**Assignment: Optimizing the Development & Training Pipeline**

**📌 Overview**

This assignment centers **on modifying and optimizing** a machine learning pipeline managed by the script run\_entire\_dev\_pipeline.py, along with its supporting files. The purpose of this pipeline is to automate the environment setup, data preprocessing, and model training workflow for an image classification of **rock, paper, scissors**.

**🎯 Objectives:**

* Reduce total runtime to **under 1 hour**
* Achieve **model accuracy of 97% or higher**
* Ensure **GPU (CUDA)** is properly utilized (specifically **CUDA 11.8**)
* Remove existing bottlenecks in preprocessing and training

**🔍 Current State of the Pipeline**

**🔧 Script Breakdown:**

* run\_entire\_dev\_pipeline.py:
  + Main orchestrator script that runs the entire ML pipeline in sequence.
  + Calls:
    - create\_virtual\_env.py for environment setup
    - setup\_training.py for data preparation (which utilizes datautils.py)
    - train\_model.py for model training (which utilizes model\_utils.py)
* create\_virtual\_env.py:
  + Creates a Python virtual environment dynamically
  + Installs dependencies in requirements.txt
  + Also handles extraction of **Image\_Dataset.zip**
* dateutils.py:
  + Handles all image preprocessing:
    - Resizing
    - Grayscale conversion
    - Background removal (uses an ONNX model under the hood)
* model\_utils.py:
  + Handles all model construction, training, and export:
    - Defines the model architecture (can be changed if desired)
    - Training pipeline, including some data augmentation
    - Export of final model to an ONNX format

**🚫 Current Limitations:**

* **No GPU support** — all scripts run on CPU by default
* **Background removal** is slow due to CPU-based ONNX inference
* **Training is limited to 5 epochs and poor hyper parameter choices**, resulting in:
  + Low accuracy (~70%)
  + Underfitting
* Total pipeline runtime exceeds **1 hour**

**✅ Optimization Goals**

To meet the assignment requirements, the pipeline must be modified to:

1. ✅ Run on GPU (CUDA 11.8) where possible:
   * Use ONNX runtime with GPU support for preprocessing (background removal)
   * Enable torch with CUDA for training
2. ✅ Improve training configuration:
   * Increase epochs to 15 or more
   * Optimize training loop to use torch.device("cuda")
3. ✅ Reduce runtime for preprocessing:
   * Use batch processing where applicable
   * Leverage GPU inference for background removal
4. ✅ Preserve modularity and simplicity of pipeline

**🆕 Key Script Updates**

**📜 create\_virtual\_env.py (Updated)**

* Sets up a virtual environment
* Installs GPU-compatible versions of libraries (torch, onnxruntime-gpu, rembg)

**📜 run\_entire\_dev\_pipeline.py (Updated)**

* Ensures setup\_training.py and train\_model.py use GPU when available
* Improve training performance overall

**💡 Expected Outcomes**

| **Metric** | **Before Optimization** | **After Optimization** |
| --- | --- | --- |
| **Runtime** | 1h 15m+ | **< 1 hour** |
| **Model Accuracy** | ~70% | **≥ 97%** |
| **Preprocessing Time** | 15–20 min (CPU) | 5–8min (GPU + batch) |
| **Training** | 5 epochs (underfit) | 15+ epochs (GPU-accelerated) |
| **Automation** | Fully automated (for CPU) | Fully automated (GPU-accelerated) |

**📝 Assessment**

| **Metric** | **After Optimization** | **Importance** |
| --- | --- | --- |
| **PreProcessing** | < 20 minutes | 50% |
| **Training** | < 20 minutes | 35% |
| **Accuracy** | 97% or higher | 15% |