ASSIGNMENT [1] ON COMPUTER VISION		
Student's Code		Deadline
Kouecking Pontiane		**[Date, Time]**
May 25, 2025		2024-2025
Lecturer: Dr Jordan Felicien Masakuna, University of Kinshassa, DRC		

1 Introduction

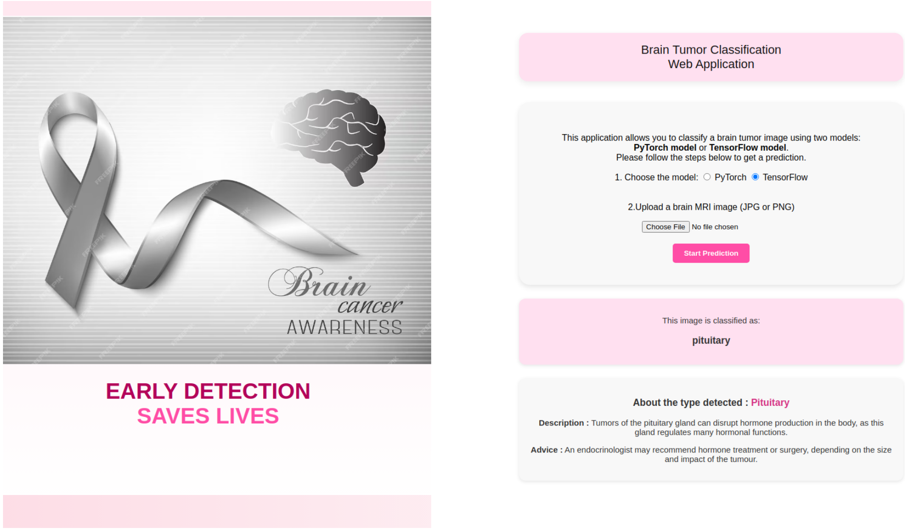
Early detection of brain tumors plays a crucial role in improving patients' chances of survival. This project proposes a web application for classifying brain MRI images into four categories: **glioma**, **meningioma**, **no tumor** and **pituitary tumor**. We combine Deep Learning (Convolutional neural network) techniques with Flask to offer a tool to aid medical diagnosis.

2 Data and Methodology

The data come from a set of cerebral MRI images, divided into four classes: glioma, meningioma, pituitary, notumor. Before use, the images were resized and normalized, and then divided into training and test sets.
Two convolutional neural network (CNN) models have been developed: one with TensorFlow, the other with PyTorch. The architecture comprises several layers: convolution, pooling, dropout and fully connected. The choice of framework is made via a command-line argument.
We first built a PyTorch model and a TensorFlow from scratch model to understand the fundamentals and master each training parameter. These models offered good accuracy, but had one major drawback: once trained, the backup file was very large (over 100 MB) and difficult to manipulate without a GPU, which slowed down deployment.
To remedy this, we adopted a transfer learning strategy for the Tenso model, using MobileNet pre-trained models. This enabled us to significantly reduce file size, save training time and improve overall project stability.

3 Web application development

The web application was built using Flask. It allows the user to upload an image, which is then pre-processed (resized, normalized) before being submitted to the model for prediction. The user interface, designed with HTML and CSS, adopts a two-column layout: an illustrative image on the left and a submission form on the right.
Once the prediction has been made, a description and a few hints are given about the type of prediction. This are the result of the local deployment



Brain Tumor Classification
Web Application

This application allows you to classify a brain tumor image using two models:
PyTorch model or **TensorFlow model**.
Please follow the steps below to get a prediction.

1. Choose the model: ○ PyTorch ● TensorFlow

2.Upload a brain MRI image (JPG or PNG)

This image is classified as:

pituitary

About the type detected : Pituitary

Description : Tumors of the pituitary gland can disrupt hormone production in the body, as this gland regulates many hormonal functions.

Advice : An endocrinologist may recommend hormone treatment or surgery, depending on the size and impact of the tumour.

Figure 1: Interface of our application

This link give the deposite github `https://github.com/Pontiane21K/brain_classif_web_app`

4 Difficulties encountered

The project went through several stages of adjustment and recurring technical difficulties:

- We initially trained a TensorFlow model from scratch. However, this model was very resource-intensive (notably without GPU access), with a large output file.
- Deployment on platforms such as PythonAnywhere proved difficult due to file size limitations (100 MB) and the absence of a GPU.
- When loading the saved model, we encountered compatibility errors due to changes in network architecture (incompatible dimensions).
- The other alternative was to deploy on render, but using GitHub caused problems when pushing large files (greater than 100 MB), generating errors and connection interruptions.

5 Conclusion

This project has given us comprehensive hands on experience in computer vision: data preparation, CNN model design and training, integration into a web interface, and management of compatibility and deployment issues. The adjustments we made demonstrate the importance of flexibility and mastery of the development environment in the success of this type of project.