COMPUTER VISION		
Student's Code	AIMS African Institute for Mathematical Sciences SENEGAL	Deadline
[Ndeye Bineta Dieng]		**[Date, Time]**
May 25, 2025		2024–2025
Lecturer: [Lecturer Name]		

Project 1: Brain Tumor Classification

This project aims to design a complete image classification pipeline for brain MRI tumors (glioma, meningioma, notumor, pituitary). Two CNN models were implemented: one in PyTorch and another in TensorFlow, both based on ResNet50 through *transfer learning*. The goal was to integrate them into a web classification interface.

Pipeline and Models

- **Preprocessing**: resize to 224×224, normalization, RGB conversion.
- **Augmentation**: flips, zoom, rotation.
- **Data split**: balanced train/val/test sets.
- **Optimization**: Adam optimizer, cross-entropy loss, early stopping.

TensorFlow Model (`bineta_model.tensorflow`):

- Based on `ResNet50` with ImageNet weights (frozen layers).
- Additions: GlobalAveragePooling, Dense(128), Dropout, Softmax.

PyTorch Model (`bineta_model.torch`):

- Fine-tuned `ResNet50`.
- Custom head: Dropout - Linear(128) - ReLU - Dropout - Softmax.

Model Results

PyTorch Model

Classification Report:

Class	Precision	Recall	F1-score	Support
Glioma	0.97	0.99	0.98	300
Meningioma	0.99	0.94	0.96	306
No tumor	0.99	1.00	0.99	405
Pituitary	0.98	0.99	0.99	300
Accuracy	**98% (1312 images)**			

TensorFlow Model

- **Overall Accuracy**: **75%** on 1312 images.

Web Interface

- Built with **Flask**, styled using HTML/CSS.
- ComboBox to choose the model (PyTorch or TensorFlow).
- Image upload and classification result display.

GitHub Repository: `https://github.com/opendatame/Bineta_computerv-io`
Deployed Interface (static): `https://opendatame.github.io/Bineta_computerv-io/`
Note: The deployed link shows the interface structure only (frontend via GitHub Pages). Model prediction is not active in this static version, as Flask requires a backend server to run locally.

Conclusion

The PyTorch model achieved strong performance (98% accuracy), while the TensorFlow model remains functional for demonstration (75%). Both were successfully integrated into a user-friendly web application. Future improvements may include deeper fine-tuning, use of specialized neural architectures, and more robust class balancing strategies.