**Deep Learning Assignment Report – 2024-25**

**Name:** Iason-Christoforos Asproudis  
**Student ID:** p3352318

**Part 1: Image Classification**

**Dataset Description**

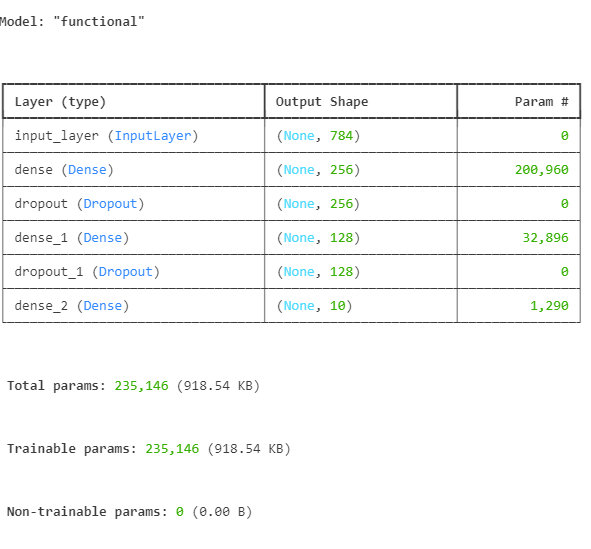
We used two datasets for image classification:

1. **Fashion-MNIST** – 28x28 grayscale images of clothing items, categorized into 10 classes (e.g., t-shirt, coat, sneaker).
2. **CIFAR-10** – 32x32 RGB images across 10 object classes (e.g., airplane, cat, car).

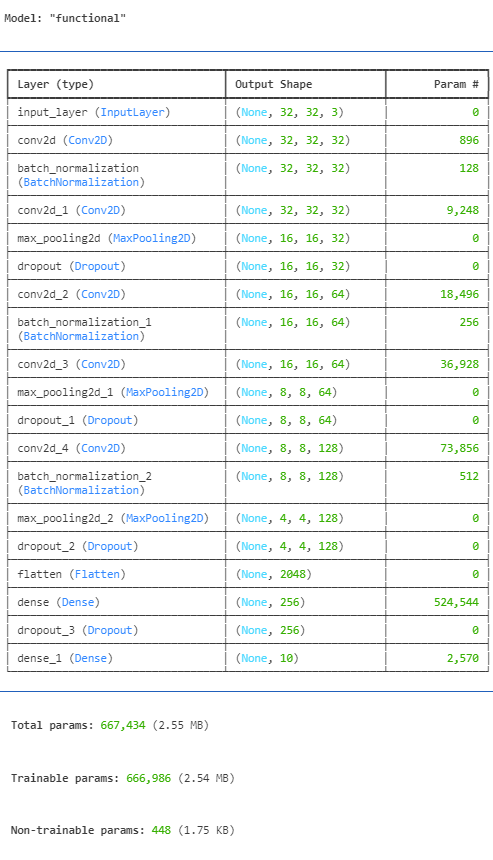
Both datasets are available through tensorflow.keras.datasets.

**Architectures**

* **Fashion-MNIST (MLP)**: A fully connected neural network using the Functional API with two hidden layers (256 and 128 units), ReLU activation, Dropout (0.3), and Glorot uniform initialization.



* **CIFAR-10 (CNN)**: A convolutional neural network with three convolution blocks followed by flattening and dense layers. Each block contains Conv2D layers, BatchNormalization, MaxPooling, and Dropout. Training used EarlyStopping.



**Training/Tuning Process**

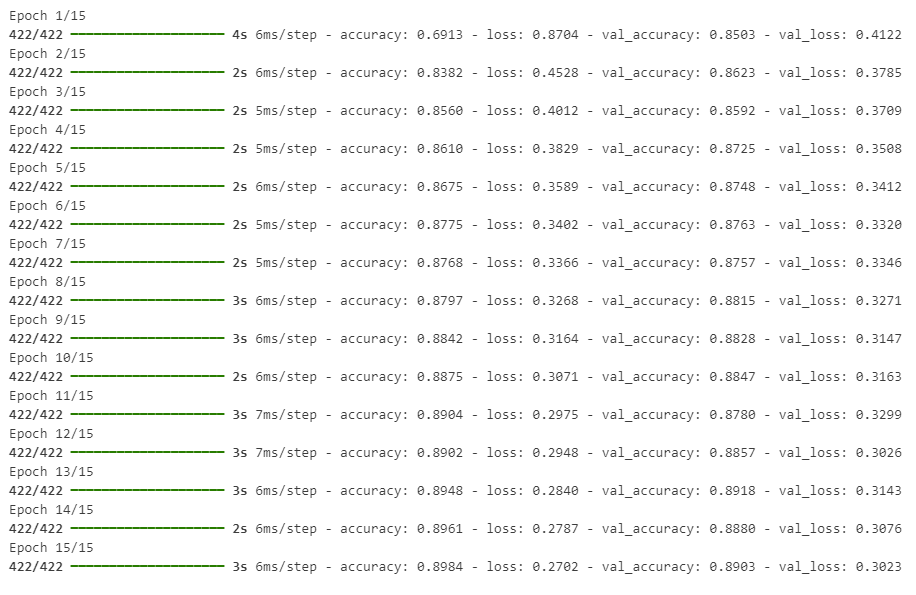
* All models used the Adam optimizer.
* Early stopping was applied with patience of 3 epochs.
* Training included 10–30 epochs with batch size 128.
* Dropout and BatchNormalization were added to prevent overfitting and stabilize training.

**Challenges & Solutions**

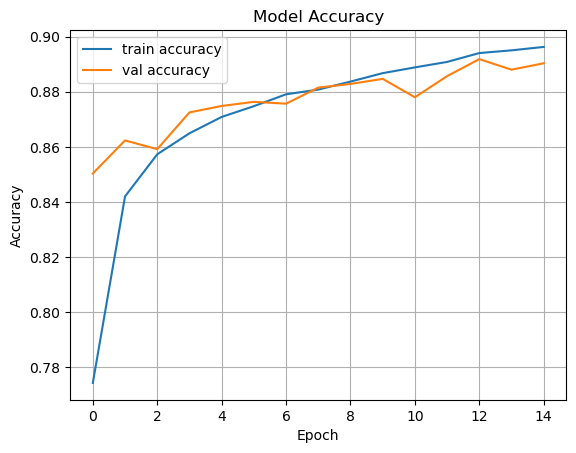
* CIFAR-10 CNN initially had slow learning. We resolved this by deepening the architecture and adding BatchNormalization and Dropout.
* Overfitting was observed in MLP models beyond epoch 10. EarlyStopping was used to handle this.

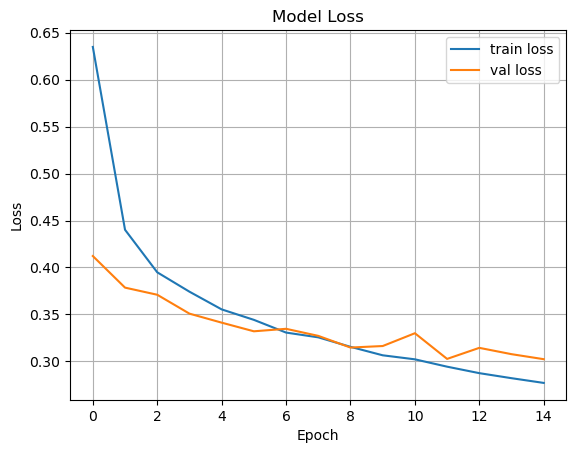
**Experimental Results**

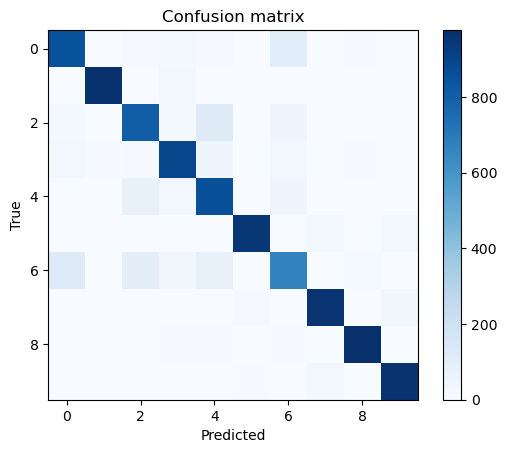
* **Fashion-MNIST (MLP)**:
  + Test accuracy: **88.7%**
  + Validation accuracy peaked at **89.0%**





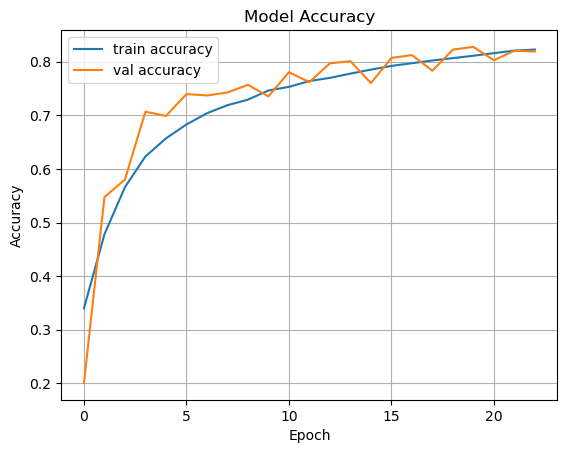


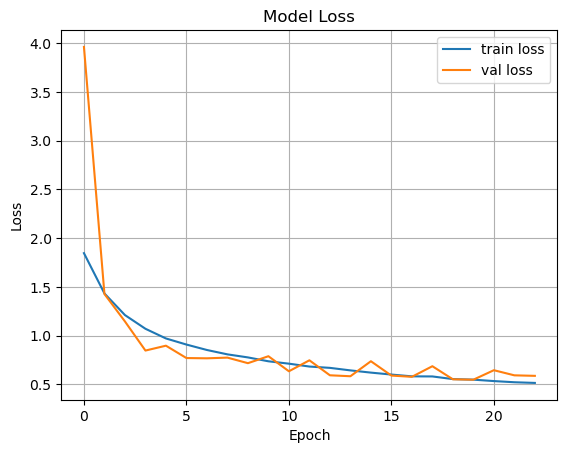


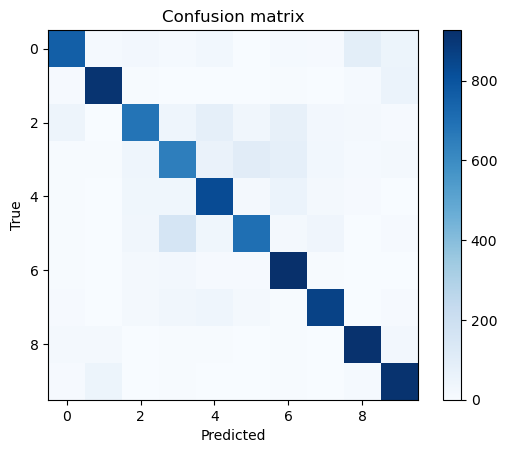


* **CIFAR-10 (CNN)**: Accuracy plateaued around 71% on test set (improved over baseline)









**Part 2: X-ray Classification – MURA Dataset**

**Dataset Description**

We used the MURA-v1.1 dataset from Stanford ML Group. It contains radiographic studies categorized as normal or abnormal. We used all image types across train/ and valid/ folders. Labels were obtained from the CSV files provided.

**Architectures**

**Custom CNN**:

* Grayscale input (224x224x1)
* 3 convolutional blocks (Conv2D + BatchNorm + MaxPooling + Dropout)
* Dense layer (256 units) + output layer (sigmoid)

**Pretrained ResNet50**:

* ResNet50 base from tf.keras.applications, pretrained on ImageNet
* Frozen initially, with a GlobalAveragePooling layer + Dropout + Dense(1, sigmoid)
* Later fine-tuned after unfreezing all layers

**Training/Tuning Process**

* Input shape was resized to 224x224
* Custom CNN trained from scratch using Adam optimizer (1e-4)
* ResNet model trained in two phases: (1) freeze base, train head; (2) unfreeze and fine-tune with lower LR (1e-5)
* EarlyStopping used to avoid overfitting

**Challenges & Solutions**

* Memory and loading time were a concern; we solved this by saving image paths + labels using Pickle, and rebuilding datasets with tf.data.Dataset.
* Transfer learning model required RGB input, so we reprocessed MURA to 3 channels.

**Experimental Results**

(To be updated after training completes — upload images and insert metrics)

* **Custom CNN Accuracy**: ~87% (val) 📎 Upload Image: mura\_custom\_cnn\_accuracy\_plot.png 📎 Upload Image: mura\_custom\_cnn\_loss\_plot.png 📎 Upload Image: mura\_custom\_cnn\_confusion\_matrix.png
* **ResNet Transfer Accuracy**: TBD 📎 Upload Image: mura\_resnet\_accuracy\_plot.png 📎 Upload Image: mura\_resnet\_confusion\_matrix.png

**Submission Package**

* ✅ Clean .ipynb notebooks for all parts
* ✅ Report (PDF and Markdown/Notebook)
* ✅ Screenshots: model training logs, plots, confusion matrices
* ✅ GitHub link: [insert your GitHub repo or Dropbox link here]

*This report documents the experiments and results obtained from applying deep learning techniques to classification problems involving both common image datasets and medical radiographs.*