ASSIGNMENT 1 ON COMPUTER VISION		
Student's Code	AIMS African Institute for Mathematical Sciences SENEGAL	Deadline
Cheikh Fall		**[Date, Time]**
May 25, 2025		2024–2025
		Lecturer: [Lecturer Name]

1. Introduction

Brain cancer is one of the most serious diseases. This project develops a web app to classify MRI brain images and detect tumor presence/type using deep learning (PyTorch and TensorFlow), deployed via Streamlit.

2. Objectives

- Train two MRI classifiers (PyTorch and TensorFlow).
- Create a web interface for uploads and predictions.
- Note: Deployment on PythonAnywhere was limited by model size and dependencies.

3. Dataset

MRI scans are divided into 4 classes: Glioma, Meningioma, Pituitary, and No Tumor. Each class is in a separate folder and loaded with `ImageFolder`.

4. Models

PyTorch (ResNet50)

```
model_pytorch.fc = nn.Sequential(
    nn.Linear(2048, 256),
    nn.ReLU(),
    nn.Dropout(0.4),
    nn.Linear(256, num_classes)
)
```

Trained locally, saved as `Cheikh_Fall_model.torch`.

TensorFlow (ResNet50 pretrained)

```
def get_pretrained_model():
    base = ResNet50(weights='imagenet', include_top=False, input_shape=(224,224,3))
    base.trainable = False
    model = Sequential([
        base,
        GlobalAveragePooling2D(),
        Dense(256, activation='relu'),
        Dropout(0.3),
```

```
        Dense(4, activation='softmax')
    ])
    return model
```

Achieved 83% accuracy (loss: categorical_crossentropy, optimizer: Adam).

5. Results on Moons Data

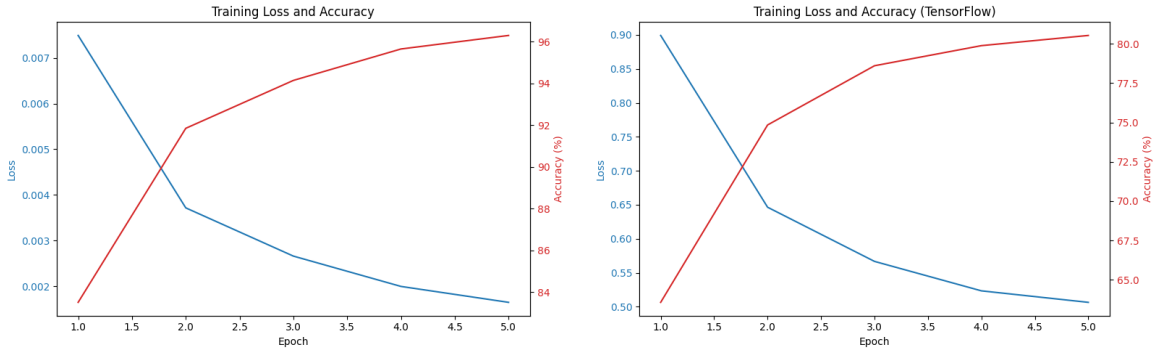

Figure 1: Decision boundaries: PyTorch (97%) vs TensorFlow (83%)

6. Web Application

Interface features:

- Upload MRI image
- Select model
- Display prediction and image

Uses Streamlit; images are processed in-memory and displayed directly without server-side storage. The app is publicly accessible at: `https://braintumorproject.streamlit.app/`.

7. Prediction Pipeline

Preprocessing:

- PyTorch: Resize, tensor, normalize
- TensorFlow: Rescale to [0,1], reshape to (1,224,224,3)

Prediction:

- PyTorch: `torch.max()`
- TensorFlow: `np.argmax()`

8. Limitations & Perspectives

- Models are large and slow to load
- No GPU for inference
- No interpretability tools (e.g., Grad-CAM)

Future work:

- Add Grad-CAM

- Compress with TensorFlow Lite
- Export prediction reports (PDF)

9. Conclusion

We demonstrated an accurate, user-friendly app for brain tumor detection. PyTorch outperformed TensorFlow (97% vs 83%), and the platform serves as a base for future medical imaging tools.