuntos de dados de língua de sinais utilizados nos testes, apresentando informações sobre o número de classes, tamanhos das imagens e quantidade de amostras destinadas aos testes. As línguas de sinais suportadas pelos conjuntos de dados são: ASL (*American Sign Language*), ISL (*Indian Sign Language*), LSA (*Argentine Sign Language*), ArSL (*Arabic Sign Language*), BdSL (*Bengali Sign Language*), PSL (*Pakistan Sign Language*), DGS (*German Sign Language*) e LIBRAS (Língua Brasileira de Sinais).

Nome	Linguagem de Sinal	Classes	Tamanho das imagens	Quantidade de amostras usadas para teste
NUS Hand Posture dataset I	Não definida	9	160x120	241
NUS Hand Posture dataset II	Não definida	9	160x120	2.000
OUHANDS	Não definida	10	640x480	1.000
ASL Digits	ASL	10	100x100	2.062
Indian Alphabet	ISL	13	128x128	15.600
HAGRID	Não definida	13	512x683	13.000
HG14	Não definida	14	256x256	14.000
LSA16 handshapes	LSA	15	640x480	800
Pugeault	ASL	21	87x124	12.547
ArSL21L	ArSL	21	416x416	14.202
ASL Alphabet	ASL	23	200x200	28
KU-BdSL	BDSL	25	3024x4032	1.500
PSL	PSL	31	640x480	1.480
Bengali Alphabet	BDSL	34	224x224	1.520
PHOENIX-14 Handshapes	DGS	44	93x132	1.837
LSWH100	LIBRAS	100	500x500	4.000

Fonte: A autora (2025)

7. Descrição do hardware e/ou do software necessários para o desenvolvimento

Hardware:

- **Computador Pessoal:** Um computador com processador moderno (ex: Intel Core i5/i7 ou equivalente) e, no mínimo, 8 GB de RAM.
- **GPU (Unidade de Processamento Gráfico):** Para o treinamento do modelo de *deep learning*, é altamente recomendável o uso de uma GPU NVIDIA com suporte a CUDA, o que acelera significativamente o processo de treinamento. O treinamento pode ser realizado em um computador pessoal com uma GPU dedicada ou

utilizando serviços de nuvem.
- **Webcam:** Uma webcam padrão para a captura de vídeo em tempo real.
- **Origem dos recursos:** Os recursos de hardware são próprios do aluno.

Software:

- **Sistema Operacional:** O desenvolvimento será realizado em um sistema operacional Linux (Mint 22.04+).
- **Linguagem de Programação:** Python 3.8+.
- **Ambiente de Desenvolvimento:** Será utilizado um ambiente virtual Python (`venv`) para gerenciar as dependências do projeto.
- **Bibliotecas Principais:**
    - **PyTorch:** Framework de *deep learning* para a implementação do modelo.
    - **OpenCV-Python:** Para processamento de imagem e vídeo em tempo real.
    - **MediaPipe:** Para a extração de esqueletos 2D.
    - **NumPy, Pandas, Scikit-learn:** Para manipulação de dados e avaliação do modelo.
    - **PyYAML:** Para o gerenciamento de configurações do projeto.

8. Contexto do Projeto

Este projeto, embora não tenha sido impulsionado por um cliente externo, teve seu início promissor dentro da THI (Technische Hochschule Ingolstadt) como parte de um intercâmbio acadêmico. A iniciativa surgiu de uma proposta do Prof. Dr. Alessandro Zimmer, cujo foco principal era a exploração e o desenvolvimento de soluções inovadoras para a automação de automóveis. Esse contexto proporcionou um ambiente de pesquisa e experimentação, permitindo que a equipe se aprofundasse em tecnologias emergentes e conceitos avançados no campo da direção autônoma, sem as restrições e prazos muitas vezes impostos por demandas comerciais. O intercâmbio, além de enriquecer a experiência dos participantes, visava fomentar a colaboração internacional e o avanço do conhecimento em um setor de crescente relevância tecnológica e econômica.

Duas principais áreas de aplicação serão exploradas como casos de uso:

1. **Indústria Automotiva:** O sistema de reconhecimento de gestos pode ser aplicado para criar interfaces de controle dentro de veículos. O motorista poderia, por exemplo, controlar o sistema de som, o ar-condicionado ou o sistema de navegação com gestos simples, sem a necessidade de desviar a atenção da estrada para interagir com botões ou telas.
2. **Robótica:** O sistema pode ser utilizado para o controle de robôs, como braços robóticos. Isso permitiria uma interação mais intuitiva e remota com os robôs, o que é particularmente útil em ambientes industriais ou em aplicações de assistência. Há a possibilidade de integrar o projeto com o laboratório de robótica da PUCPR para testes e demonstrações práticas.

9. Referências bibliográficas básicas

ALMJALLY, Abrar; ALMUKADI, Wafa Sulaiman. Deep computer vision with artificial intelligence based sign language recognition to assist hearing and speech-impaired individuals. **Scientific Reports**, v. 15, n. 1, p. 32268, 2025.

JIANG, Songyao et al. Skeleton Aware Multi-modal Sign Language Recognition. **arXiv preprint arXiv:2103.08833**, 2021.

LI, Dongxu et al. Transferring Cross-domain Knowledge for Video Sign Language Recognition. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. 2020. p. 6204-6213.

SILVA, Karolayne Teixeira da. **SignWriting para Reconhecimento de Gestos em Língua de Sinais**. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2025.

VASWANI, Ashish et al. Attention is all you need. In: **Advances in neural information processing systems**. 2017. p. 5998-6008.

Assinaturas

Prof. Dr. Alceu de Souza Brito Junior

Orientador

Importante: A proposta deve ter, no máximo, **duas páginas** e **ser assinada pelo orientador.**