

Homework Assignment 2

Deep Learning - 25/26

Žan Pušenjak

1 Introduction

The goal of this project was to implement a fully connected artificial neural network (NN). To improve the performance of the network, several enhancements were added to the basic implementation. The model was trained and evaluated on the CIFAR-10 dataset that is a widely used benchmark for image classification.

The goal of the project was to implement **ResNet**¹ and **UNet**² convolutional network architectures and train and use them on two different datasets for classification and segmentation tasks.

2 ResNet implementation

Firstly we implemented ResNet, more precisely the 18-layer architecture. The goal was to achieve a 90% classification accuracy when annotating bird types in a dataset with 400 different bird types.

To achieve such results we used 15 epochs with batch size 64. The model parameters were set to:

- learning rate: 0.001
- optimizer: SGD
 - weight decay: 1e-5
 - momentum: 0.85

With such network we were able to achieve classification accuracy of **91%**. Some of the false classification examples can be seen on Figure 1.

3 Segmentation

We expended our basic ResNet implementation to be able to semantically segment images. We achieved that by using **Transpose Convolution** to up-sample the last layer back to the original size.

Furthermore we implemented a UNet architecture for the same segmentation task and compared the two by calculating the Intersection over Union metric.

We trained both networks on a part of the Lyft self driving dataset³.

¹<https://arxiv.org/pdf/1512.03385>

²<https://arxiv.org/pdf/1505.04597>

³<https://www.kaggle.com/datasets/kumaresanmanickavelu/lyft-udacity-challenge>



Figure 1: ResNet classification false classification examples

Network	IoU
ResNet	0.4638
UNet	0.5659

Table 1: segmentation results for different networks architectures

The results in Table 1 show that UNet worked much better. Observing the results on Figures 2 and 3 further we can see that the problem ResNet runs into are very imprecise borders. It finds the segments but they are not sharply defined. As we will see in the next task this is the result of only up-sampling one time and not giving the model any skip connections. Since at the last layer the data is very semantically rich but has very little original structural information we get a more *blurred* result.

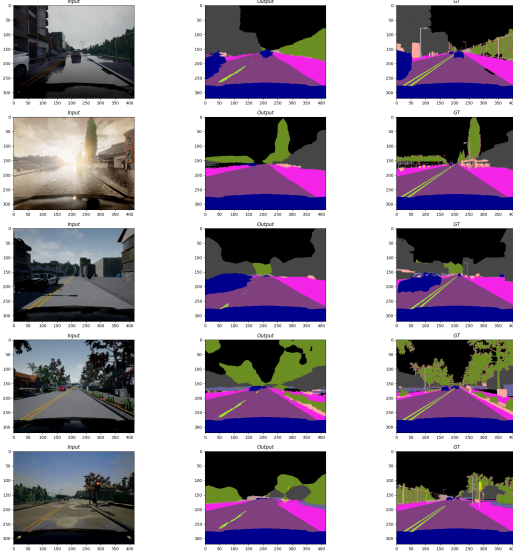


Figure 2: Segmentation results for ResNet



Figure 4: Image colorization with UNet

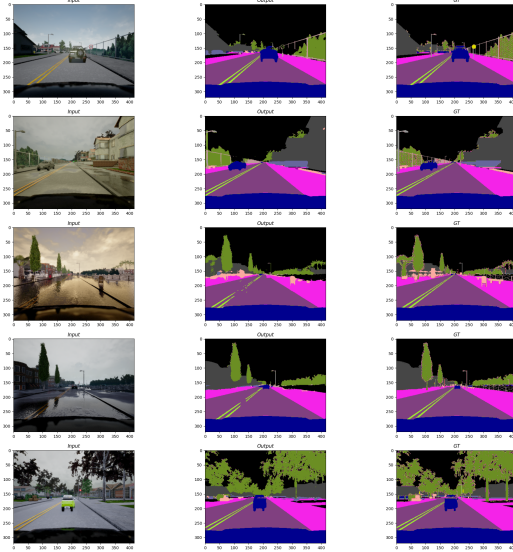


Figure 3: Segmentation results for UNet

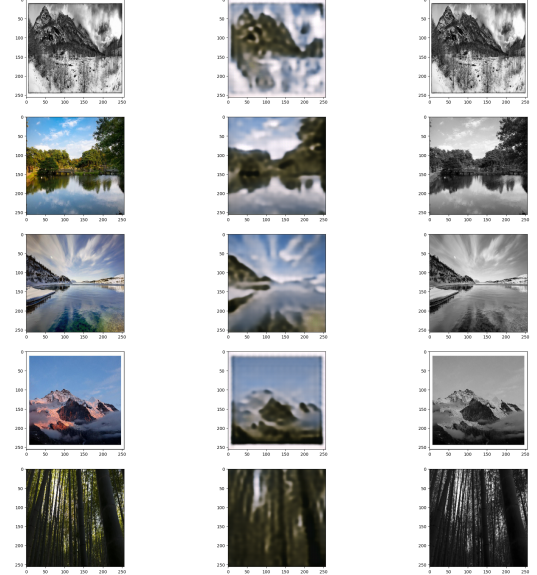


Figure 5: Image colorization with UNet (without skip connections)

4 Image colorization

Lastly we used our previous UNet architecture to color grayscale images. We compared how the network behaves if we remove all of the skip connections.

Even further proofing our findings from previous task we can observe that the network without the skip connections (Table 5) generates very blurry colored images. As explained earlier this is because of the lac of structural data the skip connections provide. The results of the normal network (Figure 4) are impressive considering the network is relatively small and was trained in only a few minutes

5 Conclusion

We were able to train two network architectures on three different datasets by only slightly changing their layers.

The results were very good considering the networks are relatively small and the training time was limited. Here we should mention that the datasets were also very high quality which helped us get the best possible performance out of the models.

In the end the newer UNet architecture clearly outperformed the classic ResNet, showing how the technology and ideas developed over the years and built on top of each other.