UPM - DEEP LEARNING COURSE

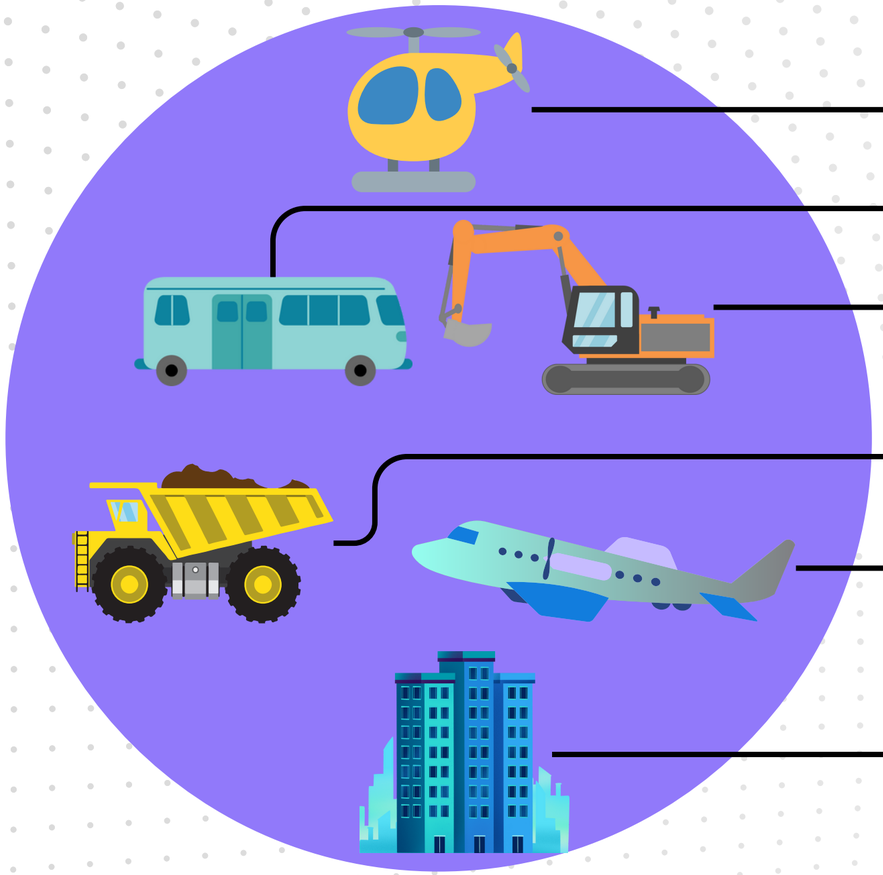
The CNN challenge 2024

Assignment 2

Federico Paschetta - Cecilia Peccolo

The Problem

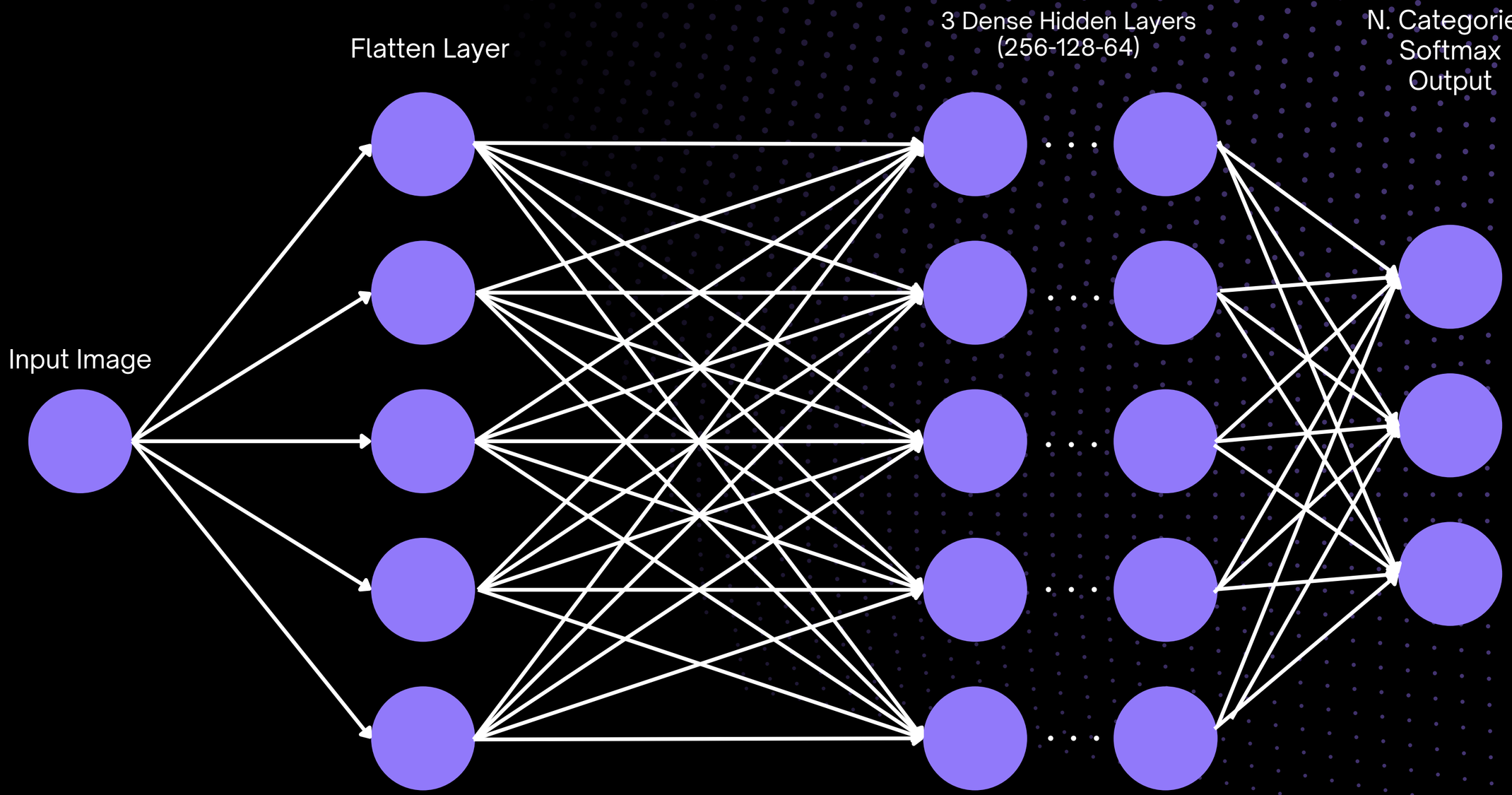
Images Dataset



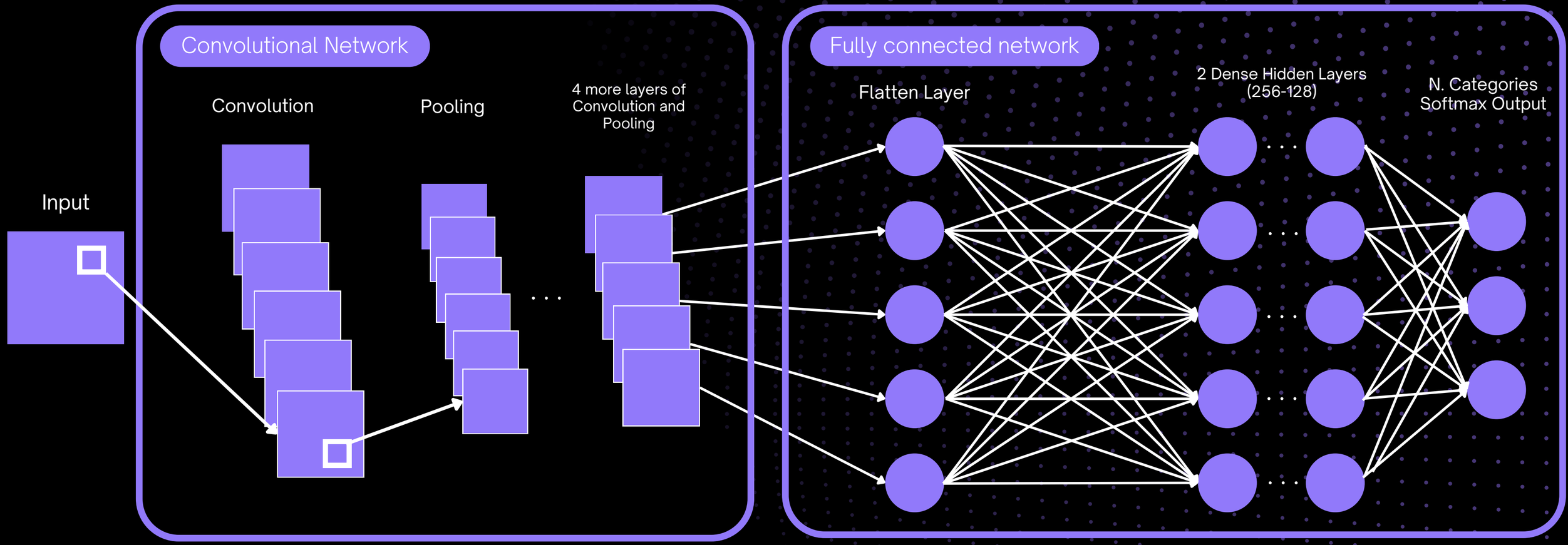
Correct Label

- Building
- Bus
- Cargo Plane
- Dump Truck
- Excavator
- Helicopter

The starting architecture



The new architecture



Convolutional Network Performance Boost

In order to catch the complexity of the images, we implemented a Convolutional Neural Network with alternating a convolutional layer and a pooling layer in the first part of the net, followed by the ffNN.

- **Conv2D Layers**
  4 layers with increasing number of filters (32-64-128-256-512), strides equal to 1, padding is set to “VALID”, so zero padding is not applied

- **Conv2D Configuration**
  RELU activation function has been used in every layer except the output one

- **Pooling Layers Configuration**
  Max pooling layers have been used, with 2x2 pooling kernel, no strides and “valid” padding. In order to avoid overfitting, as Dropout rate of 30% has been applied.

Neural Network Performance Boost

After Convolutional Network layers we place a fully connected network with all previous performance boosting characteristics

Deep Network

2 hidden layers with 256 and 128 neurons each

Leaky ReLU

Leaky ReLU gives best results possible

Dropout

0.3 Dropout rate makes the network overfits less

Normalization

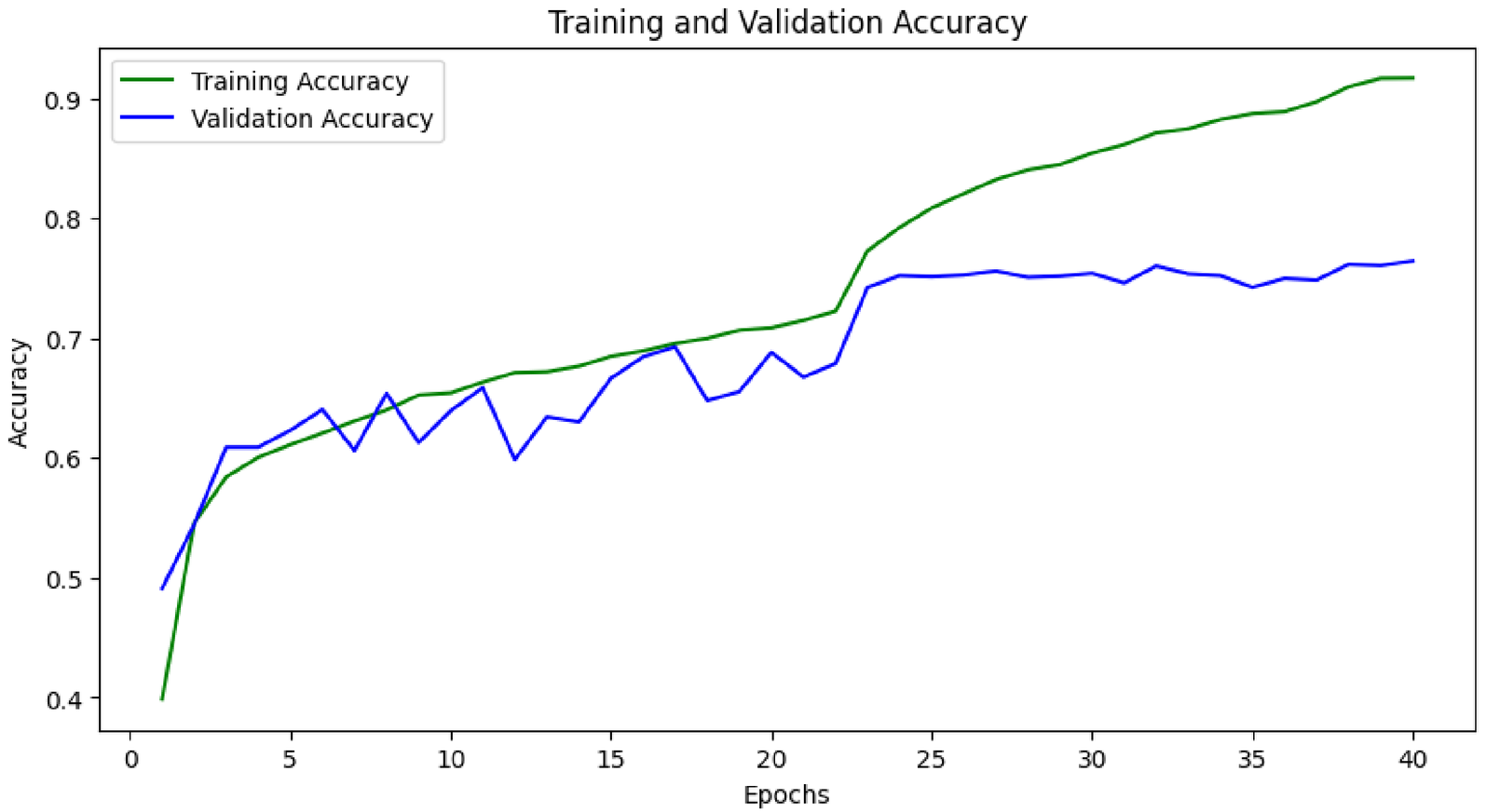
Batch Normalization copes with vanishing gradient problem

Adam

Adam combines all pros from best optimizers

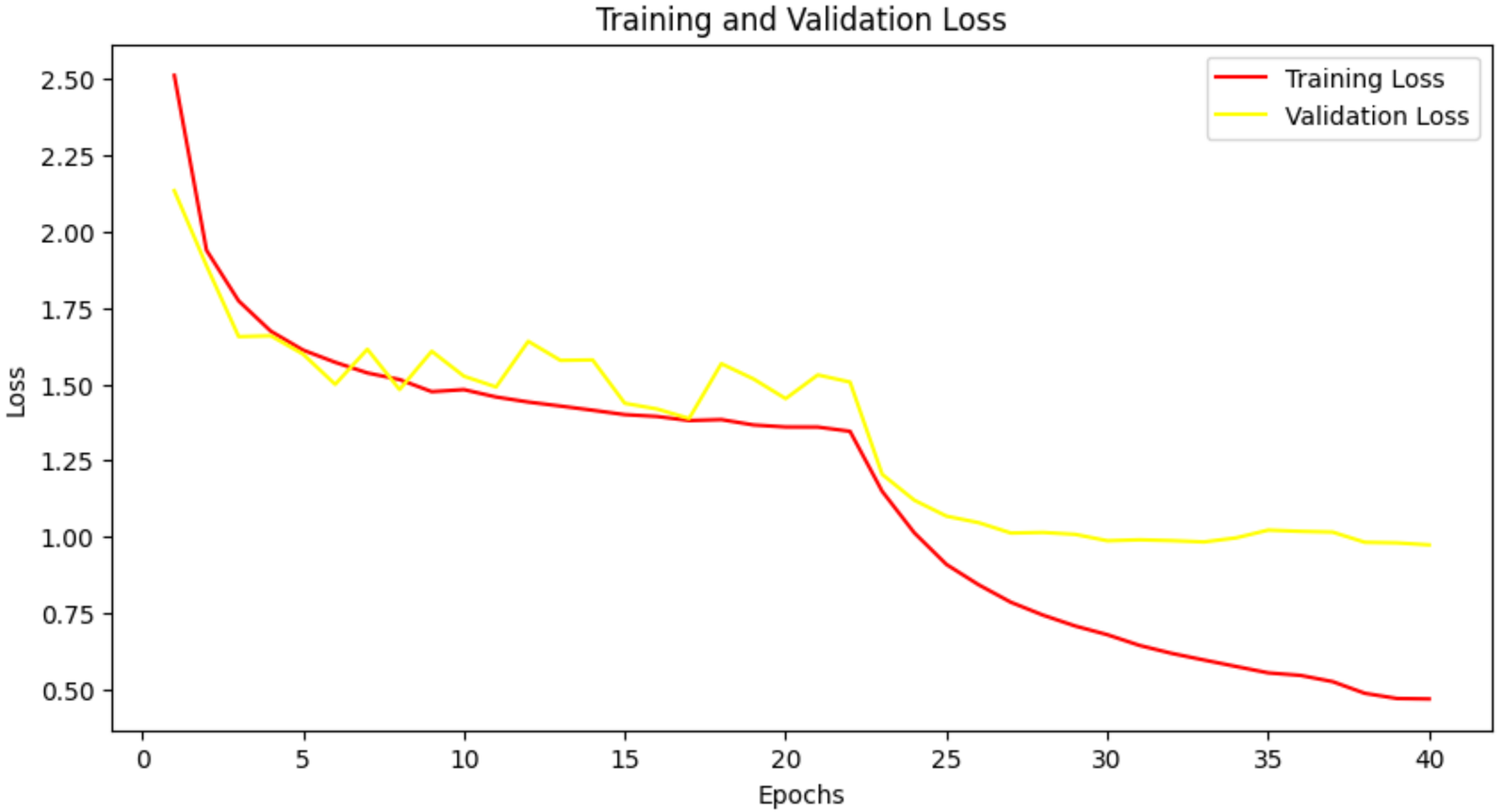
Accuracy vs Epochs

As we can imagine, the curve of the accuracy in the training set keeps increasing while the validation accuracy doesn't improve after 23 epochs.
At 23 epochs we have the best models that optimizes the accuracy of both training and validation and avoids overfitting.



Loss vs Epochs

As we can notice, also concerning the loss we have the best results that compromise performance and overfitting for 23-25 epochs. In fact, this is the model selected to reach the following results on the test set.



Deep Neural Network

The initial FF Neural Network implemented had the following hyperparameters:

- **3 hidden layers** with respectively 256-128 and 64 neurons
- The **RELU** activation function was applied to each layer
- Batch nornalization
- **Adam** optimization
- Dropout rate of **20%**

RESULTS in terms of maximum ACCURACY

validation set: 57.8%
test test: 49.5%

Convolutional Neural Network

The new Neural Network applied differentiates mainly for the following changes:

- we **added 5 convolutional layers**
- changed the activation function of the ffNN layers to **Leaky ReLU** except the output one
- theDropout rate has been increased to **30%**
- we applied the He normalization technique to the weights of the ffNN

RESULTS in terms of maximum ACCURACY

validation set: 76.4%
test test: 67.1%

Confusion matrix

ACCURACY: 67.1%

PRECISION: 59.2%

RECALL: 59.7%

CONCLUSIONS

AND POSSIBLE FUTURE DEVELOPMENT

The optimization of hyperparameters can be taken to a deeper level: in fact, a potential developments of the project to achieve better results would involve employing a grid search methodology, combining different numbers of hidden layers, activation functions, number of neurons per layer, additional dropout rates, or other normalizations, and exploring all the combinations of these parameters to determine the model that yields the best predictions. The choice of the hyperparameters selected in this presentation has been primarily based on intuition, given the knowledge acquired in Unit 1 and Unit 2 of the Deep Learning course.

However, we can consider our results satisfying since we improved the perfomance of both the Neural Network and professors' Convolutional Neural Network.