Computer Vision

Project 2

Group 11

Frederico Lopes (201904580)
Marta Mariz (201907020)
Miguel Freitas(201906159)

Problem 1

Objective

The objective of this problem was to present a solution for the task of colorizing black and white images. This task involves training a neural network to predict the AB channels of an input grayscale image, given the black and white components. The proposed solution utilizes the CIFAR-10 dataset, which is divided into grayscale images for training and AB channels as labels.

Methodology

To address the colorization task, a model based on the concept of an autoencoder was employed. The model architecture consists of two main parts: the encoder and the decoder. The encoder component utilizes the first 6 sets of layers from the ResNet-18 architecture, adapted for grayscale images. These layers extract meaningful features from the input grayscale image. The decoder component consists of several convolutional and upsampling layers, which generate the AB channels from the encoded features. The output of the model corresponds to the predicted AB channels for the grayscale input. The output can be later combined with the input to create an RGB image that should resemble the original.

The implementation of the model was carried out using the PyTorch framework within a Jupyter notebook. The dataset was split into training and test sets, with the latter used to monitor the model's performance at regular intervals during training. The weights of the model were saved periodically to facilitate model reusability. A batch size of 32 was employed, a normal size for the training of Neural Networks, and the training process utilized the Mean Squared Error (MSE) loss function and Adam optimizer with a learning rate of 0.001, which consistently and steadily decreased the loss during training.

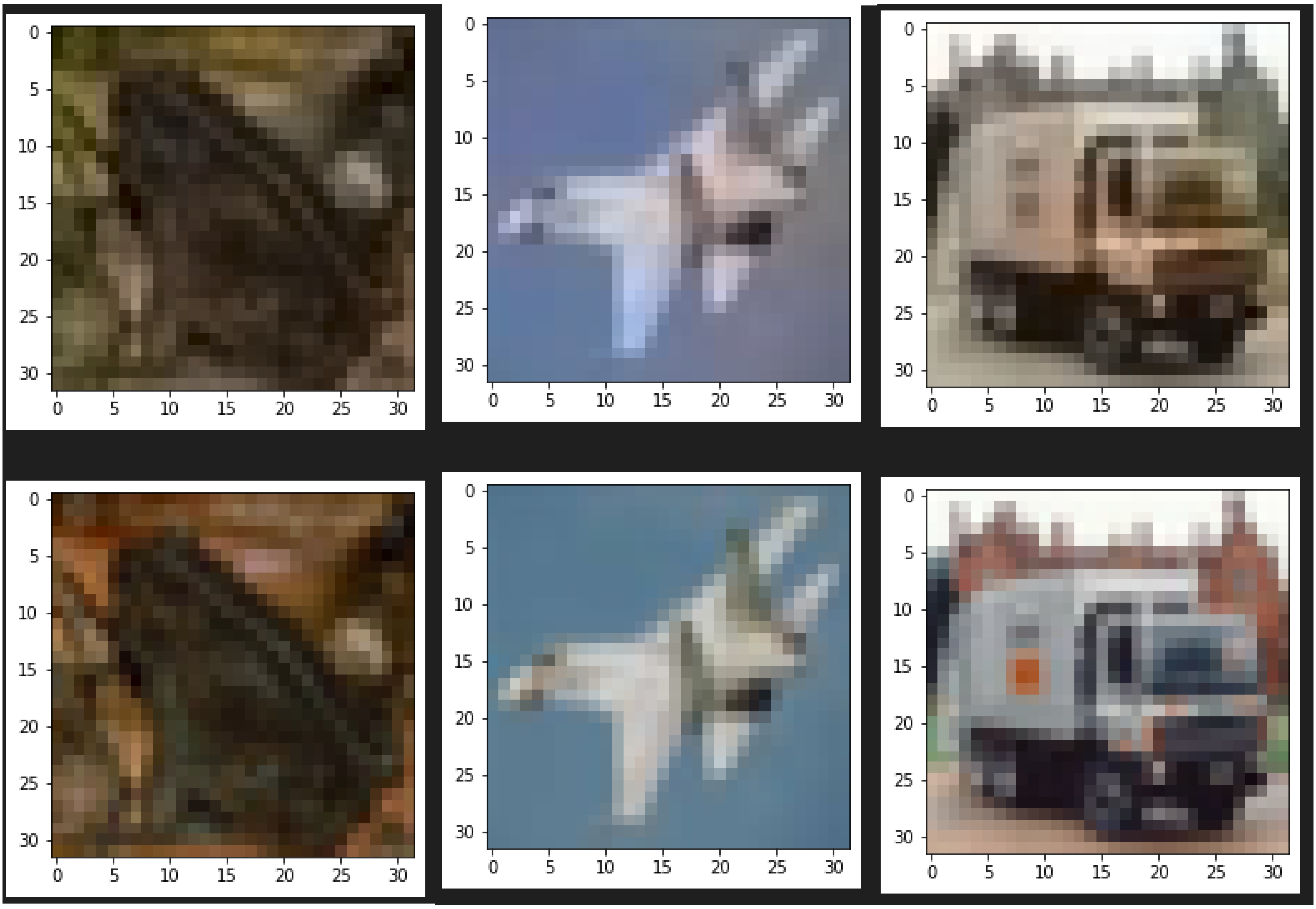
Results and Discussion

The proposed solution achieved promising results in colorizing black and white images. After training for 20 epochs, the model demonstrated significant progress. The training loss reached a minimum value of around 0.0008, indicating that the model successfully learned to generate color information. In the test set, the loss reached 0.002, further confirming the model's capability to generalize well to unseen images.

The colorization results were visually appealing, and, even in most cases most of the color was restored, existing some results where it is hard to guess which one is the real one. The model successfully captured details and preserved the overall structure of the original images. Despite the complexity of the task, the proposed method managed to generate colorizations without major artifacts or distortions.

The colorization task is inherently challenging due to the ambiguity involved in assigning colors to grayscale images. Because of this, some difficulties were encountered and the model relied heavily on the quality and diversity of the training dataset. Insufficient representation of certain color patterns or uncommon objects in the training set might result in less accurate colorization for such cases. Moreover, the training process required a significant amount of computational resources and time due to the complexity of the network architecture.

Possible improvements could be a more extensive training, more training data and using a different, possibly larger, neural network.



(top images are the generated ones)

Problem 2

Objective

The goal for the second problem is to test the effects of using Image Colorization as a pretext task in an Image Classification problem. We will compare the approach of initializing random weights against using the weights obtained after training the Colorization Neural Network in the first task.

Methodology

To address the new challenge we will need to design a new model with two parts: the first will simply be a repeat of the encoder from the first task as required in order to achieve our goals. The second part will be a simple classifier component composed of three layers.

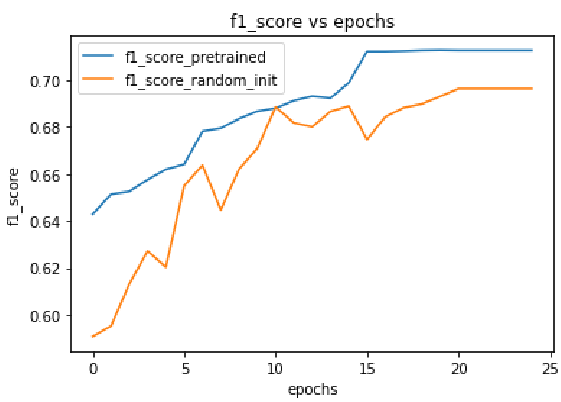
We are classifying grayscale images because the previous encoder was designed with those in mind. In the first training approach we simply initialize all the weights with random values, in the second we use the saved weights from the previous task to initialize the encoder and start with random weights only in the classifier.

Results

The following table shows the performance of both models after training for 25 epochs.

	Accuracy	**F1-Score**	**Precision**
Random parameters	0.6990	0.6964	0.7485
Loaded weights from image colorization	0.7128	0.7036	0.7493

As we can see, we obtain pretty similar results by the end of the 25 epochs, however the difference rests on how quickly both models arrived at these results. As demonstrated by the following graphs, our pre-trained model managed to achieve these results after only 15 epochs, while our randomly initialized model only converged to these values after 20 epochs:



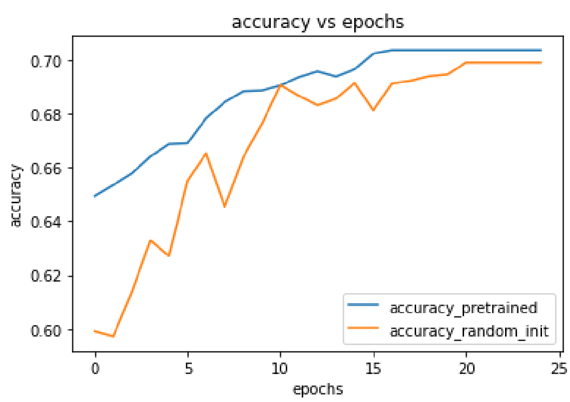


Fig.1 - Evolution of F1-Score for both approaches

Fig.2 - Evolution of accuracy for both approaches

Discussion

As expected the model which uses the weights generated from the image colorization task managed to converge to the optimal accuracy much faster which can be explained by the fact that it has already learned meaningful representations of the dataset, such as information about shapes, edges, and textures, which can be beneficial for the classification task. The image colorization task requires the model to understand and capture high-level visual patterns, such as object shapes and semantic content which is also useful for our classifier.

These factors provide the pretrained model with a more informative initialization and a headstart in training as opposed to the random model which must learn everything from the limited labeled data provided by the CIFAR-10 dataset.