**Naomi Berumen**

**Chisom Nwanonenyi**

**Adeyemi Kolawole**

**ITAI-3377**

**L02**

**Conceptual Report: Converting and Deploying AI Models using TensorFlow Lite**

Course: AI & Robotics - Data Science Applications

Lab Title: Practical AI Model Deployment with TensorFlow Lite

## **Introduction**

This lab was a deep dive into the real-world process of deploying AI models, and I have to say—it was both exciting and challenging. The goal was to train a neural network, convert it to TensorFlow Lite, and successfully run inference, which seems straightforward at first. But once I started running into conversion errors, I quickly realized that real-world AI deployment isn't always a smooth ride.

## **Challenges Faced**

I ran into a few frustrating roadblocks along the way, but each one ended up being a great learning experience.

TensorFlow Lite Conversion Issues:

* Initially, I saved the model in .h5 format, but TensorFlow Lite didn't like it. It threw errors about missing attributes.
* After troubleshooting, I learned that the latest versions of TensorFlow prefer SavedModel format over .h5 for conversion. Switching to .export() finally made things work!

Attribute Errors When Converting the Model:

* The first few attempts at conversion resulted in the dreaded AttributeError: 'Sequential' object has no attribute '\_get\_save\_spec'.
* After some research and trial-and-error, I found that running a dummy prediction on the model before converting initialized the input shapes properly. That small change fixed the issue.

TensorFlow Version Compatibility Issues:

* At one point, I thought I did everything right, but the conversion still failed.
* Checking my TensorFlow version helped—I needed to upgrade to the latest version to make sure the experimental converter worked correctly.

## **What We Learned**

This lab made me appreciate the deployment side of AI way more. It's one thing to train a model, but getting it to work in a production-like setting is a whole different challenge. Here are my biggest takeaways:

* TensorFlow Lite isn't just a plug-and-play solution—you need to understand how models are structured and saved.
* Preprocessing and input shape configuration is crucial—a model might work fine during training but fail at deployment if the input tensor isn't properly defined.
* Debugging AI models takes patience—a lot of the errors I encountered weren’t straightforward, but breaking them down step by step helped me understand how to troubleshoot effectively.

## Real-World Applications of TensorFlow Lite

The biggest reason why TensorFlow Lite is exciting is because it allows AI to run on edge devices like smartphones, IoT gadgets, and even microcontrollers. I can definitely see how this could be used for:

* Handwriting recognition apps (like scanning receipts and documents).
* AI-powered embedded systems (maybe even robotics projects!).
* Offline AI models that don’t require cloud processing (great for privacy-focused apps).

## **Final Thoughts**

Even though this lab took longer than expected due to troubleshooting, I think the problem-solving process was just as valuable as the final result. Now, I feel much more comfortable working with TensorFlow Lite and debugging model conversion issues.