Machine Learning a.y. 22-23

Homework 2: Report

Davide Gabrielli - 1883616
gabrielli.1883616@studenti.uniroma1.it

January 7, 2023

1 Introduction

For this homework we have been asked to solve a image classification problem choosing a dataset having at least 10 classes and 150 images for each class. In this report I will describe the dataset, the preprocessing steps, the models used and the results obtained.

1.1 Dataset

The dataset I have chosen is the 10 Monkey Species dataset from Kaggle. It is composed of 10 different classes of monkeys, each class has more than 150 images as shown in Figure 1. There are almost 1400 images with resolution of 400×300 px or larger and the images are in RGB format, some examples are shown in Figure 2.

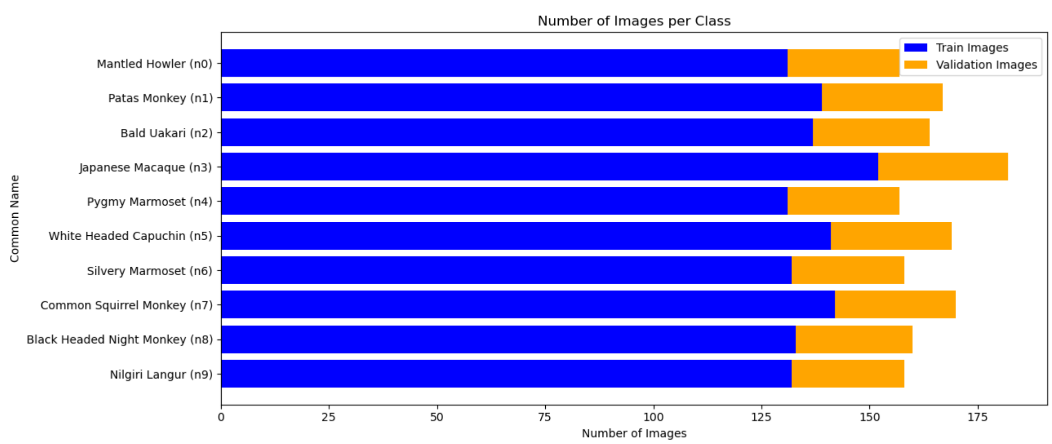


Figure 1: Dataset distribution by class

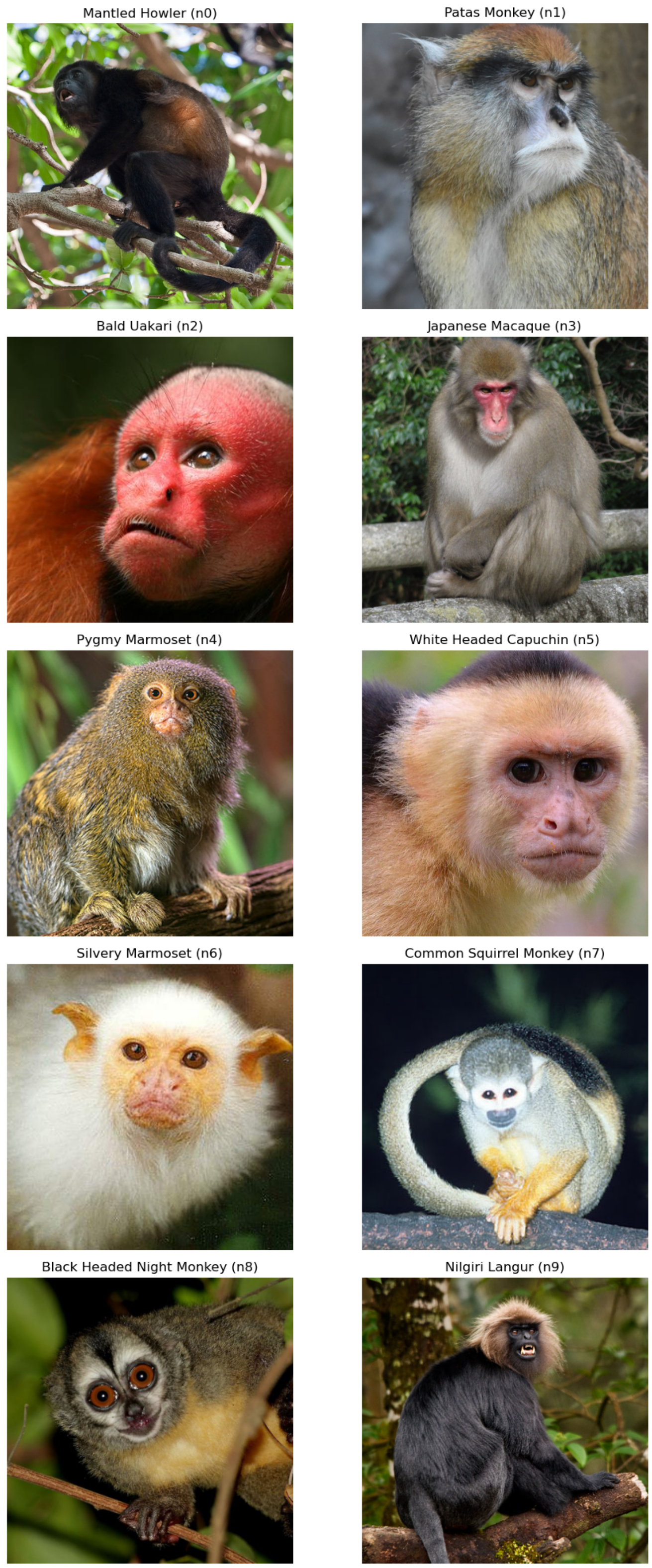


Figure 2: Dataset examples

2 Preprocessing

In this section I will describe the preprocessing steps I have applied to the dataset.

2.1 Dataset loading

Since the dataset already have a train and a validation set, I loaded them separately and used them to train and evaluate the models. The sets have been shuffled and the images have been resized to 256×256 px.

2.2 Normalization

Since the image pixels are in the range [0, 255] I have normalized them to the range [0, 1] dividing each pixel by 255. This step is important because it allows the network to converge faster and it is also important to avoid numerical issues.

2.3 Data augmentation

The second step I have applied to the dataset is the data augmentation. I have used the ImageDataGenerator class from TensorFlow which allows to apply different transformations to the images in real time. So the following transformations have been applied to the images:

- **Rotation**: the images have been rotated by a random angle between -20 and 20 degrees.
- **Width shift**: the images have been shifted horizontally by a random amount between -0.2 and 0.2 of the width.
- **Height shift**: the images have been shifted vertically by a random amount between -0.2 and 0.2 of the height.
- **Shear**: the images have been sheared by a random angle between -20 and 20 degrees.
- **Zoom**: the images have been zoomed by a random amount between 0 and 0.2.
- **Horizontal flip**: the images have been flipped horizontally with a probability of 0.5.

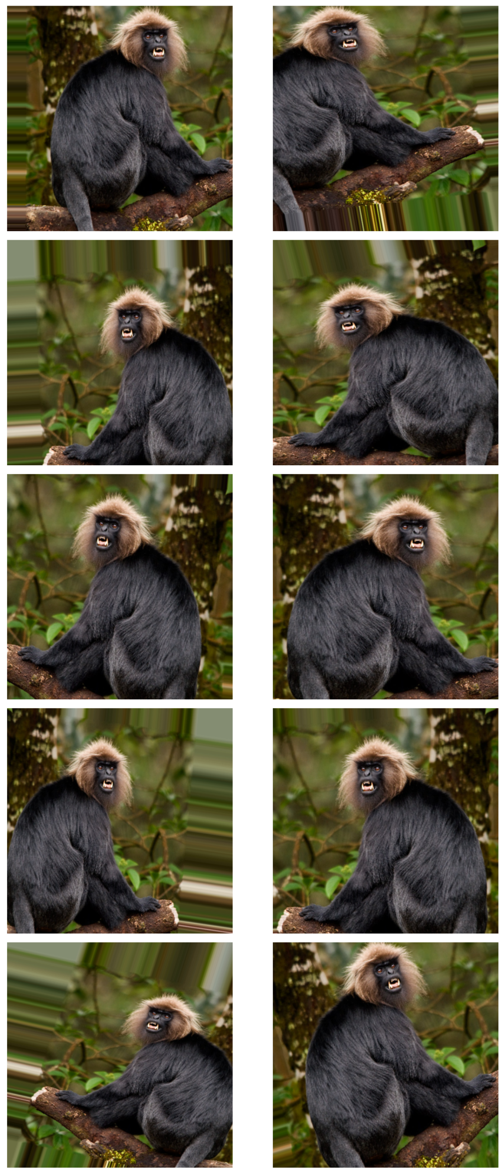


Figure 3: Data augmentation examples

3 AlexNet

In this section I will describe the AlexNet architecture and the results obtained using it.

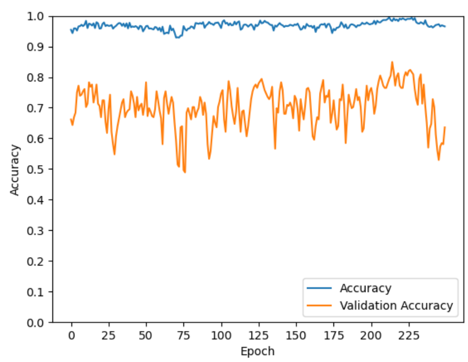
3.1 Architecture

The AlexNet architecture is a convolutional neural network designed by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton in 2012. It was the first architecture that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. The architecture is structured as follows:

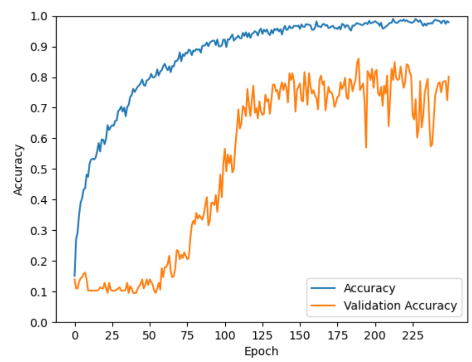
- **Input**: $256 \times 256 \times 3$
- **Convolution**: 96 filters of size $11 \times 11 \times 3$ with stride 4 and padding valid
- **Max Pooling**: 3×3 with stride 2
- **Convolution**: 256 filters of size $5 \times 5 \times 96$ with stride 1 and padding same
- **Max Pooling**: 3×3 with stride 2
- **Convolution**: 384 filters of size $3 \times 3 \times 256$ with stride 1 and padding same
- **Convolution**: 384 filters of size $3 \times 3 \times 384$ with stride 1 and padding same
- **Convolution**: 256 filters of size $3 \times 3 \times 384$ with stride 1 and padding same
- **Max Pooling**: 3×3 with stride 2
- **Flatten**
- **Dense**: 4096 neurons
- **Dropout**: rate 0.5
- **Dense**: 4096 neurons
- **Dropout**: rate 0.5
- **Dense**: 10 neurons

3.2 Training

I have trained two AlexNet models with different learning rates (0.001 and 0.0001) using the Adam optimizer. It has been trained using Google Colab with a GPU runtime and the training took about 1 hour for each model. The training and validation accuracy and loss are shown in Figure 4 and Figure 5. As we can see, a learning rate of 0.001 is too high and the model does not converge.

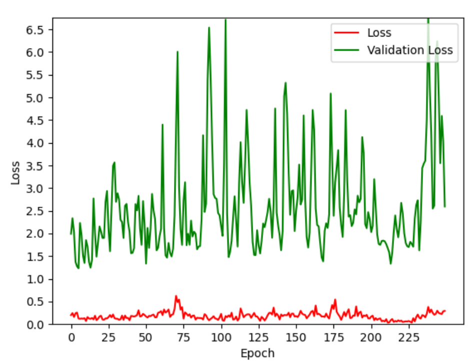


(a) Learning Rate = 0.001

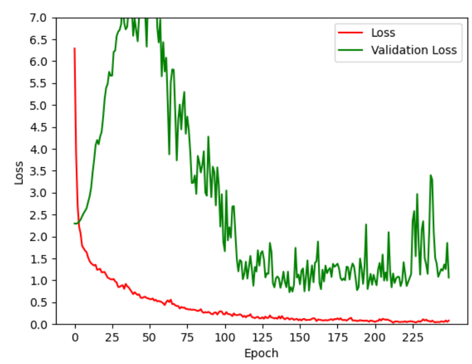


(b) Learning Rate = 0.0001

Figure 4: Training and validation accuracy for the AlexNet with different learning rates



(a) Learning Rate = 0.001



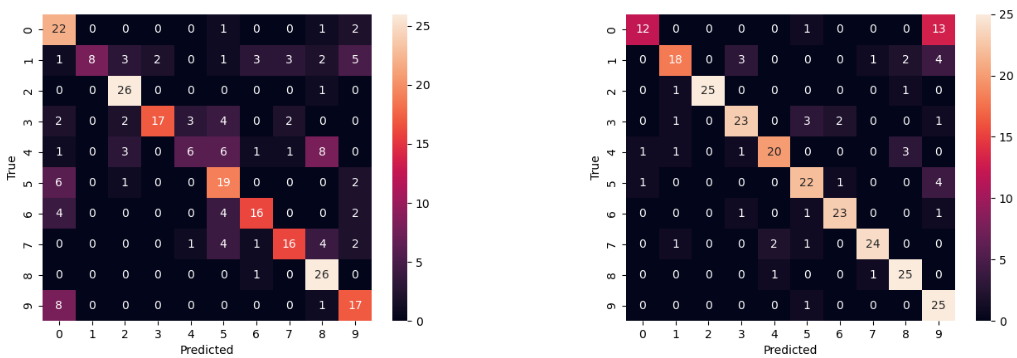
(b) Learning Rate = 0.0001

Figure 5: Training and validation loss for the AlexNet with different learning rates

3.3 Evaluation

The model was evaluated using the test dataset and the results are shown in the confusion matrix in Figure 6. The accuracy, precision, recall, F1-score and AUC are shown in Table 1.

As we can see from the results, the model model with a learning rate of 0.0001 performs better than the one with a learning rate of 0.001. However, the model with a learning rate of 0.0001 still has a low accuracy and the AUC is not very high. Looking at the confusion matrix, we can see that the model is not able to distinguish between the classes 0 and 9 which are the most similar.



(a) Learning Rate = 0.001

(b) Learning Rate = 0.0001

Figure 6: Confusion matrix for the AlexNet with different learning rates

LR	Accuracy	Precision	Recall	F1-Score	AUC
0.001	0.6985	0.9748	0.9431	0.9587	0.8562
0.0001	0.7977	0.9457	0.9919	0.9683	0.7267

Table 1: Evaluation of the AlexNet with different learning rates

4 Convolutional Neural Network

In this section I will describe a Convolutional Neural Network architecture designed by me.

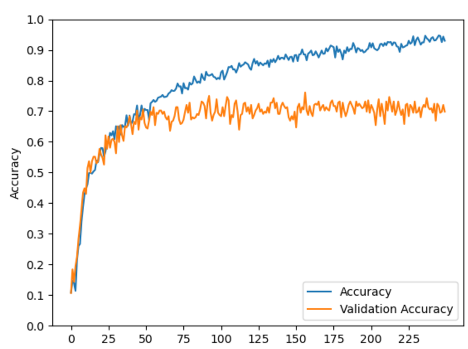
4.1 Architecture

The architecture is structured as follows:

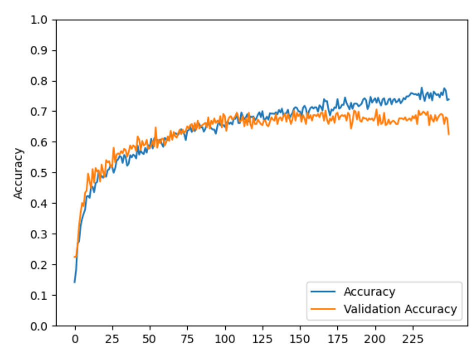
- **Input**: 256×x256×3
- **Convolution**: 32 filters of size 11×11×3 with stride 4 and padding valid
- **Max Pooling**: 2×2 with no stride
- **Convolution**: 64 filters of size 5×5×96 with stride 1 and padding valid
- **Max Pooling**: 2×2 with no stride
- **Flatten**
- **Dense**: 128 neurons
- **Dense**: 10 neurons

4.2 Training

I have trained two CNN models with different learning rates (0.001 and 0.0001) using the Adam optimizer. It has been trained using Google Colab with a GPU runtime and the training took about 1 hour for each model. The training and validation accuracy and loss are shown in Figure 7 and Figure 8.

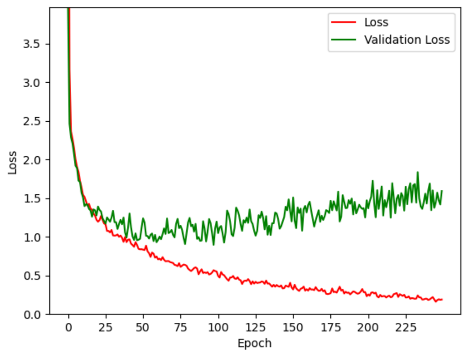


(a) Learning Rate = 0.001

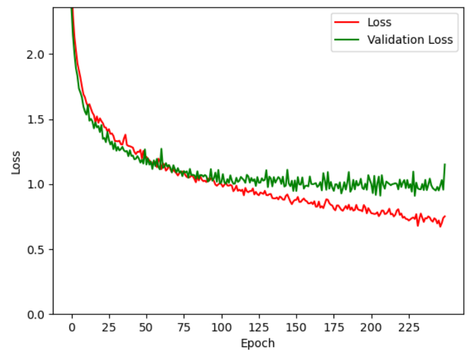


(b) Learning Rate = 0.0001

Figure 7: Training and validation accuracy for the CNN with different learning rates



(a) Learning Rate = 0.001



(b) Learning Rate = 0.0001

Figure 8: Training and validation loss for the CNN with different learning rates

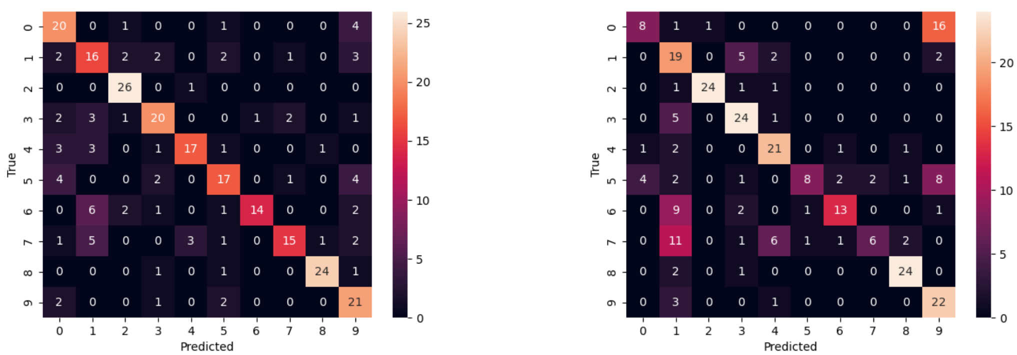
4.3 Evaluation

The model was evaluated using the test dataset and the results are shown in the confusion matrix in Figure 9. The accuracy, precision, recall, F1-score and AUC are shown in Table 2.

As we can see from the results, both the learning rates have similar results. However, the model with a learning rate of 0.001 have a better accuracy and a better AUC. In fact we can see from the confusion matrix that the model with a learning rate of 0.001 is able to classify better the classes.

LR	Accuracy	Precision	Recall	F1-Score	AUC
0.001	0.6360	0.9825	0.9106	0.9451	0.8784
0.0001	0.6213	0.9305	0.9797	0.9545	0.6437

Table 2: Evaluation of the CNN with different learning rates



(a) Learning Rate = 0.001

(b) Learning Rate = 0.0001

Figure 9: Confusion matrix for the CNN with different learning rates

5 Conclusions

In this section I will compare the results of the AlexNet with learning rates of 0.0001 and the CNN with learning rates of 0.001 since they have the best results. As we can see in Figure 10 and Figure 11 both the models have similar results but the CNN seeems to converge faster than the AlexNet.

However the dataset is not big enough to train a deep neural network and the results are not very good. In fact also training for more epochs the model does not improve the results because the model starts to overfit the training data.

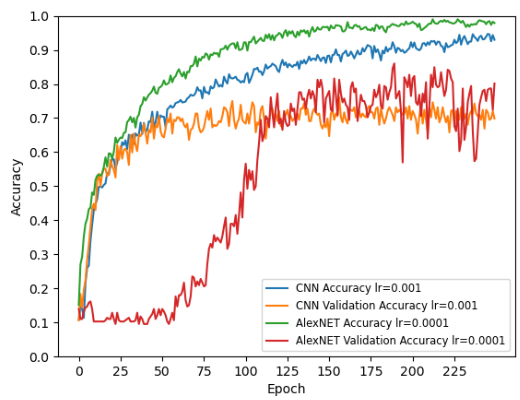


Figure 10: Training and validation accuracy for the AlexNet and the CNN

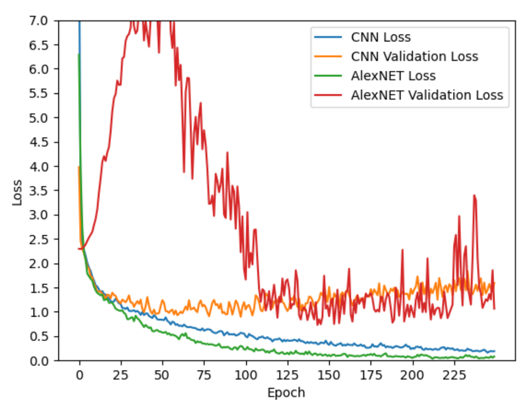


Figure 11: Training and validation loss for the AlexNet and the CNN