AI & SUSTAINABILITY IN VET EDUCATION

Nome do(s) autor(es): Sucessos Criativos, Lda

Título do WP: WP3 – IA-SAVES@ProjectHub

Referência de Atividade: 3.2 – Modelo de projeto para IA-SAVES@ProjectHub

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Índice

1. Título do Projeto: 4
2. Categoria do Projeto: 4
3. Descrição do Projeto: 4
4. Palavras-chave: 4
5. Público-alvo: 4
6. Objetivos do Projeto: 4
7. Cronologia do Projeto: 5
8. Recursos Fornecidos: 5
9. Ferramentas e Tecnologias Utilizadas: 6
10. Plano de Implementação Passo a Passo: 7
Passo 1: Prepare o Raspberry Pi para Interação de Hardware 7
Passo 2: Ligue o Sensor de LDR 7
Passo 3: Escreve o programa Python para Detetar Luz 8
11. Critérios de Avaliação: 12
12. Línguas Disponíveis: 12
13. Colaboradores: 12
17. Feedback e Melhoria Contínua: 13

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

1. Título do Projeto:

Treino MobileNet sem Aprendizagem por Transferência

---

2. Categoria do Projeto:

IA na Educação

---

3. Descrição do Projeto:

Neste projeto, vamos treinar uma rede neural baseada em MobileNetV3 do zero (ou seja, sem pesos pré-treinados) para classificar imagens de rosas, girassóis e tulipas. Ao contrário da abordagem de aprendizagem por transferência abordada no curso "Training MobileNet With Transfer Learning", este método inicializa todos os pesos do modelo aleatoriamente e aprende inteiramente a partir do seu conjunto de dados.

---

4. Palavras-chave:

Classificação de imagens, MobileNet, treino do zero, deep learning, Google Colab, Kaggle, conjunto de dados de flors, conjunto de dados personalizado, aumento de dados, aprendizagem sem transferência

---

5. Público-alvo:

Educadores, aprendizes e criadores, com ou sem experiência prévia, que estão a iniciar a sua jornada em IA.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

6. Objetivos do Projeto:

Este micro-curso ensina como construir e treinar um modelo de rede neural convolucional (CNN) do zero usando **o MobileNetV3Small** sem depender de pesos pré-treinados. O projeto irá focar-se na classificação de imagens de rosas, **girassóis** e **tulipas** usando um conjunto de dados personalizado, destacando as principais diferenças e desafios ao trabalhar sem aprendizagem por transferência.

No final do projeto, será capaz de:

- Compreenda como instanciar o MobileNetV3 sem pesos pré-treinados
- Treine uma CNN leve do zero usando um conjunto de dados balanceado
- Aplicar pré-processamento de imagem e aumento de dados para alimentar o modelo
- Guardar e reutilizar o modelo treinado para inferência e testes

Este projeto destina-se a **complementar** o curso de aprendizagem de transferência anterior e a desenvolver o seu conhecimento sobre técnicas de preparação e avaliação de conjuntos de dados.

---

7. Cronologia do Projeto:

A duração estimada é de aproximadamente **2,5 a 3 horas**.

---

8. Recursos Fornecidos:

- Para estas tarefas pode usar tanto um computador como o seu raspberry pi.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- Referência do conjunto de dados: Kaggle Flowers Dataset

---

9. Ferramentas e Tecnologias Utilizadas:

Ferramenta / Biblioteca	Finalidade
Python 3	Programação geral e scripting
TensorFlow / Keras	Estrutura de rede neural para construção e treino do MobileNetV3
MobileNetV3Small	Arquitetura CNN leve usada para tarefas de classificação eficientes
ImageDataGenerator	Aumento de imagem e pré-processamento
Google Colab	Ambiente de notebooks Python baseados na cloud com suporte a GPU
Matplotlib / Seaborn	(via reutilização do Curso 2) Usado para visualização de resultados de treino

---

10. Plano de Implementação Passo a Passo:

Passo 1 — Reutilizar a Configuração do Curso 2

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Antes de continuar, certifique-se de que já completou os passos de configuração descritos no Curso 2 – MobileNet com Transfer Learning, incluindo:

Recolha e pré-processamento de conjuntos
de dados Organização de pastas e aumento de dados
Montagem do Google Drive no Colab
Ativação do índice para navegação mais
fácil Criação do diretório:

/Treino/conjuntos de dados/floresToTrain/

├── comboio/

├── val/

└── teste/

> Se estes passos ainda não foram concluídos, por favor regresse ao Curso "MobileNet com Transferência de Aprendizagem" e siga-os antes de prosseguir. Este curso assume que os seus conjuntos de treino e validação já estão preparados e acessíveis através da Colab.

Depois de confirmado, pode passar a construir o modelo do zero abaixo.

Passo 2: Defina o Modelo MobileNet do Zero

Esta versão **não** utiliza pesos pré-treinados:

```
baseModel = MobileNetV3Small(weights=Nenhum, include_top=False, input_shape=(224, 224, 3))


x = baseModel.output

x = GlobalMédiaPooling2D()(x)

x = Denso(512, ativação='relu')(x)

x = Denso(256, ativação='relu')(x)
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
x = Denso(128, ativação='relu')(x)

previsões = Denso(3, ativação='softmax')(x)



model = Model(entradas=baseModel.input, outputs=previsões)
```

Passo 3: Compilar o Modelo

Defina o otimizador, a função de perda e a métrica de precisão:

```
model.compile(

    loss="categorical_crossentropy",

    optimizer=Adam(learning_rate=0.0001),

    métricas=['Precisão']

)
```

Porquê esta configuração:

- O Adam otimiza os pesos de forma eficiente
- categorical_crossentropy adequa-se a problemas multiclasse
- A precisão acompanha o desempenho da classificação

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Passo 4: Treinar o Modelo

Agora começamos o treino completo:

```
model.fit(train_generator, validation_data=val_generator,
epochs=10)
```

Dicas:

- Podes aumentar `as épocas` se os resultados não forem satisfatórios.
- O treino a partir do zero pode exigir **mais dados** e **tempos de treino mais** longos do que a aprendizagem por transferência.

Passo 5: Guardar o Modelo

Depois de treinado, guarde o seu modelo para testes futuros ou implementação:

```
model.save('/content/drive/My
Drive/Training/models/flower_classifier_model_NO_TL.keras')
```

Podes carregá-lo mais tarde usando:

```
de tensorflow.keras.models importar load_model

model=load_model('/content/drive/My
Drive/Training/models/flower_classifier_model_NO_TL.keras')
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Passo 6: Avaliação e Previsão do Modelo

Em vez de reescrever estes passos, **reutilize os Passos 9 e 10** do **curso MobileNet Transfer Learning**, que já o orientam per:

- Carregamento do conjunto de testes
- Cálculo da precisão e matriz de confusão
- Previsão de classes para novas imagens
- Interpretação das pontuações de confiança

Estas ferramentas de avaliação mantêm-se válidas quer tenha treinado com ou sem aprendizagem transferida.

11. Critérios de Avaliação:

1. Implementação do Modelo

- Definir e compilar corretamente um modelo MobileNetV3Small do zero (weights=Nenhum), sem usar quaisquer pesos pré-treinados.
- Personalize o modelo adicionando camadas densas e uma saída softmax para classificação multi-classe.

2. Execução do treino

- Treine o modelo usando o conjunto de dados de classificação de flores preparado no Curso 2, dividido em conjuntos de treino e validação.
- Use o ImageDataGenerator para alimentar imagens com aumento básico (por exemplo, rotação).

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- Realizar treino durante um mínimo de 10 épocas, observar e interpretar a precisão/perda de validação.

3. Gravação de modelos

- Guarde o modelo treinado usando o model.save() no caminho correto dentro da estrutura de Drive do aprendiz:
  /TrainingGreeceCV/models/flower_classifier_model_NO_TL.keras

4. Compreensão do Comportamento do Modelo

- Explique as principais diferenças entre treinar de raiz e usar aprendizagem por transferência (em termos de velocidade, desempenho e risco de overfitting).

- Demonstre consciência de porque é que treinar o MobileNet de raiz requer mais dados e épocas.

5. Estrutura e Reutilização

- Reutilizar corretamente a estrutura do conjunto de dados, a montagem do disco e a configuração definidas no curso anterior.

- Mantenha a nomeação, a estrutura dos ficheiros e as práticas consistentes estabelecidas ao longo da série de micro-cursos.

|

---

12. Línguas Disponíveis:

Inglês, **Português** (traduções em progresso)

---

13. Colaboradores:

Fátima Pais, Luís Soares – Designers e Facilitadores de Projetos

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

14. Feedback e Melhoria Contínua:

- **Inquéritos**: No final deste marco, alunos e educadores irão preencher inquéritos de feedback para avaliar a eficácia do projeto.
- **Revisões de Projetos**: Após a conclusão do projeto, será realizada uma reunião final de revisão para discutir potenciais melhorias para futuros projetos baseados em IA.
- **Análise de Dados**: Os dados de feedback serão analisados para identificar pontos fortes e áreas a melhorar, informando projetos futuros.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.