AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Sucessos Criativos, Lda

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

1. Project Title: 4

2. Project Category: 4

3. Project Description: 4

4. Keywords: 4

5. Target Audience: 4

6. Project Objectives: 5

7. Project Timeline: 5

8. Resources Provided: 5

9. Tools and Technologies Used: 6

10. Step-by-Step Implementation Plan: 6

Step 1: Reusing the Setup from Course 2 6

Step 2: Define MobileNet Model from Scratch 7

Step 3: Compile the Model 8

Step 4: Train the Model 9

11. Evaluation Criteria: 10

12. Languages Available: 11

13. Contributors: 11

14. Feedback and Continuous Improvement: 11

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. Project Title:

Training MobileNet Without Transfer Learning

---

2. Project Category:

AI in Education

---

3. Project Description:

In this project, we will train a MobileNetV3-based neural network from scratch (i.e., without pre-trained weights) to classify images of roses, sunflowers, and tulips. Unlike the transfer learning approach covered in the course “Training MobileNet With Transfer Learning”, this method will initialize all model weights randomly and learn entirely from your dataset.

---

4. Keywords:

Image classification, MobileNet, training from scratch, deep learning, Google Colab, Kaggle, flowers dataset, custom dataset, data augmentation, no transfer learning

---

5. Target Audience:

Educators, learners, and makers with or without prior experience who are beginning their journey in AI.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6. Project Objectives:

This micro-course teaches how to build and train a convolutional neural network (CNN) model from scratch using **MobileNetV3Small** without relying on pre-trained weights. The project will focus on classifying images of **roses**, **sunflowers**, and **tulips** using a custom dataset, highlighting the key differences and challenges when working without transfer learning.

By the end of the project, you will be able to:

- Understand how to instantiate MobileNetV3 without pre-trained weights
- Train a lightweight CNN from scratch using a balanced dataset
- Apply image preprocessing and data augmentation to feed the model
- Save and reuse the trained model for inference and testing

This project is meant to **complement** the prior transfer learning course and build on your knowledge of dataset preparation and evaluation techniques.

---

7. Project Timeline:

The estimated duration is approximately **2.5 to 3 hours**.

---

8. Resources Provided:

- For these tasks you can use either a computer or your raspberry pi.
- Dataset reference: Kaggle Flowers Dataset

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

9. Tools and Technologies Used:

Tool / Library	Purpose
Python 3	General programming and scripting
TensorFlow / Keras	Neural network framework for building and training MobileNetV3
MobileNetV3Small	Lightweight CNN architecture used for efficient classification tasks
ImageDataGenerator	Image augmentation and preprocessing
Google Colab	Cloud-based Python notebook environment with GPU support
Matplotlib / Seaborn	(via reuse from Course 2) Used for visualization of training results

10. Step-by-Step Implementation Plan:

Step 1: Reusing the Setup from Course 2

Before continuing, make sure you've already completed the setup steps described in Course 2 – MobileNet with Transfer Learning, including:

Dataset collection and preprocessing
Folder organization and data augmentation

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Google Drive mounting in Colab
Activating Table of Contents for easier navigation
Creating the directory:

/Training/datasets/flowersToTrain/

├── train/

├── val/

└── test/

> If these steps haven't been completed yet, please return to Course " MobileNet with Transfer Learning " and follow them before proceeding. This course assumes your training and validation sets are already prepared and accessible via Colab.

Once confirmed, you can move on to building the model from scratch below.

Step 2: Define MobileNet Model from Scratch

This version **does not** use any pre-trained weights:

```
baseModel = MobileNetV3Small(weights=None, include_top=False, input_shape=(224, 224, 3))


x = baseModel.output

x = GlobalAveragePooling2D()(x)

x = Dense(512, activation='relu')(x)

x = Dense(256, activation='relu')(x)

x = Dense(128, activation='relu')(x)

predictions = Dense(3, activation='softmax')(x)
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
model = Model(inputs=baseModel.input, outputs=predictions)
```

Step 3: Compile the Model

Set the optimizer, loss function, and accuracy metric:

```
model.compile(
    loss="categorical_crossentropy",
    optimizer=Adam(learning_rate=0.0001),
    metrics=['accuracy']
)
```

Why this configuration:

- Adam optimizes weights efficiently
- categorical_crossentropy suits multi-class problems
- accuracy tracks classification performance

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 4: Train the Model

Now we start the full training:

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

Tips:

- You can increase epochs if results aren't satisfactory.
- Training from scratch may require **more data** and **longer training times** than transfer learning.

Step 5: Save the Model

Once trained, save your model for future testing or deployment:

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

You can load it later using:

```
from tensorflow.keras.models import load_model

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

Step 6:Model Evaluation & Prediction

Instead of rewriting these steps, **re-use Steps 9 and 10** from the **MobileNet Transfer Learning course**, which already guide you through:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Loading the test set
- Calculating accuracy and confusion matrix
- Predicting classes for new images
- Interpreting confidence scores

These evaluation tools remain valid whether you trained with or without transfer learning.

---

11. Evaluation Criteria:

1. Model Implementation

- Correctly define and compile a MobileNetV3Small model from scratch (weights=None), without using any pre-trained weights.
- Customize the model by adding dense layers and a softmax output for multi-class classification.

2. Training Execution

- Train the model using the flower classification dataset prepared in Course 2, split into training and validation sets.
- Use ImageDataGenerator to feed images with basic augmentation (e.g., rotation).
- Run training for a minimum of 10 epochs, observe and interpret validation accuracy/loss.

3. Model Saving

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Save the trained model using model.save() into the correct path inside the learner's Drive structure:
  /TrainingGreeceCV/models/flower_classifier_model_NO_TL.keras

4. Understanding Model Behavior

- Explain the key differences between training from scratch and using transfer learning (in terms of speed, performance, and overfitting risk).
- Demonstrate awareness of why training MobileNet from scratch requires more data and epochs.

5. Structure and Reusability

- Properly reuse the dataset structure, drive mounting, and setup defined in the previous course.
- Maintain consistent naming, file structure, and practices established throughout the micro-course series.

|

---

12. Languages Available:

English, **Portuguese** (translations in progress)

---

13. Contributors:

Fátima Pais, Luís Soares – Project Designers and Facilitators

---

14. Feedback and Continuous Improvement:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- **Surveys**: At the end of this milestone, students and educators will complete feedback surveys to assess the effectiveness of the project.
- **Project Reviews**: After project completion, a final review meeting will be held to discuss potential improvements for future AI-based projects.
- **Data Analysis**: Feedback data will be analyzed to identify strengths and areas for improvement, informing future projects.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.