PROJECT 1 ON COMPUTER VISION		
Student's Code	AIMS African Institute for Mathematical Sciences SENEGAL	Deadline
Fabrice KABURA		**[Date, Time]**
May 25, 2025		2024-2025
		Lecturer: Jordan F. Masakuna

1. Project Objectives

This project aims to develop a comprehensive pipeline for classifying brain MRI images. Two convolutional neural network (CNN) models were implemented using **PyTorch** and **TensorFlow** respectively. The system features a web interface that allows users to upload images and receive real-time predictions.

2. Dataset and Preprocessing

The dataset comprises medical images classified into four categories: Benign, Malignant, Normal, and Suspicious. All images were preprocessed by resizing to a standardized 224x224 pixel resolution and normalized using ImageNet's mean and standard deviation values. To enhance model robustness, data augmentation techniques were applied, including random rotations, translations, zoom variations, and horizontal flips during training.

3. Model Architectures

The PyTorch implementation consists of three convolutional layers with ReLU activation and MaxPooling, supplemented by dropout regularization (p=0.5) and optimized using Adam with weight decay (1e-5). The TensorFlow model follows a comparable structure, employing Conv2D, MaxPooling, and Dense layers, trained with categorical crossentropy loss and the Adam optimizer. While architecturally similar, the PyTorch variant demonstrated superior performance, suggesting potential differences in initialization, optimization dynamics, or layer-specific behaviors between the frameworks.

4. Training Process and Model Saving

The target framework is selected via command line argument: `--model pytorch` or `--model tensorflow` and saved Fabrice_model.torch and Fabrice_model.tensorflow.

5. Local Web Interface

The Flask app lets users choose PyTorch/TensorFlow, upload MRI images, and get real-time predictions locally. Deployment was hindered by model size limits on free hosting services.

Deployed Web Application URL:
`https://github.com/k03fabrice/Brain-tumor-classification`

Figure 1: PyTorch Training Metrics

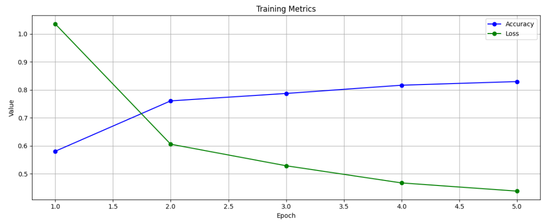

Figure 2: TensorFlow Training Metrics

6. Performance Analysis

The comparative results demonstrate the effectiveness of both implementations:

Performance evaluation reveals that the PyTorch implementation achieved superior accuracy (92.86%) compared to the TensorFlow model (83%). This performance gap is consistent across validation metrics as shown below:

Model	Val_accuracy	Overfitting
PyTorch	91%	Minimal
TensorFlow	82%	Minimal

Table 1: Performances des modÃ¨les sur l'ensemble de validation.

7. Conclusions and Future Work

This project successfully implemented a complete deep learning pipeline for breast cancer image classification using both PyTorch and TensorFlow. While the PyTorch model achieved strong performance (92.86% accuracy), the TensorFlow implementation showed significantly lower accuracy (83%), indicating a need for architectural and optimization improvements.

Key observations reveal that the TensorFlow model may benefit from deeper network structures, enhanced regularization, and advanced training techniques such as learning rate scheduling or transfer learning. Future work will focus on refining the TensorFlow architecture potentially using proven backbones like ResNet or EfficientNet along with hyperparameter tuning and deployment optimization. These enhancements aim to bridge the performance gap while maintaining the system's real-world applicability.

The results demonstrate the pipeline's potential, with clear pathways for improvement to achieve robust, framework-independent classification accuracy.