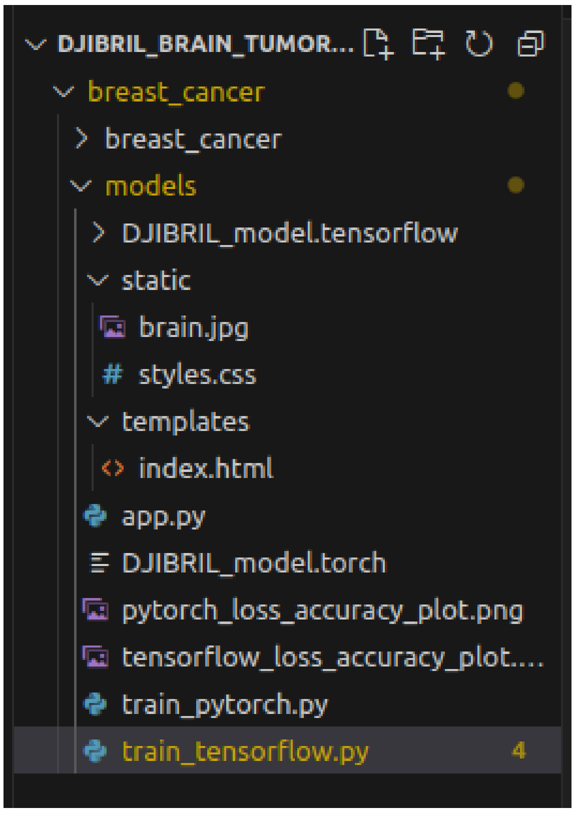
ASSIGNMENT [1] ON [COMPUTER VISION]		
Student's Code		Deadline
[DJIBRIL NDOUR]		**[25/05, 11h 59pm]**
May 25, 2025		2024-2025
		Lecturer: [Dr Jordan Felicien Masakuna]

1 Introduction

This project develops an image classification pipeline for a brain tumor dataset, classifying images as glioma, meningioma, notumor, and pituitary. Two convolutional neural networks (CNNs) were implemented using PyTorch and TensorFlow, trained, and deployed in a web application.



```
DJIBRIL_BRAIN_TUMOR...
  breast_cancer
    breast_cancer
    models
      DJIBRIL_model.tensorflow
      static
        brain.jpg
        styles.css
      templates
        index.html
      app.py
      DJIBRIL_model.torch
      pytorch_loss_accuracy_plot.png
      tensorflow_loss_accuracy_plot....
      train_pytorch.py
      train_tensorflow.py  4
```

Figure 1: architecture

2 Methodology

2.1 Dataset and Preprocessing

The dataset used consists of MRI images brain tumor contains training and testing images in four classes. Images were resized to 224x224 pixels and normalized using

ImageNet statistics. Data augmentation included random flips, rotations, zoom, and shifts to enhance robustness.

2.2 Model Architecture

Pytorch model:
A Convolutional Neural Network was implemented with the following structure:
3 blocks of: Conv2D,BatchNorm,ReLU,MaxPool
Flatten layer
Fully Connected: $128 * 28 * 28$, 512, 4 (output classes)
Dropout (0.5) before final layer
TensorFlow Model
for tensorflow a similar CNN A similar CNN was built with :
3 Conv2D layers (with BatchNorm and MaxPooling)
Flatten layer
Dense(512), Dropout, Dense(4, activation='softmax')
Both models were designed to handle color MRI images with 3 channels (RGB).

2.3 Training

for the training we use for the Loss:
Pytorch:CrossEntropyLoss
TensorFlow: categorical_crossentropy
for the metric we use the accuracy.
Our oupts are saved in : DJIBRIL_model.torch (PyTorch), DJIBRIL_model.tensorflow (TensorFlow)
Training curves were saved to PNG files showing loss and accuracy over epochs.

3 Results

After training both models:

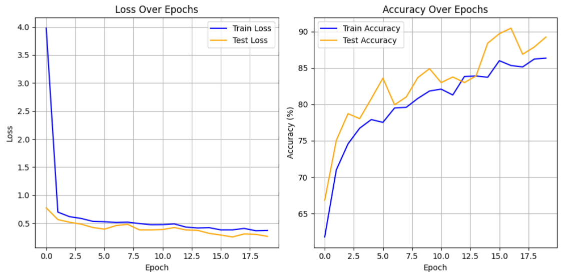
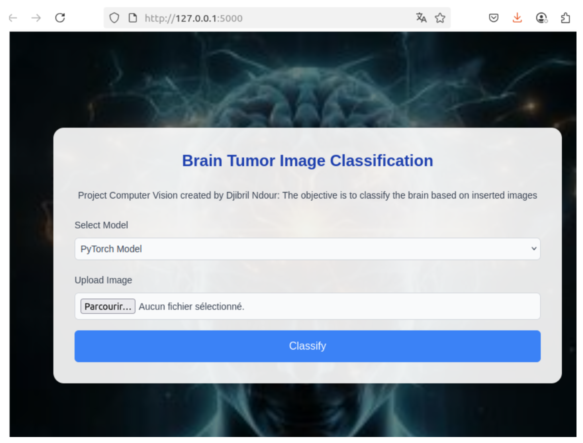


Figure 2: graph loss and accuracy

4 Web Application

A Flask-based web application was developed with a user-friendly interface using Tailwind CSS. Users can select either the PyTorch or TensorFlow model, upload an image, and view the predicted class.The app reads the uploaded image, preprocesses it, runs inference using the selected model, and returns the predicted tumor class. .Error handling ensures robust user interaction.



Project Computer Vision created by Djibril Ndour: The objective is to classify the brain based on inserted images

Figure 3: web application

5 Deployment

Our web application runs loccally via Flask:
python app.py
github repository:mon_project

6 Conclusion

This project demonstrates the effectiveness of deep learning in medical image classification. By combining PyTorch and TensorFlow models with a user-friendly web interface, it offers a practical tool for brain tumor classification.
The solution can assist radiologists and healthcare professionals by automating part of the diagnostic process and improving efficiency.