Train Image Classification Model with TensorFlow and DVC on EC2

This repository contains scripts and instructions to set up an EC2 instance for training an image classification model using TensorFlow with GPU support.   
The setup includes data versioning using DVC, pulling datasets from S3, and using Conda for managing the environment.

# Prerequisites

- AWS EC2 instance with NVIDIA GPU support (e.g., p2, p3, g4dn instances)  
- Basic knowledge of `git`, `conda`, `dvc`, and `AWS`

# Setup Instructions

## 1. Install NVIDIA Drivers and CUDA Toolkit

Update the system and install NVIDIA drivers with CUDA support:

- sudo dnf update -y  
- sudo dnf install -y dkms kernel-devel kernel-modules-extra  
- sudo dnf config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/repos/amzn2023/x86\_64/cuda-amzn2023.repo  
- sudo dnf clean expire-cache  
- sudo dnf module install -y nvidia-driver:latest-dkms  
- sudo dnf install -y cuda-toolkit  
- sudo reboot

Verify that the NVIDIA drivers and CUDA toolkit are correctly installed:

- nvidia-smi  
- /usr/local/cuda/bin/nvcc --version  
- sudo yum install -y mesa-libGL mesa-libGL-devel libGLU libX11

## 2. Install Miniconda

Install Miniconda to manage Python environments and dependencies:

- wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh  
- bash Miniconda3-latest-Linux-x86\_64.sh  
- source ~/.bashrc  
 conda init

Change owner for any folder (e.g., Miniconda folder):

- sudo chown -R ec2-user:ec2-user /home/ec2-user/miniconda3

## 3. Install TensorFlow with GPU Support using Conda

Create a Conda environment and install TensorFlow with GPU support:

- conda create --name tf python=3.9  
- conda activate tf  
- conda install -c conda-forge cudatoolkit=11.2.2 cudnn=8.1.0  
  
# Create directories to store activation and deactivation scripts  
- mkdir -p $CONDA\_PREFIX/etc/conda/activate.d  
- mkdir -p $CONDA\_PREFIX/etc/conda/deactivate.d  
  
# Add commands to the scripts  
- printf 'export OLD\_LD\_LIBRARY\_PATH=${LD\_LIBRARY\_PATH}  
-export LD\_LIBRARY\_PATH=${LD\_LIBRARY\_PATH}:${CONDA\_PREFIX}/lib/  
' > $CONDA\_PREFIX/etc/conda/activate.d/env\_vars.sh  
- printf 'export LD\_LIBRARY\_PATH=${OLD\_LD\_LIBRARY\_PATH}  
unset OLD\_LD\_LIBRARY\_PATH  
' > $CONDA\_PREFIX/etc/conda/deactivate.d/env\_vars.sh  
  
# Run the script once  
- source $CONDA\_PREFIX/etc/conda/activate.d/env\_vars.sh  
  
pip install --upgrade pip  
pip install tensorflow==2.11  
conda install numpy==1.26  
conda install git

Check that TensorFlow can detect the GPU:

python -c "import tensorflow as tf; print(tf.config.list\_physical\_devices('GPU'))"

## 4. Clone the GitHub Repository

Clone the project repository and install necessary packages:

git clone https://github.com/Kartik-A-1820/train\_model.git  
cd train\_model  
pip install pyyaml==6.0.1  
pip install mlflow  
pip install dvc[s3]   
dvc pull  
pip install scipy==1.12  
pip install opencv-python==4.9.0.80  
pip install keras-tuner  
pip install h5py==3.10.0