National College of Ireland

Configuration Manual

MSc Research Practicum
MSc Data Analytics

Prashanth Reddy Voladri
Student ID: x23310162

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Prashanth Reddy Voladri		
Student ID:	X23310162		
Programme:	MSc Data Analytics	**Year:**	2024-2025
Module:	MSc (Research)Practicum		
Lecturer:	Vladimir Milosavljevic		
Submission Due Date:	15/09/2025		
Project Title:	Vision-Language Models for Underwater Plastic Detection with Parameter-Efficient Fine-Tuning		
Word Count:	1094	**Page Count:**	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Prashanth Reddy Voladri

Date: 14/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Underwater Plastic Detection with PaliGemma: Complete Setup Guide

Prashanth Reddy Voladri
Student ID: x23310162

Introduction

PaliGemma is Google's cutting-edge multimodal vision-language model that combines powerful vision and language understanding capabilities. This manual provides a comprehensive guide for fine-tuning PaliGemma models using Parameter-Efficient Fine-Tuning (PEFT) methods, specifically LoRA (Low-Rank Adaptation) and DoRA (Weight-Decomposed Low-Rank Adaptation).

What is PaliGemma?
PaliGemma is a multimodal model that can understand both images and text, making it ideal for tasks such as:

- **Object Detection**: Identifying and localizing objects in images
- **Visual Question Answering**: Answering questions about image content
- **Image Captioning**: Generating descriptive text for images
- **Document Understanding**: Extracting information from visual documents

Why Use LoRA/DoRA Fine-Tuning?
Traditional fine-tuning requires updating all model parameters, which is:

- **Memory Intensive**: Requires storing gradients for billions of parameters
- **Computationally Expensive**: Long training times and high GPU costs
- **Storage Heavy**: Each fine-tuned model requires full parameter storage

LoRA and DoRA offer efficient alternatives:

- **LoRA**: Introduces trainable low-rank matrices while keeping pre-trained weights frozen
- **DoRA**: Enhances LoRA by decomposing weights into magnitude and direction components
- **Benefits**: 90%+ memory reduction, faster training, minimal performance loss

Model Variants Covered
This manual covers fine-tuning for:
PaliGemma 1 Series:

- google/paligemma-3b-pt-224 (3B parameters, 224px input)
- google/paligemma-3b-pt-448 (3B parameters, 448px input)
- google/paligemma-3b-pt-896 (3B parameters, 896px input)

PaliGemma 2 Series (Recommended):

- google/paligemma2-3b-pt-224 (Latest 3B model)
- google/paligemma2-3b-pt-448 (Enhanced 3B model)
- google/paligemma2-10b-pt-224 (Large 10B model for best performance)

Hardware Requirements

Minimum Requirements:

- GPU: NVIDIA V100 (32GB) or better
- RAM: 32GB system memory
- Storage: 100GB+ free space

Recommended Setup:

- GPU: NVIDIA A100 (40GB) for optimal performance
- RAM: 64GB+ system memory
- Storage: 500GB+ SSD storage
- Platform: Google Colab Pro+ or dedicated cloud instance

Environment Setup

Google Colab Setup

Prerequisites:

- Google Colab Pro/Pro+ recommended for A100 GPU access
- Google Drive account for model storage
- HuggingFace account for model access

Upload Notebook to Google Colab

Step 1: Access Google Colab

1. Go to colab.research.google.com
2. Sign in with your Google account

Step 2: Upload the Fine-tuning Notebook

1. Click "File" → "Upload notebook"
2. Upload your finetuning_paligemma_LoRA.ipynb notebook
3. Alternatively, you can:
    - Use "File" → "Open notebook" → "GitHub" to import from repository
    - Use "File" → "Open notebook" → "Google Drive" if stored in Drive

Configure A100 GPU Runtime

Step 3: Change Runtime to A100 GPU

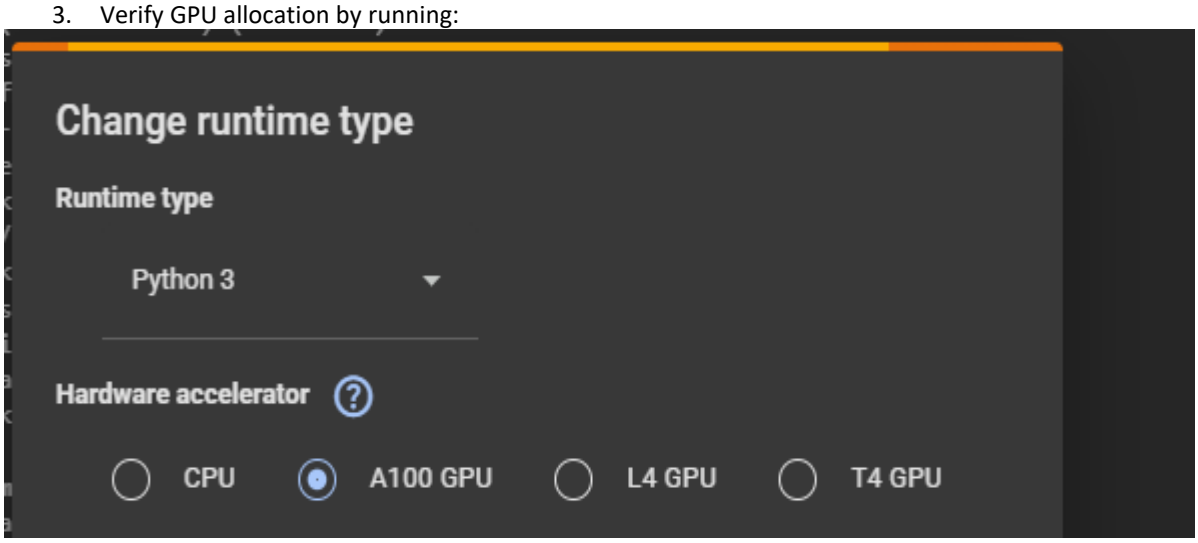
1. **Navigate to Runtime Settings:**
    - Click "Runtime" in the top menu
    - Select "Change runtime type"
2. **Configure Hardware Settings:**
3. Runtime type: Python 3
4. Hardware accelerator: GPU

GPU type: A100 (High-RAM) - if available

5. **Alternative GPU Options (if A100 unavailable):**
    - Primary choice: A100 (40GB VRAM)
    - Secondary choice: V100 (32GB VRAM)
    - Fallback: T4 (16GB VRAM) - for testing only

Step 4: Connect to Runtime Session

1. Click "Connect" button in the top-right corner
2. Wait for the session to initialize (green checkmark appears)
3. Verify GPU allocation by running:



 # Verify GPU setup
!nvidia-smi

```
1  !nvidia-smi

Thu Aug  7 03:20:33 2025
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A100-SXM4-40GB          Off |   00000000:00:04.0 Off |                    0 |
| N/A   32C    P0             47W /  400W |       0MiB /  40960MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Required Library Installation

 # Essential libraries installation
!pip install roboflow supervision seaborn
!pip install transformers accelerate peft datasets
!pip install torch torchvision torchaudio
!pip install tensorboard matplotlib

Key Configuration Points:

- Install order matters for dependency resolution
- Use --upgrade flag for latest versions
- Monitor installation for any conflicts

HuggingFace Authentication

 # HuggingFace Token Configuration
HF_TOKEN = "hf_YOUR_TOKEN_HERE" *# Replace with your token*
login(token=HF_TOKEN)

Security Notes:

- Never commit tokens to version control
- Use environment variables in production
- Token requires read access to gated models

```
[ ]
    PaliGemmaForConditionalGeneration,
    BitsAndBytesConfig,
    Trainer,
    TrainingArguments,
    TrainerCallback
)
from peft import get_peft_model, LoraConfig

# Check GPU
!nvidia-smi

# ====================== CELL 3: AUTHENTICATION WITH YOUR TOKEN ======================
print("Setting up HuggingFace authentication...")

#  YOUR TOKEN IS HERE - DELETE AFTER USE!
HF_TOKEN = "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"  # DELETE THIS LINE AFTER RUNNING!

# Login to HuggingFace
login(token=HF_TOKEN)
print("✅ Logged in to HuggingFace")
```

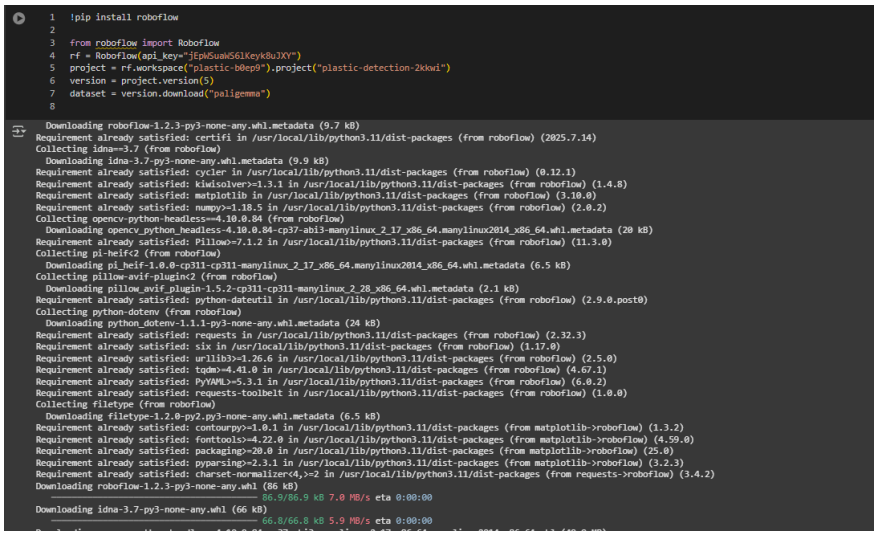
```
Wed Jul 30 13:10:18 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A100-SXM4-40GB          Off |   00000000:00:04.0 Off |                    0 |
| N/A   30C    P0             42W /  400W |       0MiB /  40960MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
Setting up HuggingFace authentication...
✅ Logged in to HuggingFace
```

Dataset Configuration

Dataset Source Setup

Download the current dataset by executing the following



```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="jEpWSuaWS6lKeyk8uJXY")
project = rf.workspace("plastic-b0ep9").project("plastic-detection-2kkwi")
version = project.version(5)
dataset = version.download("paligemma")
```

```
Downloading roboflow-1.2.3-py3-none-any.whl.metadata (9.7 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from roboflow) (2025.7.14)
Collecting idna==3.7 (from roboflow)
  Downloading idna-3.7-py3-none-any.whl.metadata (9.9 kB)
Requirement already satisfied: cycler in /usr/local/lib/python3.11/dist-packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from roboflow) (1.4.8)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from roboflow) (3.10.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.11/dist-packages (from roboflow) (2.0.2)
Collecting opencv-python-headless==4.10.0.84 (from roboflow)
  Downloading opencv_python_headless-4.10.0.84-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (20 kB)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.11/dist-packages (from roboflow) (11.3.0)
Collecting pi-heif<2 (from roboflow)
  Downloading pi_heif-1.0.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.5 kB)
Collecting pillow-avif-plugin<2 (from roboflow)
  Downloading pillow_avif_plugin-1.5.2-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (2.1 kB)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.11/dist-packages (from roboflow) (2.9.0.post0)
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from roboflow) (2.32.3)
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-packages (from roboflow) (1.17.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.11/dist-packages (from roboflow) (2.5.0)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.11/dist-packages (from roboflow) (4.67.1)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.11/dist-packages (from roboflow) (6.0.2)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.11/dist-packages (from roboflow) (1.0.0)
Collecting filetype (from roboflow)
  Downloading filetype-1.2.0-py2.py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (4.59.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->roboflow) (3.2.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->roboflow) (3.4.2)
Downloading roboflow-1.2.3-py3-none-any.whl (86 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.9/86.9 kB 7.0 MB/s eta 0:00:00
Downloading idna-3.7-py3-none-any.whl (66 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.8/66.8 kB 5.9 MB/s eta 0:00:00
```

Google Drive Integration

from google.colab import drive
drive.mount('/content/drive')

Drive Configuration:

python
 # Drive paths in AdvancedTrainingConfig
drive_mount_point: str = "/content/drive"
drive_save_path: str = "MyDrive/G2Testing/PaliGemma_Advanced"

Dataset Analysis

Configurable Dataset Parameters:

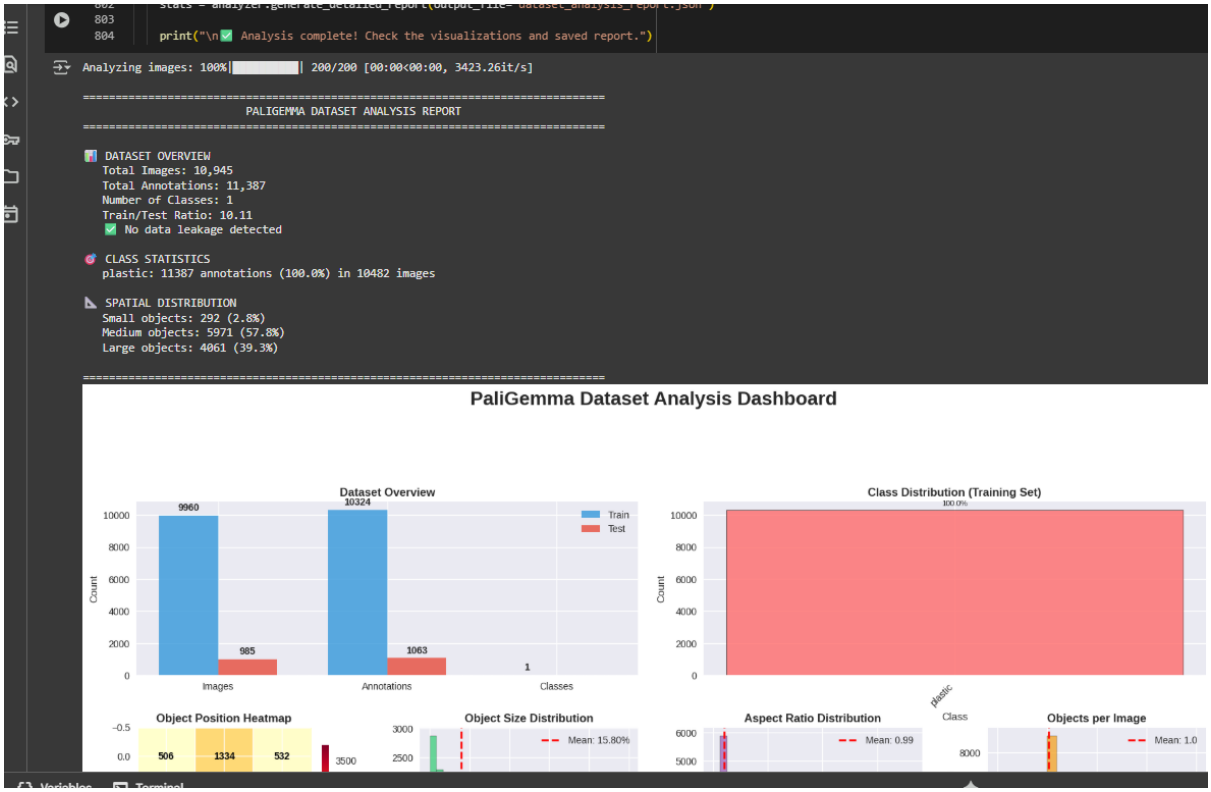
@dataclass
class DatasetConfig:
 # Dataset paths
train_path: str = "dataset/_annotations.train.jsonl"
test_path: str = "dataset/_annotations.test.jsonl"
dataset_root: str = "dataset"

 # Data split configuration
validation_split: float = 0.1 *# 10% for validation*

 # Data augmentation
augment_data: bool = True
shuffle_detections: bool = True
max_detections_per_image: int = 20

Key Parameters to Configure:

Parameter	Default	Range	Description
validation_split	0.1	0.0-0.3	Validation data percentage
max_detections_per_image	20	5-50	Memory optimization
augment_data	True	True/False	Enable data augmentation



Model Configuration

Base Model Selection

PaliGemma Models Available:

 # PaliGemma 1 Models
"google/paligemma-3b-pt-224"
"google/paligemma-3b-pt-448"
"google/paligemma-3b-pt-896"

 # PaliGemma 2 Models
"google/paligemma2-3b-pt-224"
"google/paligemma2-3b-pt-448"
"google/paligemma2-10b-pt-224"

Model Configuration Parameters

```
@dataclass
class ModelConfig:
# Model selection
model_id: str = "google/paligemma2-3b-pt-224"  # Choose based on requirements

# Training method
training_method: str = "lora"  # "lora" or "qlora" or "dora"

# Precision settings
torch_dtype: torch.dtype = torch.bfloat16  # or torch.float16

# Memory optimization
gradient_checkpointing: bool = True
   use_8bit_optimizer: bool = False
```

Model Selection Guide:

Model	VRAM Required	Speed	Accuracy	Use Case
3b-pt-224	24GB	Fast	Good	Quick prototyping
3b-pt-448	32GB	Medium	Better	Balanced performance
2-3b-pt-224	24GB	Fast	Improved	Latest architecture
2-10b-pt-224	40GB+	Slow	Best	Production quality

Training Parameters

Core Training Configuration

```
@dataclass
class AdvancedTrainingConfig:
   # Training duration
num_epochs: int = 3  # CONFIGURABLE: 1-10

   # Batch configuration
   batch_size: int = 4  # CONFIGURABLE: 1-8 (memory dependent)
gradient_accumulation_steps: int = 6  # CONFIGURABLE: 1-16

   # Learning rate settings
learning_rate: float = 1e-4  # CONFIGURABLE: 1e-5 to 5e-4
min_learning_rate: float = 1e-6
warmup_ratio: float = 0.1  # CONFIGURABLE: 0.05-0.2
weight_decay: float = 0.01  # CONFIGURABLE: 0.0-0.1

   # Gradient settings
max_grad_norm: float = 1.0  # CONFIGURABLE: 0.5-2.0
```

Configuration Notes:

- Always test with small configurations first
- Monitor GPU memory usage during training
- Adjust batch size based on available VRAM

- Use validation split for better model selection
- Save configurations for reproducibility

Important Note for All Notebooks

Follow these same configuration steps for all PaliGemma fine-tuning notebooks:

- **LoRA Fine-tuning**: finetuning_paligemma_LoRA.ipynb
- **DoRA Fine-tuning**: finetuning_paligemma_DoRA.ipynb
- **QLoRA Fine-tuning**: finetuning_paligemma_QLoRA.ipynb
- **PaliGemma 2 variants**: Any PaliGemma2 fine-tuning notebooks

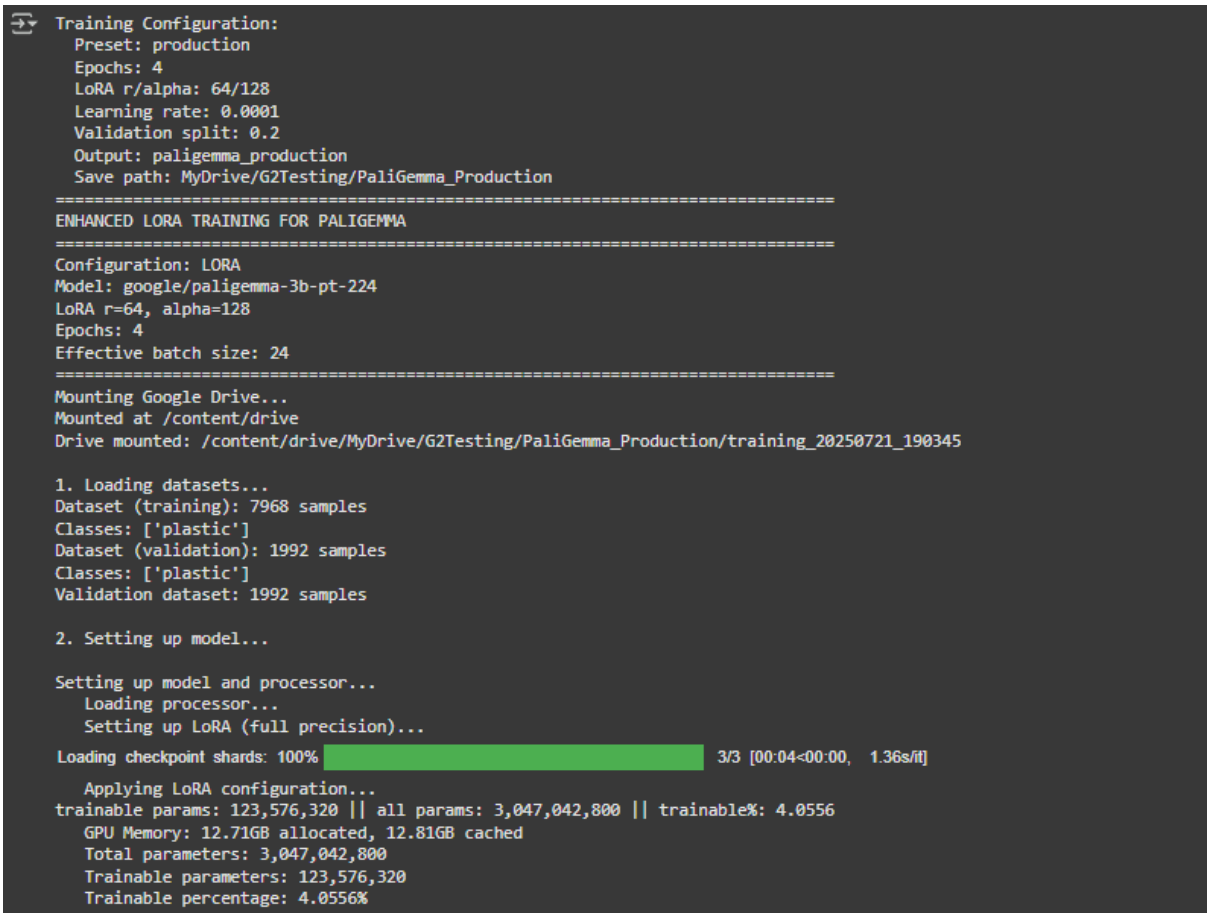
Universal Configuration Process:

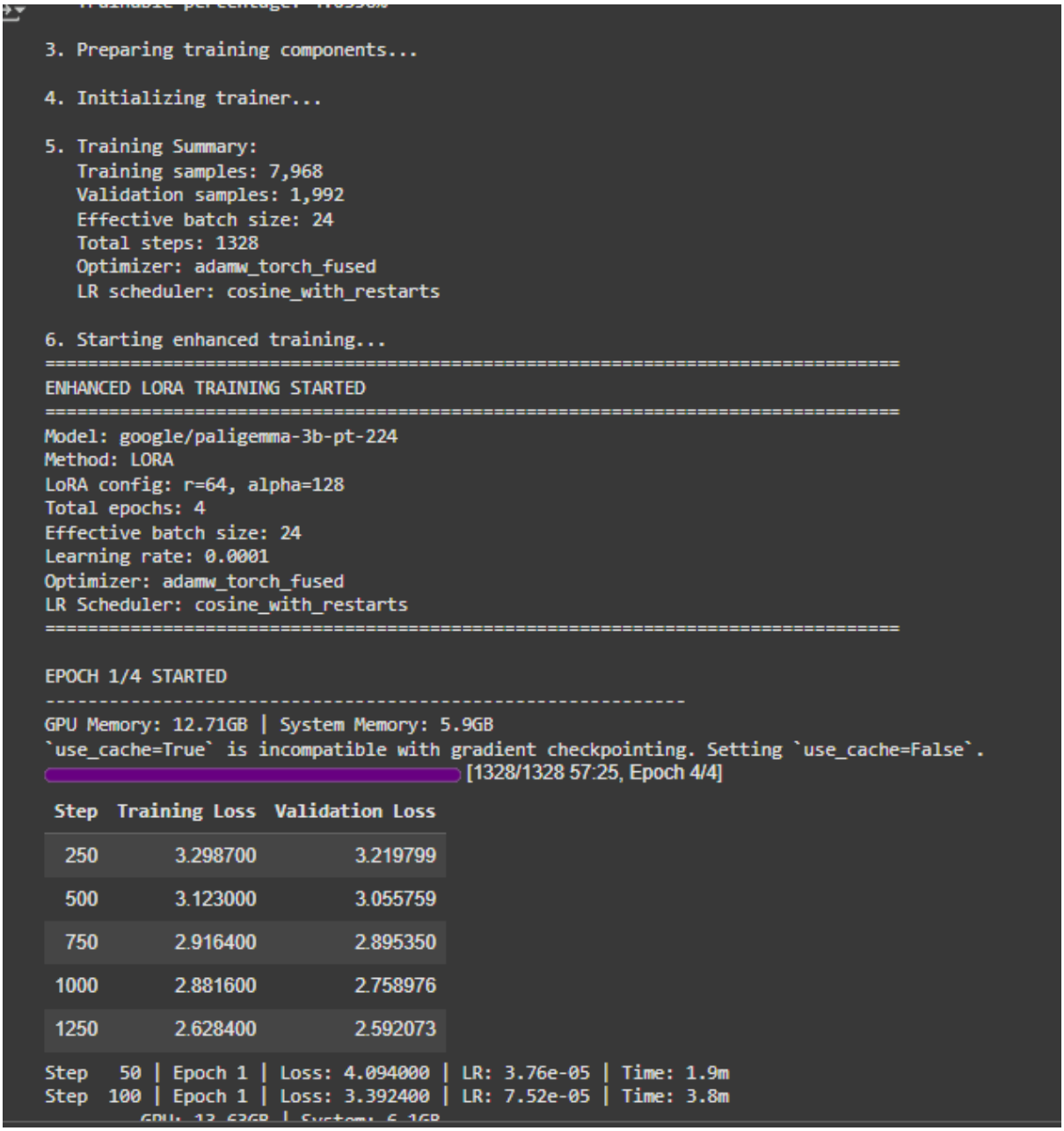
1. **Upload notebook** → Upload your specific notebook to Google Colab
2. **A100 GPU setup** → Change runtime to A100 GPU for all notebooks
3. **Library installation** → Install the same required libraries
4. **HuggingFace authentication** → Use the same token setup
5. **Dataset configuration** → Apply same dataset paths and validation split
6. **Model selection** → Choose appropriate PaliGemma variant
7. **Parameter tuning** → Adjust the same core parameters:
    - num_epochs, batch_size, learning_rate
    - validation_split, gradient_accumulation_steps
    - Method-specific settings (lora_r, dora_r)

Method-Specific Differences:

- **LoRA**: Standard parameter-efficient fine-tuning
- **DoRA**: Enhanced LoRA with weight decomposition
- **PaliGemma 2**: Latest model architecture with improved performance

The configuration principles and steps remain consistent across all methods and model variants.


Training Configuration:
Preset: production
Epochs: 4
LoRA r/alpha: 64/128
Learning rate: 0.0001
Validation split: 0.2
Output: paligemma_production
Save path: MyDrive/G2Testing/PaliGemma_Production
ENHANCED LORA TRAINING FOR PALIGEMMA
Configuration: LORA
Model: google/paligemma-3b-pt-224
LoRA r=64, alpha=128
Epochs: 4
Effective batch size: 24
Mounting Google Drive...
Mounted at /content/drive
Drive mounted: /content/drive/MyDrive/G2Testing/PaliGemma_Production/training_20250721_190345
1. Loading datasets...
Dataset (training): 7968 samples
Classes: ['plastic']
Dataset (validation): 1992 samples
Classes: ['plastic']
Validation dataset: 1992 samples
2. Setting up model...
Setting up model and processor...
Loading processor...
Setting up LoRA (full precision)...
Loading checkpoint shards: 100%
3/3 [00:04<00:00, 1.36s/it]
Applying LoRA configuration...
trainable params: 123,576,320 || all params: 3,047,042,800 || trainable%: 4.0556
GPU Memory: 12.71GB allocated, 12.81GB cached
Total parameters: 3,047,042,800
Trainable parameters: 123,576,320
Trainable percentage: 4.0556%


3. Preparing training components...
4. Initializing trainer...
5. Training Summary:
Training samples: 7,968
Validation samples: 1,992
Effective batch size: 24
Total steps: 1328
Optimizer: adamw_torch_fused
LR scheduler: cosine_with_restarts
6. Starting enhanced training...
ENHANCED LORA TRAINING STARTED
Model: google/paligemma-3b-pt-224
Method: LORA
LoRA config: r=64, alpha=128
Total epochs: 4
Effective batch size: 24
Learning rate: 0.0001
Optimizer: adamw_torch_fused
LR Scheduler: cosine_with_restarts
EPOCH 1/4 STARTED
GPU Memory: 12.71GB | System Memory: 5.9GB
`use_cache=True` is incompatible with gradient checkpointing. Setting `use_cache=False`.
[1328/1328 57:25, Epoch 4/4]
Step Training Loss Validation Loss
250 3.298700 3.219799
500 3.123000 3.055759
750 2.916400 2.895350
1000 2.881600 2.758976
1250 2.628400 2.592073
Step 50 | Epoch 1 | Loss: 4.094000 | LR: 3.76e-05 | Time: 1.9m
Step 100 | Epoch 1 | Loss: 3.392400 | LR: 7.52e-05 | Time: 3.8m