Exercise
Image Classification

In this exercise, you will design your **deep model** for classification purposes. You will use **Fashion-Mnist (https://github.com/zalandoresearch/fashion-mnist)**, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.

Dataset:
Keras already include the dataset. You can read it by the following command:

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.fashion_mnist.load_data()

Don't forget to preprocess the data (at least convert to float32!)

Design the model:

You will have to design and compare at least two models (only Conv2D and dense layers): one purely convolutional and the other including residual blocks or, if you like, inception blocks.

You are free to design the models as you like the most.
Your models can include any combination of layers we have studied during the course including simple Conv2D, Conv2D 1x1, Inception module, residual blocks, pooling layers. If you feel unsure, you can always start from the popular architectures we have studied as a starting point.

For sure, **the last layer of your model will be fully connected with 10 neurons (one per class).**
Feel free to include any non-linear activation function, despite *it is advisable to use ReLU*.
Choose an *appropriate loss function* and *optimizer*, and experiment with several changes to the model and, if necessary, to the learning rate until you are not satisfied.
Use part of the training set as validation data (*do never use the test set to perform model selection!!!*).
If needed, you must properly regularize your network, for instance by dropout layers. Check if the model is overfitting by comparing the training and the validation loss. If you like, also include an early stopping layer.

When training your model, use 20% of the training set as a validation set.

Data generator:
Despite not necessary with this dataset, you may want to try implementing a data generator to augment your dataset by mirroring the images or adding some rotation.

Performance:
After the training is complete, you will evaluate the model on your test set.
You should prepare a table comparing the accuracy values in test for your developed models. This can be done by including in the **compile** method the attribute **metrics=[tf.keras.metrics.CategoricalAccuracy()].**
The table should also report the total number of parameters in your models, the final training loss, the final validation loss, and the final test loss.