

Projeto Checkpoint 02: Redes Neurais com Keras + Visão Computacional

Este repositório contém a solução para o Checkpoint 02 da disciplina "DISRUPTIVE ARCHITECTURES: IOT, IOB E GENERATIVE AI", abordando Redes Neurais com Keras e Visão Computacional com YOLOv8 e Hugging Face (DETR).

Objetivo do Projeto

O objetivo deste projeto é demonstrar a aplicação de Redes Neurais para classificação e regressão utilizando Keras, e explorar técnicas de Visão Computacional para detecção de objetos com duas ferramentas distintas: YOLOv8 e Hugging Face (DETR). O projeto visa comparar o desempenho e as características de cada abordagem.

Estrutura do Repositório

O repositório está organizado da seguinte forma:

Plain Text

```
cpiotpart2/
├─ exercise_1_wine_classification.py
├─ exercise_1_results.txt
├─ exercise_2_california_housing_regression.py
├─ exercise_2_results.txt
├─ FINAL_Exemplo_Redес_Neурais_Com_Keras.py
├─ Relatório de Treinamento de Redes Neurais com Keras.md
├─ yolov8_detection.py
├─ yolov8n.pt
├─ yolov8_results_summary.txt
├─ huggingface_detection.py
├─ huggingface_results_summary.txt
├─ README_part2.md
└─ ... (outros arquivos gerados)
```

- **exercise_1_wine_classification.py** : Script Python para classificação multiclasse de vinhos usando Keras e RandomForestClassifier.
- **exercise_2_california_housing_regression.py** : Script Python para regressão de valores de casas (California Housing) usando Keras e RandomForestRegressor.
- **FINAL_Exemplo_Redес_Neурais_Com_Keras.py** : Exemplo final de redes neurais com Keras.

- **Relatório de Treinamento de Redes Neurais com Keras.md** : Relatório detalhado sobre a parte de Redes Neurais.
- **yolov8_detection.py** : Script Python para detecção de objetos usando YOLOv8.
- **yolov8n.pt** : Modelo pré-treinado YOLOv8 (nano).
- **huggingface_detection.py** : Script Python para detecção de objetos usando Hugging Face (DETR).
- **README_part2.md** : README original fornecido para a parte de Visão Computacional.
- ***_results.txt** : Arquivos de texto gerados pelos scripts contendo os resultados das execuções.

Requisitos de Ambiente

Para executar os scripts, você precisará ter Python 3.x instalado, juntamente com as seguintes bibliotecas. Recomenda-se criar um ambiente virtual.

Instalação das Dependências

Bash

```
pip install pandas scikit-learn keras tensorflow ultralytics transformers  
torch torchvision timm
```

Download do Dataset de Visão Computacional

Os scripts de Visão Computacional utilizam o **Tiny COCO Dataset** . Você pode cloná-lo do GitHub:

Bash

```
git clone https://github.com/lizhogn/tiny_coco_dataset.git  
/home/ubuntu/tiny_coco_dataset
```

Certifique-se de que o diretório **/home/ubuntu/tiny_coco_dataset/tiny_coco/val2017** contenha as imagens para a detecção.

Como Rodar o Projeto

Siga os passos abaixo para executar cada parte do projeto.

1. Navegar para o Diretório do Projeto

Primeiro, navegue até o diretório onde os arquivos do projeto foram extraídos:

```
Bash
```

```
cd /home/ubuntu/cpiotpart2
```

2. Executar os Exercícios de Redes Neurais

Exercício 1: Classificação Multiclasse (Wine Dataset)

Este script treina e compara uma Rede Neural com Keras e um `RandomForestClassifier` para classificar vinhos.

```
Bash
```

```
python3.11 exercise_1_wine_classification.py
```

Os resultados (acurácia) serão exibidos no console e salvos em `exercise_1_results.txt`.

Exercício 2: Regressão (California Housing Dataset)

Este script treina e compara uma Rede Neural com Keras e um `RandomForestRegressor` para prever valores de casas.

```
Bash
```

```
python3.11 exercise_2_california_housing_regression.py
```

Os resultados (RMSE e MAE) serão exibidos no console e salvos em `exercise_2_results.txt`.

3. Executar os Exercícios de Visão Computacional

Detecção de Objetos com YOLOv8

Este script realiza a detecção de objetos nas imagens do `Tiny COCO Dataset` usando o modelo `yolov8n.pt`.

```
Bash
```

```
python3.11 yolov8_detection.py
```

Os resultados da detecção serão exibidos no console e salvos em `yolov8_results_summary.txt`. As imagens com as detecções visualizadas serão salvas no diretório

```
/home/ubuntu/yolov8_output/ .
```

Detecção de Objetos com Hugging Face (DETR)

Este script realiza a detecção de objetos em uma imagem do Tiny COCO Dataset usando o modelo facebook/detr-resnet-50 do Hugging Face. (Nota: Para evitar problemas de memória em ambientes restritos, este script foi ajustado para processar apenas a primeira imagem encontrada no dataset).

Bash

```
python3.11 huggingface_detection.py
```

Os resultados da detecção serão exibidos no console e salvos em huggingface_results_summary.txt . A imagem com as detecções visualizadas será salva no diretório /home/ubuntu/huggingface_output/ .

Análise e Resultados

Uma análise detalhada dos requisitos do projeto e dos resultados obtidos por cada script pode ser encontrada no arquivo execution_instructions.md (gerado pelo agente) e no Relatório de Treinamento de Redes Neurais com Keras.md .

Os arquivos *_results.txt contêm os resumos das métricas de desempenho para cada modelo, e os diretórios yolov8_output/ e huggingface_output/ conterão as imagens com as detecções visuais.