

Projeto de Reconhecimento de Libras

Este projeto visa desenvolver um sistema de reconhecimento de Libras utilizando MediaPipe para detecção de pontos-chave e Keras para treinamento de uma rede neural.

Estrutura de Diretórios

- `libras_project/`
- `data/` : Armazenará os dados coletados (coordenadas dos pontos-chave e rótulos).
- `models/` : Armazenará os modelos de rede neural treinados.
- `scripts/` : Conterá os scripts Python para coleta de dados, treinamento e aplicação em tempo real.
- `README.md` : Este arquivo, contendo a documentação do projeto.

Formato de Armazenamento dos Dados

Os dados coletados serão armazenados em arquivos `.npy` (NumPy array) dentro do diretório `data/`. Cada arquivo representará um gesto e conterá uma sequência de frames. Cada frame, por sua vez, conterá as coordenadas dos pontos-chave detectados pelo MediaPipe.

Estrutura de um arquivo `.npy` para um gesto:

```
# Exemplo de estrutura de dados para um gesto
[
  # Frame 1
  [
    [x1, y1, z1], # Ponto 1 (ex: pulso da mão direita)
    [x2, y2, z2], # Ponto 2 (ex: ponta do dedo indicador da mão
direita)
    ...
  ],
  # Frame 2
  [
    [x1, y1, z1],
    [x2, y2, z2],
    ...
  ],
]
```

```
] ...
```

Cada ponto-chave terá 3 coordenadas (x, y, z). O número de pontos-chave por frame será fixo, determinado pelo MediaPipe Hands (21 pontos por mão, se ambas as mãos forem detectadas, serão 42 pontos, além de outros pontos como ombros e cotovelos que podem ser adicionados com MediaPipe Pose).

O nome do arquivo `.npy` será o rótulo do gesto (ex: `ola.npy`, `obrigado.npy`).

Arquitetura Geral dos Programas

1. Programa de Coleta de Coordenadas (`scripts/collect_data.py`)

- **Entrada:** Nome do gesto (rótulo) e número de amostras/frames por gesto.
- **Processo:**
 - Inicializa a câmera e o MediaPipe Hands (e MediaPipe Pose, se necessário).
 - Para cada amostra do gesto:
 - Captura frames da câmera.
 - Processa cada frame com MediaPipe para detectar pontos-chave.
 - Extrai as coordenadas (x, y, z) dos pontos-chave.
 - Armazena as coordenadas em uma estrutura de dados temporária para o frame.
 - Agrupa os frames para formar uma sequência de um gesto.
 - Salva a sequência de coordenadas em um arquivo `.npy` no diretório `data/`, com o nome do rótulo.
- **Saída:** Arquivos `.npy` contendo as coordenadas dos gestos.

2. Programa de Treinamento da Rede Neural (`scripts/train_model.py`)

- **Entrada:** Dados coletados do diretório `data/`.
- **Processo:**
 - Carrega os dados `.npy` e os rótulos.
 - Pré-processa os dados (normalização, padding/truncamento de sequências, one-hot encoding dos rótulos).
 - Constrói a arquitetura da rede neural em Keras (LSTM/GRU para sequências).
 - Treina o modelo com os dados.
 - Avalia o desempenho do modelo.
 - Salva o modelo treinado em um arquivo `.h5` no diretório `models/`.

- **Saída:** Arquivo `.h5` contendo o modelo de rede neural treinado.

3. Programa de Aplicação Final (`scripts/realtime_app.py`)

- **Entrada:** Câmera em tempo real e o modelo treinado (`.h5`).
- **Processo:**
 - Carrega o modelo treinado.
 - Inicializa a câmera e o MediaPipe Hands (e MediaPipe Pose, se necessário).
 - Em um loop contínuo:
 - Captura um frame da câmera.
 - Processa o frame com MediaPipe para detectar pontos-chave.
 - Extrai as coordenadas e as formata para a entrada do modelo.
 - Realiza a inferência com o modelo para prever o gesto.
 - Exibe o gesto reconhecido na tela.
- **Saída:** Reconhecimento de gestos em tempo real na tela.

Dependências

- `Python 3.x`
- `mediapipe`
- `tensorflow` (com Keras)
- `numpy`
- `opencv-python`
- `scikit-learn` (para pré-processamento e avaliação)

Como Usar o Projeto

Este projeto é dividido em três etapas principais:

1. Coleta de Dados (`scripts/collect_data.py`)

Este script é usado para coletar as coordenadas dos pontos-chave das mãos para diferentes gestos. Cada gesto será salvo como um arquivo `.npy`.

Pré-requisitos: - Câmera conectada e funcionando.

Instruções: 1. Certifique-se de que todas as dependências estão instaladas (`pip install mediapipe numpy opencv-python tensorflow`). 2. Execute o script no terminal: `bash python3 libras_project/scripts/collect_data.py` 3. O programa pedirá para você digitar o rótulo (nome) do gesto que deseja coletar (ex: `ola` ,

obrigado). 4. Em seguida, ele perguntará quantas sequências (amostras) você quer coletar para este gesto e quantos frames por sequência. Para gestos simples, 10-20 sequências com 30-50 frames por sequência podem ser um bom ponto de partida. Para gestos mais complexos, você pode precisar de mais. 5. A câmera será aberta. Posicione suas mãos de forma que o MediaPipe possa detectá-las. O script irá exibir o vídeo com as anotações dos pontos-chave. 6. Realize o gesto para cada sequência. O script irá coletar os frames automaticamente. 7. Pressione `q` para sair da janela de visualização da câmera a qualquer momento. 8. Os dados coletados serão salvos no diretório `libras_project/data/` com o nome `[rotulo]_[numero_sequencia].npy`.

Dicas para Coleta de Dados:

- **Variação:** Colete dados com variações na iluminação, ângulo da câmera e velocidade do gesto para tornar o modelo mais robusto.
- **Fundo:** Tente usar um fundo limpo e consistente para minimizar ruídos.
- **Número de Amostras:** Quanto mais amostras por gesto, melhor o modelo poderá aprender. Comece com um número razoável e aumente se a precisão for baixa.
- **Gestos Complexos:** Para gestos mais complexos que envolvem movimento, certifique-se de que a sequência de frames capture todo o movimento do gesto.

2. Treinamento da Rede Neural (`scripts/train_model.py`)

Este script utiliza os dados coletados para treinar um modelo de rede neural capaz de reconhecer os gestos.

Pré-requisitos: - Dados coletados no diretório `libras_project/data/`.

Instruções:

1. Execute o script no terminal: `bash python3 libras_project/scripts/train_model.py`
2. O script irá carregar os dados, pré-processá-los, construir a arquitetura da rede neural, treinar o modelo e salvar o modelo treinado (`libras_model.h5`) e as classes (`classes.npy`) no diretório `libras_project/models/`.
3. O progresso do treinamento será exibido no terminal, incluindo a precisão do modelo.

Dicas para Treinamento:

- **Quantidade de Dados:** Se a precisão do modelo for baixa, considere coletar mais dados para cada gesto.
- **Arquitetura do Modelo:** Para projetos mais avançados, você pode experimentar diferentes arquiteturas de rede neural (mais camadas LSTM, diferentes tamanhos de unidades, etc.).
- **Hiperparâmetros:** Ajuste os hiperparâmetros de treinamento (épocas, tamanho do batch) para otimizar o desempenho.

3. Aplicação em Tempo Real (`scripts/realtime_app.py`)

Este script é a aplicação final que usa o modelo treinado para reconhecer gestos em tempo real através da câmera.

Pré-requisitos: - Modelo treinado (`libras_model.h5`) e arquivo de classes (`classes.npy`) no diretório `libras_project/models/` . - Câmera conectada e funcionando.

Instruções: 1. Execute o script no terminal: `bash python3 libras_project/scripts/realtime_app.py` 2. A câmera será aberta e o script tentará reconhecer os gestos em tempo real. O gesto previsto será exibido na tela. 3. Pressione `q` para sair da aplicação.

Observações: - A precisão do reconhecimento em tempo real dependerá da qualidade dos dados de treinamento e do desempenho do modelo. - Certifique-se de que as condições de iluminação e o posicionamento das mãos sejam semelhantes aos da fase de coleta de dados para obter melhores resultados.