1. **Code Creation:** I have developed three Python scripts: load\_data.py, data\_preprocessing.py, and model\_training.py, which serve as the components to be executed within the pipelines. Additionally, I've created an Airflow DAG file named ml\_pipeline.py. To containerize the solution, I've included a Dockerfile and docker-compose.yml.

Dataset: The dataset comprises brain MRI images.

Model Architecture: Utilizing a Convolutional Neural Network (CNN).

1. **Base Image Selection:** For Dockerization, I opted for the Apache Airflow version 2.8.4 base image. However, alternative versions could also be utilized. I've installed all necessary dependencies via a requirements.txt file.
2. **Testing:** I've rigorously tested the solution by running the pipeline end-to-end, closely examining logs and addressing any encountered errors.
3. **Optimization:** Next steps for optimization could include fine-tuning the model hyperparameters, optimizing the data preprocessing steps, implementing distributed training for scalability, and improving CI/CD pipeline efficiency through caching and parallelization. Additionally, optimizing Docker image size and dependency management can enhance deployment efficiency.  
     
   ***Note: I had created a basic pipeline just for the assignment creation.***