

L03_Eduardo_Cabrera-Lopez_ITAI3377 - Simulation Documentation

Edge Impulse Model Upload

Following TensorFlow Lite conversion of the trained model, I uploaded the model.tflite file to Edge Impulse platform via online dashboard. This stage was required since the CLI, edge-impulse-uploader, only enables uploading datasets including photos, audio, and sensor data—not model files.

Edge Impulse immediately examined the model and reported its performance traits over several hardware profiles once the file was uploaded under the Deployment > Custom Models section. For systems like the Cortex-M4F, Cortex-M7, and Raspberry Pi 4, this covered measures including RAM use, ROM size, and inference delay.

These revelations help one decide whether a model fits for use on limited edge devices. In this scenario, the model displayed low latency and memory use and satisfied the criteria for multiple hardware configurations. This verified the success of the conversion procedure and the good optimization of the model for edge situations.

The performance of the uploaded model on Edge Impulse's evaluation dashboard is succinctly shown in the screenshot below.

The screenshot displays the Edge Impulse web interface. On the left is a navigation sidebar with options like Dashboard, Devices, Data acquisition, Experiments, and Impulse #2. The main area is titled 'Impulse #2' and shows 'Step 2: Process "model.tflite"'. It includes configuration fields for Model input (Image), input shape (28, 28, 1), input scaling (Pixels ranging 0..1), Resize mode (Fit shortest axis), Model output (Classification), and Output labels (10). A 'Save model' button is at the bottom right of this section. To the right, the 'On-device performance' section shows two tables: 'MCUs' and 'Microprocessors'. The 'MCUs' table compares EON Compiler and TFLite performance across Low-end and High-end MCUs, with and without AI acceleration. The 'Microprocessors' table shows performance for CPU and GPU/accelerator. Below these is a 'Check model behavior' section with an 'Upload an image' button.

DEVICE	LATENCY	EON COMPILER		TFLITE	
		RAM	ROM	RAM	ROM
Low-end MCU	9,862 ms.	66.6K	2.7M	130.1K +43.9%	2.7M +18.0K
High-end MCU	128 ms.	66.6K	2.7M	130.3K +43.6%	2.7M +26.4K
+ AI accelerator	128 ms.	66.6K	2.7M	130.3K +43.6%	2.7M +26.4K

DEVICE	LATENCY	MODEL SIZE
CPU	5 ms.	2.6M
GPU or accelerator	1 ms.	2.6M

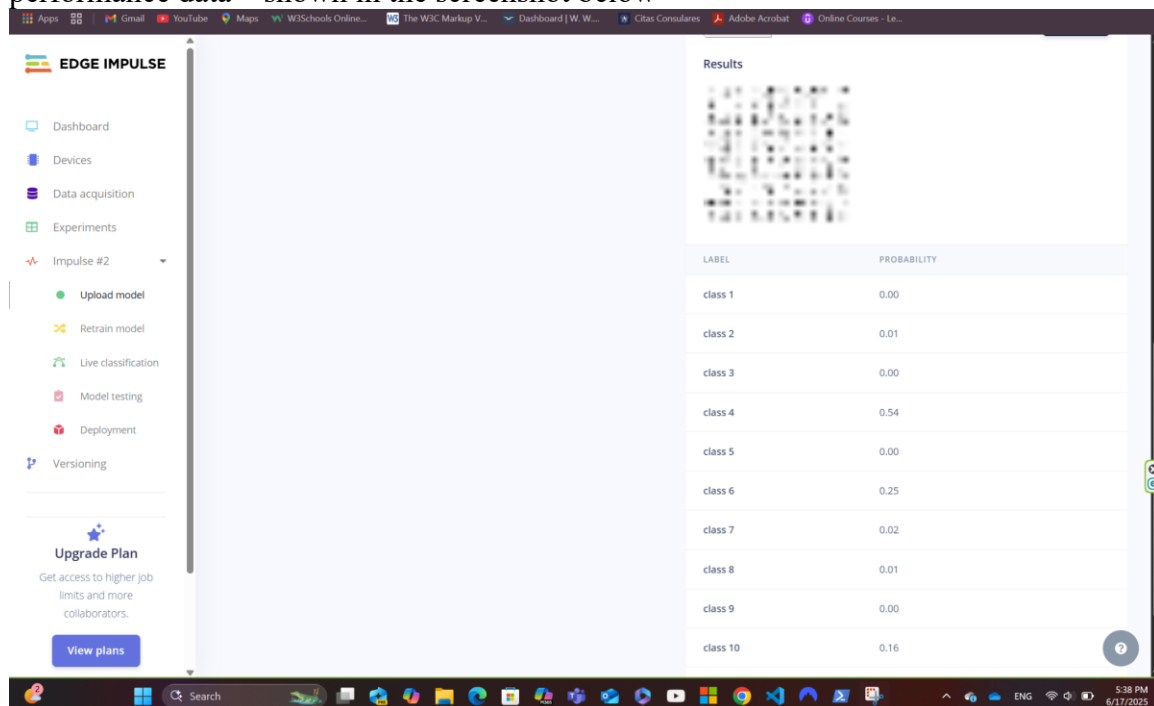
Inference Conventions

Following a successful Edge Impulse upload of the transformed TensorFlow Lite model, I tested it with the Live Classification tool included on the platform. By means of supplied sample data, this tool enables real-time testing of a model, therefore imitating the performance of the model in an actual deployment context.

I supplied a 28x28 grayscale picture of a handwritten digit from the MNIST collection for this test. The algorithm produced a prediction of class 4 with a confidence score of 54% from the image, which portrayed number 4. The model maintained its generalizing capacity after being transformed to the TensorFlow Lite version, therefore even if the confidence score was modest, the prediction was accurate.

Furthermore, the platform revealed that the model performed effectively on simulated edge devices by offering performance measures including inference time and resource use. This verified both the model's practicality for real-time embedded hardware decision-making and its accuracy.

The classification output—including the predicted label, confidence score, and performance data—shown in the screenshot below



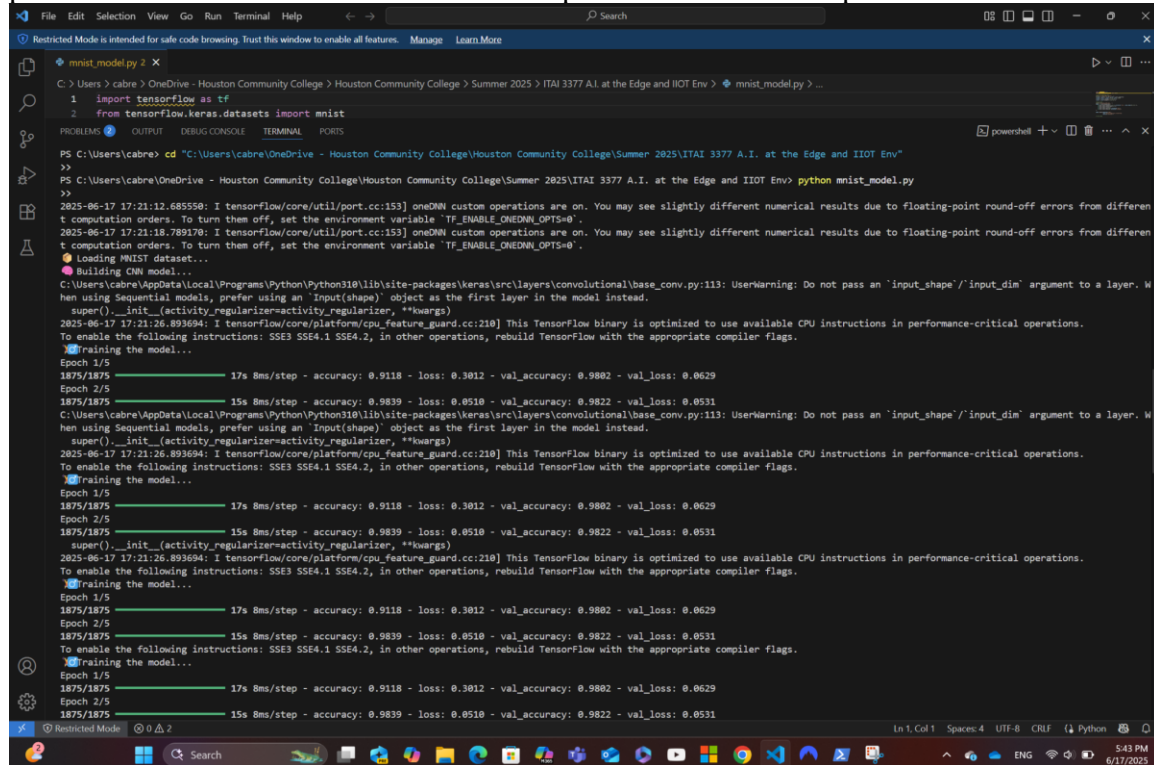
Virtual Code Model Training

The model training procedure carried out inside Visual Studio Code is seen below in a screenshot. Using a CNN developed using TensorFlow and Keras, the training was conducted on the MNIST dataset. Over five epochs, the model was taught to categorize 28x28 grayscale images of handwritten numerals ranging from 0 to 9.

For both the training and validation datasets, the terminal showed live updates on important performance measures—including loss and accuracy—all through the training process. The model showed good generalization to unseen data by the last epoch by reaching a validation accuracy of roughly 98.2%. The loss values dropped gradually over epochs, implying that the model was learning efficiently free from any evidence of overfitting.

This phase was crucial to verify proper configuration of the data preparation and architectural elements. Strong performance of the model prepared the path for effective conversion to TensorFlow Lite and later Edge Impulse platform deployment.

The screenshot offers a graphic record of the training logs, thereby verifying the good performance of the model and the correct operation of the development environment.



```
File Edit Selection View Go Run Terminal Help
mnist_model.py 2 X
C:\Users\cabre> OneDrive - Houston Community College> Houston Community College> Summer 2025\ITAI 3377 A.I. at the Edge and IIOT Env> mnist_model.py ...
1 import tensorflow as tf
2 from tensorflow.keras.datasets import mnist

PS C:\Users\cabre> cd "C:\Users\cabre\OneDrive - Houston Community College\Houston Community College\Summer 2025\ITAI 3377 A.I. at the Edge and IIOT Env"
>>
PS C:\Users\cabre\OneDrive - Houston Community College\Houston Community College\Summer 2025\ITAI 3377 A.I. at the Edge and IIOT Env> python mnist_model.py
>>
2025-06-17 17:21:12.685550: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from differen
t computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-06-17 17:21:12.789170: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from differen
t computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
Loading MNIST dataset...
Building CNN model...
C:\Users\cabre\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\convolutional\base_conv.py:113: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. W
hen using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2025-06-17 17:21:26.893694: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Training the model...
Epoch 1/5
1875/1875 17s 8ms/step - accuracy: 0.9118 - loss: 0.3012 - val_accuracy: 0.9802 - val_loss: 0.0629
Epoch 2/5
1875/1875 15s 8ms/step - accuracy: 0.9839 - loss: 0.0510 - val_accuracy: 0.9822 - val_loss: 0.0531
C:\Users\cabre\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\convolutional\base_conv.py:113: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. W
hen using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2025-06-17 17:21:26.893694: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Training the model...
Epoch 1/5
1875/1875 17s 8ms/step - accuracy: 0.9118 - loss: 0.3012 - val_accuracy: 0.9802 - val_loss: 0.0629
Epoch 2/5
1875/1875 15s 8ms/step - accuracy: 0.9839 - loss: 0.0510 - val_accuracy: 0.9822 - val_loss: 0.0531
C:\Users\cabre\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\convolutional\base_conv.py:113: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. W
hen using Sequential models, prefer using an 'Input(shape)' object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2025-06-17 17:21:26.893694: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Training the model...
Epoch 1/5
1875/1875 17s 8ms/step - accuracy: 0.9118 - loss: 0.3012 - val_accuracy: 0.9802 - val_loss: 0.0629
Epoch 2/5
1875/1875 15s 8ms/step - accuracy: 0.9839 - loss: 0.0510 - val_accuracy: 0.9822 - val_loss: 0.0531
```

Environment Setup: TensorFlow Installation

The TensorFlow package and its required dependencies were effectively installed in the local Python environment with pip package manager, as the screenshot below attests to. Creating the development environment for model building and training depends critically on this phase.

The installation method consisted in running the command:

```
pip install tensorflow
```

Once running, the terminal showed a sequence of messages indicating the installation and configuration of several TensorFlow components—including NumPy, protobuf, keras, and other necessary packages. Accessing high-level APIs used for loading data, building the neural network, training the model, and converting it to TensorFlow Lite format required correct TensorFlow installation.

This setup stage also guaranteed consistency among other dependencies, TensorFlow version, and Python interpreter. I could move boldly with coding and training the AI model in Visual Studio Code knowing the successful installation was confirmed by terminal output.

The screenshot provides proof that before development the fundamental tools for the AI pipeline were correctly set and installed.

```
S C:\Users\cabre\OneDrive - Houston Community College\Houston Community College\Summer 2025\ITAI 3377 A.I. at the Edge and IIOT Env> pip install tensorflow
ERROR: Could not find a version that satisfies the requirement tensorflow (from versions: none)
ERROR: No matching distribution found for tensorflow

S C:\Users\cabre\OneDrive - Houston Community College\Houston Community College\Summer 2025\ITAI 3377 A.I. at the Edge and IIOT Env> pip install tensorflow
Requirement already satisfied: tensorflow in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (2.19.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (2.3.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast>=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (25.0)
Requirement already satisfied: protobuf<4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.dev,>=3.20.3 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (5.29.5)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (2.32.4)
Requirement already satisfied: setuptools in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (65.5.0)
Requirement already satisfied: six>=1.12.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (4.14.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (1.73.0)
Requirement already satisfied: tensorboard>=2.19.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (2.19.0)
Requirement already satisfied: keras>=3.5.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (3.10.0)
Requirement already satisfied: numpy<2.2.0,>=2.0.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (2.1.3)
Requirement already satisfied: h5py>=3.11.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (3.14.0)
Requirement already satisfied: ml-dtypes<1.0.0,>=0.5.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (0.5.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.29.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorflow) (0.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (2025.6.15)
Requirement already satisfied: markdown>=2.6.8 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorboard>=2.19.0->tensorflow) (3.8)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorboard>=2.19.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from tensorboard>=2.19.0->tensorflow) (3.1.3)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.45.1)
Requirement already satisfied: rich in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from keras>=3.5.0->tensorflow) (14.0.0)
Requirement already satisfied: namex in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from keras>=3.5.0->tensorflow) (0.1.0)
Requirement already satisfied: optree in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from keras>=3.5.0->tensorflow) (0.16.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from werkzeug>=1.0.1->tensorboard>=2.19.0->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from rich->keras>=3.5.0->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0,>=2.13.0 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from rich->keras>=3.5.0->tensorflow) (2.19.1)
Requirement already satisfied: mdurl>=0.1 in c:\users\cabre\appdata\local\programs\python\python310\lib\site-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow) (0.1.2)

S C:\Users\cabre\OneDrive - Houston Community College\Houston Community College\Summer 2025\ITAI 3377 A.I. at the Edge and IIOT Env>
```

Model: Tflite File Confirmation

The successful production and presence of the model.tflite file in the working directory is confirmed by the snapshot below. Convert from Keras format to TensorFlow Lite (TFLite) using the TFLite converter, this file shows the last, refined version of the trained Convolutional Neural Network model.

Since it greatly lowers model size and increases runtime efficiency, the conversion procedure is a vital stage in getting AI models ready for use on edge devices. Specifically made to function on devices with minimal processing capability, such microcontrollers, IoT devices, and smartphones, TensorFlow Lite models are

The Python script modified the model using these lines of code:

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)

tflite_model = converter.convert()
with open('model.tflite', 'wb') as f:
    f.write(tflite_model)
```

As seen in the screenshot, the model.tflite file showed up in the project directory following running this block. This verified that Edge Impulse or any other platform supporting TFLite models was ready for the model to be uploaded and used.

This phase proved not only the technical flow but also my grasp of how to move from model creation to deployment-ready forms.

ITAI 3377 A.I. at the Edge and

OneDriveHouston Community CollegeSummer 2025ITAI 3377 A.I. at the Edge and IIOT Env

Search ITAI 3377 A.I. at the Edge and IIOT Env

NewSortViewDetails

AI AgentAppsAttachmentsCabrer Lopez Data Science MidTierDesktopDocumentsDownloadsFE Project LeviathanHouston CommunityMicrosoft Copilot MidTermNLP MidTermOPPA AI AssistantPicturesproduct_descriptionScreenshotsDesktopDownloadsDocumentsPicturesMusic

Name	Status	Date modified	Type	Size
ipynb_checkpoints		6/10/2025 4:17 PM	File folder	
venv		6/15/2025 10:16 PM	File folder	
A01_Eduardo Cabrera-Lopez_Duo Sync_ITAI...		6/8/2025 5:26 PM	Microsoft Word Doc...	19 KB
A01_Eduardo Cabrera-Lopez_Duo Sync_ITAI...		6/8/2025 5:26 PM	Chrome PDF Docum...	118 KB
A03_Eduardo Cabrera-lopez_ITAI_3377		6/15/2025 7:55 PM	Microsoft Word Doc...	20 KB
A03_Eduardo Cabrera-lopez_ITAI_3377		6/15/2025 7:55 PM	Chrome PDF Docum...	82 KB
L02_Cabrer Lopez_Eduardo_Conceptual Desi...		6/10/2025 9:02 PM	Microsoft Word Doc...	16 KB
L02_Cabrer Lopez_Eduardo_Conceptual Desi...		6/10/2025 9:02 PM	Chrome PDF Docum...	121 KB
L02_Cabrer Lopez_Eduardo_ITAI3377		6/10/2025 8:49 PM	Microsoft Word Doc...	16 KB
L02_Cabrer Lopez_Eduardo_ITAI3377		6/10/2025 8:51 PM	Chrome PDF Docum...	67 KB
mnist_modelh5		6/10/2025 4:20 PM	H5 File	1,216 KB
mnist_model		6/16/2025 10:05 PM	Python Source File	2 KB
mnist_modeltfite		6/10/2025 4:20 PM	TF Lite File	400 KB
MnistExamples		6/16/2025 10:23 PM	PNG File	70 KB
modeltfite		6/17/2025 5:22 PM	TF Lite File	2,714 KB
Untitled		6/10/2025 4:27 PM	Jupyter Source File	33 KB