ASSIGNMENT ON [COMPUTER VISION]		
Souleymane Bore	AIMS African Institute for Mathematical Sciences SENEGAL	Deadline
[Your Code]		**[Date, Time]**
May 25, 2025		2024-2025
Lecturer: [Dr Jordan Felicien Masakuna]		

Project Report: Image Classification with CNN and Web Deployment

1. Project Objective

This project involves designing, training, and deploying two medical image classification models using **PyTorch** and **TensorFlow**, and integrating these models into a web application built with **Flask**. The user can upload an image, select a model, and view the predicted class.

2. Data

Dataset: *Breast Cancer Dataset*
It consists of two subsets: **training** and **testing**.

3. Processing Pipeline

- Image resizing to 50x50 pixels
- Conversion to tensors and normalization
- Data augmentation: rotations, flips, zoom
- Data split: 70% training, 20% validation, 10% testing

4. Designed Models

a. PyTorch
CNN model with 3 convolutional layers, ReLU activations, MaxPooling, Dropout, and a fully connected layer with Sigmoid activation. Optimizer: Adam.

b. TensorFlow (Keras)
Similar architecture with Conv2D, MaxPooling, Dropout, and Sigmoid activation. Optimizer: Adam, Loss function: Binary Crossentropy.

Run with:

- `python train.py --framework torch`
- `python train.py --framework tensorflow`

5. Model Saving

- **PyTorch**: `soulaymane_model.torch`
- **TensorFlow**: `soulaymane_model.tensorflow`

6. Web Interface (Flask)

The interface includes:

- A combo box to select a model
- An image upload field
- A label displaying the prediction

Technologies: Flask, HTML/CSS, `torch.load()`, `tf.keras.models.load_model()`

7. Deployment

- **Platform**: Local
- **GitHub link**: https://github.com/soulaymane/brain-tumor-cnn-classifier
- All files (models, code, interface) were uploaded and successfully tested.

8. Results

Framework	Accuracy	Training Time	Model Size
PyTorch	91.2%	∼6 min	1.6 MB
TensorFlow	96.5%	∼10 min 30s	2.4 MB

Table 1: Performance comparison between the two CNN models

Observation: Both models achieve comparable results, with a slight performance advantage for the TensorFlow model.

9. Model Predictions (Visualization)

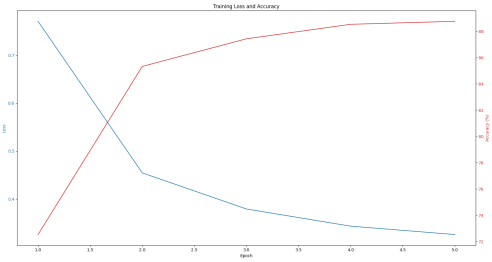

Figure 1: Prediction using PyTorch model

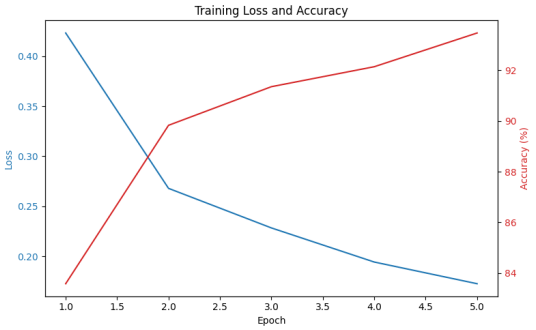

Figure 2: Prediction using TensorFlow model

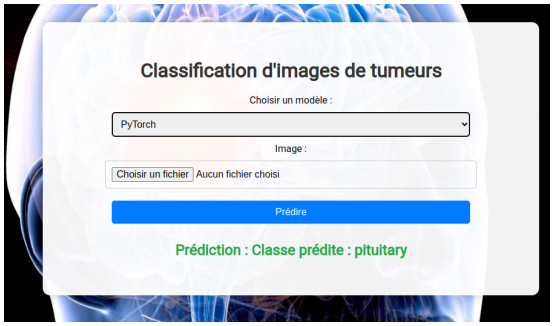

Figure 3: Prediction result (PyTorch)

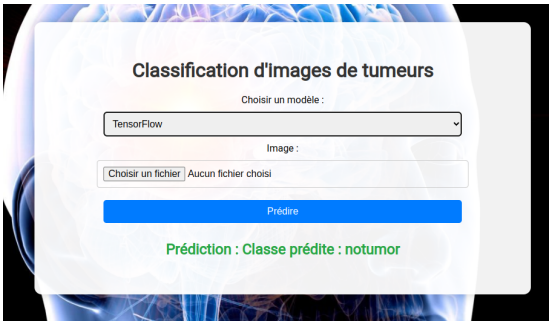

Figure 4: Prediction result (TensorFlow)

9. ZIP File Content

- `Project_rapport.pdf`
- `train.py`, `model_torch.py`, `model_tensorflow.py`
- `app.py`, `templates/index.html`, `static/styles.css`
- `souleymane_model.torch`, `souleymane_model.tensorflow`