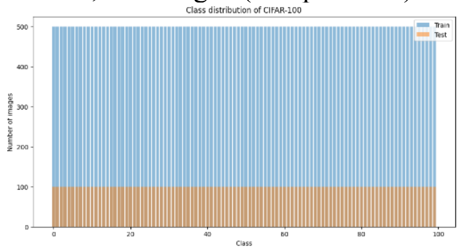
CIFAR-100 classification with CNN and MLP

Introduction

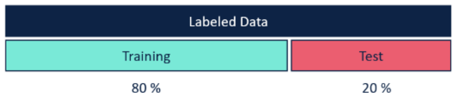
The CIFAR-100 dataset is a collection of 60,000 32x32 color images divided into 100 classes, with each class containing 600 images.
In this case we are aiming at predicting the correct class out of the 100 fine-grained categories

The dataset is divided into a training set which contains 50,000 images (500 per class) and a test set with 10,000 images (100 per class).



Validation set

For these exercises, we will split the training set using HOLDOUT 80/20, which means 80% of the original training set will be used for actual training and the 20% remaining for hyperparameter validation.



Hyperparameter tuning

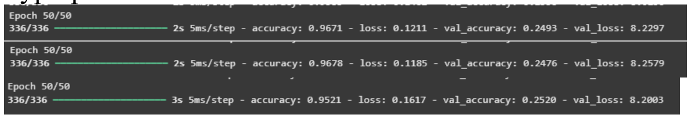
In order to get the best (or at least a fine) set of hyperparameters we will perform the *Bayesian search algorithm*. [2]

Training an MLP model

Best hyperparameters founded comparing 15 models trained with 5 epochs each (With Bayesian search):

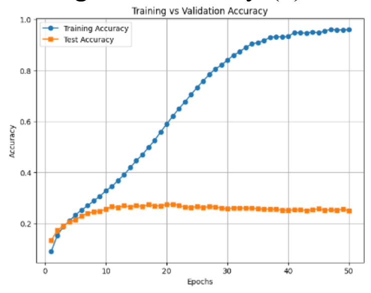
Hidden layers	3
Number of units (neurons)	805
Activation function	RELU
Learning rate	0.0003 approx.
Batch size	149

Three different trained models from the selected hyperparameter set



```
Epoch 50/50
336/336 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.9671 - loss: 0.1211 - val_accuracy: 0.2493 - val_loss: 8.2297
Epoch 50/50
336/336 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.9678 - loss: 0.1185 - val_accuracy: 0.2476 - val_loss: 8.2579
Epoch 50/50
336/336 ━━━━━━━━━━━━━━━━━━━━ 3s 5ms/step - accuracy: 0.9521 - loss: 0.1617 - val_accuracy: 0.2520 - val_loss: 8.2003
```

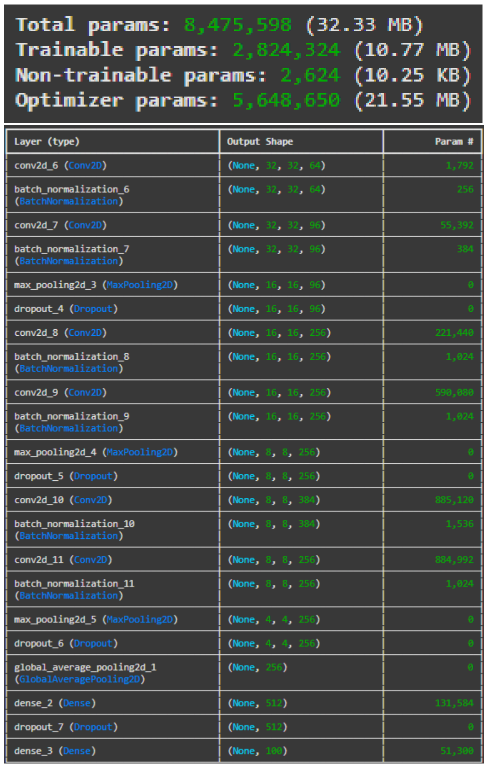
Average test accuracy (3): 24.96%



Our results demonstrate that MLP struggles with the CIFAR-100 dataset due to its inability to capture spatial hierarchies, leading to overfitting and poor generalization. Since MLPs treat images as flat vectors, they fail to extract meaningful patterns necessary for complex image classification tasks. To address these limitations, we will explore Convolutional Neural Networks (CNNs), which leverage spatial locality and hierarchical feature learning to improve accuracy and generalization on CIFAR-100.

Training a CNN model

After performing several approaches with different CNN topologies, the final CNN topology is presented in the following table



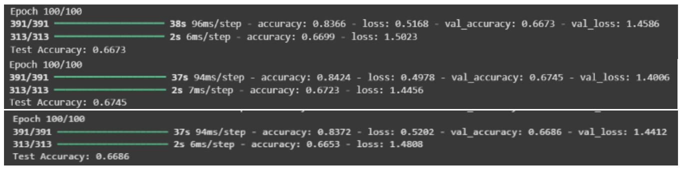
```
Total params: 8,475,598 (32.33 MB)
Trainable params: 2,824,324 (10.77 MB)
Non-trainable params: 2,624 (10.25 KB)
Optimizer params: 5,648,650 (21.55 MB)
```

```
Layer (type)                         Output Shape              Param #
conv2d_6 (Conv2D)                    (None, 32, 32, 64)          1,792
batch_normalization_6                (None, 32, 32, 64)            256
(BatchNormalization)
conv2d_7 (Conv2D)                    (None, 32, 32, 96)         55,392
batch_normalization_7                (None, 32, 32, 96)            384
(BatchNormalization)
max_pooling2d_3 (MaxPooling2D)       (None, 16, 16, 96)              0
dropout_4 (Dropout)                  (None, 16, 16, 96)              0
conv2d_8 (Conv2D)                    (None, 16, 16, 256)       221,440
batch_normalization_8                (None, 16, 16, 256)         1,024
(BatchNormalization)
conv2d_9 (Conv2D)                    (None, 16, 16, 256)       590,080
batch_normalization_9                (None, 16, 16, 256)         1,024
(BatchNormalization)
max_pooling2d_4 (MaxPooling2D)       (None, 8, 8, 256)               0
dropout_5 (Dropout)                  (None, 8, 8, 256)               0
conv2d_10 (Conv2D)                   (None, 8, 8, 384)         885,120
batch_normalization_10               (None, 8, 8, 384)           1,536
(BatchNormalization)
conv2d_11 (Conv2D)                   (None, 8, 8, 256)         884,992
batch_normalization_11               (None, 8, 8, 256)           1,024
(BatchNormalization)
max_pooling2d_5 (MaxPooling2D)       (None, 4, 4, 256)               0
dropout_6 (Dropout)                  (None, 4, 4, 256)               0
global_average_pooling2d_1           (None, 256)                     0
(GlobalAveragePooling2D)
dense_2 (Dense)                      (None, 512)               131,584
dropout_7 (Dropout)                  (None, 512)                     0
dense_3 (Dense)                      (None, 100)                51,300
```

Best hyperparameters founded comparing 10 models trained with 20 epochs each:

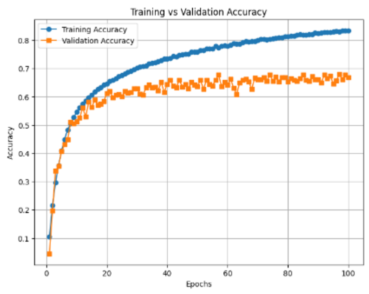
Convolution 1 Filters	64
Convolution 2 Filters	96
Convolution 3 Filters	256
Convolution 4 Filters	256
Convolution 5 Filters	384
Convolution 6 Filters	256
Dense Units (neurons)	512
Dropout 1	0.2
Dropout 2	0.4
Dropout 3	0.3
Dropout Dense	0.5
Learning Rate	0.0012 approx.

Three different trained models from the selected hyperparameter set:



```
Epoch 100/100
391/391 ━━━━━━━━━━━━━━━━━━━━ 38s 96ms/step - accuracy: 0.8366 - loss: 0.5168 - val_accuracy: 0.6673 - val_loss: 1.4586
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 6ms/step - accuracy: 0.6699 - loss: 1.5023
Test Accuracy: 0.6673
Epoch 100/100
391/391 ━━━━━━━━━━━━━━━━━━━━ 37s 94ms/step - accuracy: 0.8424 - loss: 0.4978 - val_accuracy: 0.6745 - val_loss: 1.4006
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step - accuracy: 0.6723 - loss: 1.4456
Test Accuracy: 0.6745
Epoch 100/100
391/391 ━━━━━━━━━━━━━━━━━━━━ 37s 94ms/step - accuracy: 0.8372 - loss: 0.5202 - val_accuracy: 0.6686 - val_loss: 1.4412
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 6ms/step - accuracy: 0.6653 - loss: 1.4808
Test Accuracy: 0.6686
```

Average test accuracy (3): 0.6701 %



Convolutional Neural Networks (CNNs) significantly outperform MLPs on CIFAR-100 because they efficiently capture spatial hierarchies in images. Unlike MLPs, which flatten images and lose spatial relationships, CNNs use convolutional layers to detect patterns like edges, textures, and shapes, making them ideal for image classification. Additionally, pooling layers help reduce dimensionality while preserving important features, improving generalization. These properties allow CNNs to extract more meaningful representations from complex datasets like CIFAR-100, leading to a much higher accuracy of 67% compared to the MLP approach.

Room for improvement

Although CNNs significantly outperform MLPs on CIFAR-100, achieving 67% accuracy still leaves room for improvement. The dataset's complexity, due to high intra-class variability, low inter-class differences, and limited training data, makes classification challenging. To enhance performance, more advanced architectures such as PyramidNet, ResNet with ELU activation, and Transformer-based models like Astroformer and ViT-B-16 have demonstrated superior results. These models leverage improved feature extraction, better gradient flow, and adaptive learning capabilities.

Performance of different approaches in comparison with MLP and CNN experiments

Model	**Accuracy (%)**
ViT-B-16	**94.2**
Astroformer	**93.36**
PiramidNet	**89.7**

ResNet + ELU	**73.5**
CNN	**67.01**
MLP	**24.96**

[3]

Appendix

[1] Data exploration, MLP training and CNN training code:
https://colab.research.google.com/drive/1hKk4wSGyJNe83sGgFOl2JmKgKFR4OWvF?usp=sharing

[2] Bayesian search algorithm description:
https://keras.io/keras_tuner/api/tuners/bayesian/

[3] Top models predicting CIFAR-100

ViT-B-16:
https://paperswithcode.com/paper/imagenet-21k-pretraining-for-the-masses

Astroformer:
https://paperswithcode.com/paper/astroformer-more-data-might-not-be-all-you

PiramidNet:
https://paperswithcode.com/paper/sharpness-aware-minimization-for-efficiently-1

ResNet+ELU:
https://paperswithcode.com/paper/deep-residual-networks-with-exponential