# Spring 25 |DATA 255 |Homework # 2
# Deadline – 11.59 PM – 03/18/2025
# 20 Points

Minimum number of epochs should be 100 and include early stopping criteria as callbacks - for all questions.

**Problem 1 (3 pts):** Apply the following models on the **FractureMNIST3D** Dataset. Train the model with the training data and validation data. Evaluate the model with the test data.

```
!pip install medmnist
from medmnist import FractureMNIST3D
train_dataset = FractureMNIST3D(split="train", download=True)
val_dataset   = FractureMNIST3D(split="val", download=True)
test_dataset  = FractureMNIST3D(split="test", download=True)
```

"The FractureMNIST3D is based on the RibFrac Dataset, containing around 5,000 rib fractures from 660 computed tomography 153 (CT) scans. The dataset organizes detected rib fractures into 4 clinical categories (i.e., buckle, nondisplaced, displaced, and segmental rib fractures). As we use low-resolution images, we disregard segmental rib fractures and classify 3 types of rib fractures (i.e., buckle, nondisplaced, and displaced). For each annotated fracture area, we calculate its center and resize the center-cropped 64mm×64mm×64mm image into 28×28×28. The official split of training, validation and test set is used."

You can check the benchmark performance on the dataset here - https://www.nature.com/articles/s41597-022-01721-8/tables/5

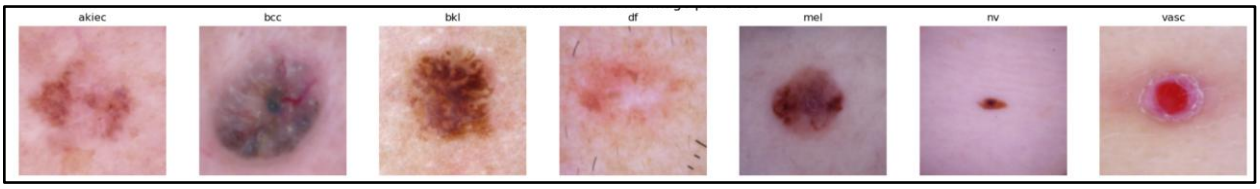
**CNN model from scratch:** Develop a CNN model with 3 convolutional layers (with kernel size= 3, stride =1, padding = "same", activation function = "relu") with following MaxPooling layer (Size= 2) and 3 fully connected layer (including one output layer). After each of the Convolutional layer apply Batch Normalization. In the fully connected layer apply dropout (rate 0.50). Show the learning curve. Report performance evaluation on the test data.

**BONUS:** You can experiment with any CNN architecture and apply it to this dataset. If you achieve over 52% accuracy, you'll earn a bonus of **3 points.**

**Problem 2 (10 pts):** Apply the following models on the **DermaMNIST** Dataset. Train the model with the training and validation datasets. Evaluate the model with the test data.

```
from medmnist import DermaMNIST
train_dataset = DermaMNIST(split="train", download=True)
val_dataset   = DermaMNIST(split="val", download=True)
test_dataset  = DermaMNIST(split="test", download=True)
train_dataset.info
```

The DermaMNIST is based on the HAM10000, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. The dataset consists of 10,015 dermatoscopic images categorized as 7 different diseases, formulized as a multi-class classification task. We split the images into training, validation and test set with a ratio of 7:1:2. The source images of 3×600×450 are resized into 3×28×28.




**a.** Design and implement a deep CNN with at least three convolutional layers, integrating batch normalization and dropout. Train your network, display learning curves, and evaluate its performance on test data. Next, apply at least three distinct image augmentation techniques to the training set, retrain your model, and compare the outcomes. Analyze how the augmentation strategies affect the learning process and overall accuracy. **3 pts**

**b.** **Transfer Learning:** Load the VGG-19 model. Drop after final pooling layer and on top of it add one **global average pooling** layer and one fully connected layer, and one final output layer. Report performance evaluation on the test data. **2 pts**

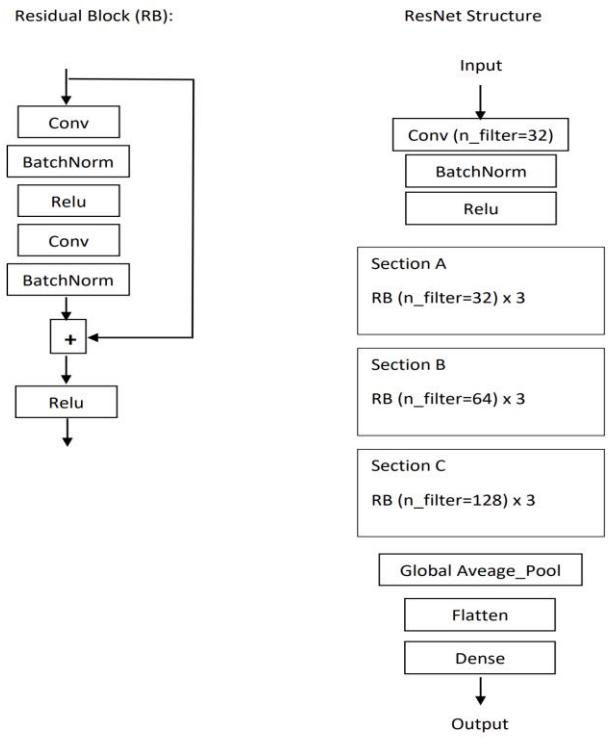
**c.** Use **Grad-CAM (Gradient-weighted Class Activation Mapping)** to create visual explanations for CNN's decisions on the DermatMNIST dataset. Use the pre-trained CNN from question 2a. Generate heatmaps for 5 images from each class to highlight the key regions influencing the model's classifications and explain. **2pts**

d. You can use the pre-trained CNN model in part (a), apply the following three quantization techniques: post-training dynamic range quantization, full integer quantization, and quantization-aware training. Evaluate and compare each quantized model to the original in terms of model size and test accuracy. Discuss the trade-offs and implications of using each quantization method. **3pts**

**BONUS:** You can experiment with any CNN architecture and apply it to this dataset. If you achieve over 77% accuracy, you'll earn a bonus of **3 points.**

**Problem 3 (7 pts):** Developing ResNet model from scratch using CIFAR10 dataset.

Apply a residual network specified in the following architecture. All convolutional layers use kernel size 3, stride = 1, and padding = “same”, minimum number of epochs 20.




Useful link-

Data Download:

DermaMinist: https://zenodo.org/records/10519652

CIFAR10: Keras: https://keras.io/api/datasets/cifar10/

Pytorch: https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR10.html

Model Quantization: https://www.tensorflow.org/model_optimization/

Layer concatenation: https://keras.io/api/layers/merging_layers/concatenate/

Model save in keras: https://www.tensorflow.org/guide/keras/save_and_serialize

You are required to submit:

1. An MS/PDF/Scanned document:
    a. Include all the steps of your calculations.
    b. Attach screenshots of the code output.
    c. Include the summary of the model
    d. Include a Table - Mention all the hyperparameters you selected: activation function in hidden layer and output layer, weight initializer, number of hidden layers, neurons in

hidden layers, loss function, optimizer, number of epochs, batch size, learning rate, evaluation metric

2. Source code:
    a. Python (Jupyter Notebook)
    b. Ensure it is well-organized with comments and proper indentation.

- Failure to submit the source code will result in a deduction of 5 points.
- Format your filenames as follows: "your_last_name_HW2.pdf" for the document and "your_last_name_HW2_source_code.ipynb" for the source code.
- Before submitting the source code, please double-check that it runs without any errors.
- Must submit the files separately.
- Do not compress into a zip file.
- HW submitted more than 24 hours late will not be accepted for credit.