Jonas Bürgel (5500163) – Deep Learning Lab 2023 – Computer Vision

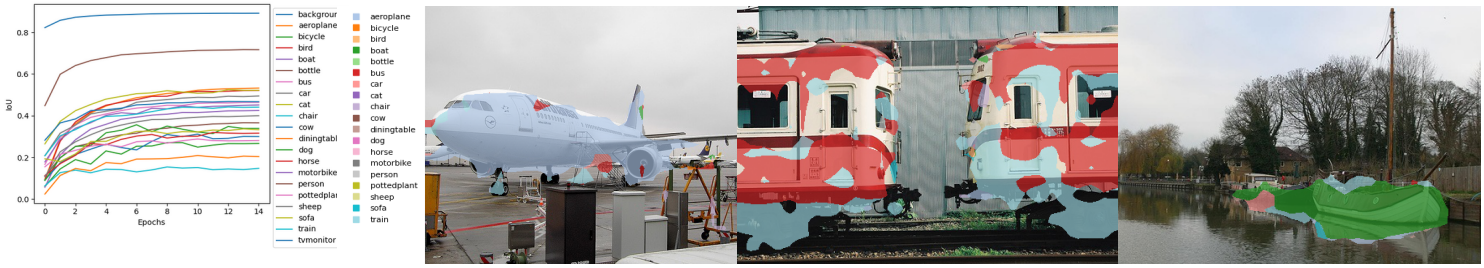
# Deep Learning Lab

Exercise 2

## Task 1: Segmentation

The following sections show train-statistics of different model-heads. The first plot shows the IoU Score of each class. After that 3 samples of the final model are shown. All models were trained for 15 epochs and with the same Hyperparameters and a batchsize of 32 (maximum that was supported by my graphic card). All other Hyperparameters were left to the defaults.

### 1.1 Linear Head




### 1.2 Convolutional Head


The convolutional head, using kernel_size=3 and padding=1, performs slightly better. This could be attributed to the convolutional layer being a better fit for image data, giving the model a slight advantage with a similar number of parameters.

### 1.3 Transformer Head


#### 1.3.1 Transformer Head (with shared Queries and Keys)

The performance of a transformer with shared queries and keys is slightly lower than that of a regular transformer. However, since the shared-queries model has far fewer parameters, this decrease in performance is justifiable. Moreover, the shared-queries transformer still outperforms linear and convolutional heads, just like the regular transformer. Visually, the normal transformer tends to classify more than necessary, whereas the Shared-Queries model tends to classify a smaller area.

Thus, if a smaller model is needed, the shared queries-keys architecture is a valid option.

### 1.4

| Head | IoU |
|---|---|
| Linear | 0.42 |
| Convolutional | 0.48 |
| Transformer | 0.62 |
| Transformer (shared Q&K) | 0.59 |

**How do the different header perform?**

Out of the three types of layers tested, the linear layer had the poorest performance, even worse than the convolutional layer with a similar number of parameters. One reason for this could be that the convolutional layer is a better fit for image data, giving it a slight advantage. On the other hand, transformer models had significantly better performance, but they require a much larger number of parameters. Therefore, if the outcome is crucial and sufficient resources are available, transformer models should be the preferred choice.

Despite having limited resources available for learning, all heads performed quite well, which can be attributed to the large prior-fitted model that they were built upon.

## Task 2: Image-Text

I had done the tasks before the error in the bleu-calculator was fixed and because of energy concerns I did not run all trainings again. All wrong scores are clearly marked to avoid confusion.

### 2.1: Image Captioning

For Task c, my objective was to investigate how the prompt influences the results, and thus, I did not modify any other hyperparameters to avoid interfering with the prompts. Notably, the attempt before the last one, which included the word "strange," proved to be more effective than the previous one that had a long text prompt, even though the word is not neutral. Additionally, the prompt "an image of" yielded remarkably good results.

| Method | Hyperparameter Change | Prompt | Result (BLEU) |
|---|---|---|---|
| Greedy (Task a) | | “a picture of” | 0.406 (wrong)<br>true: 0.28 |
| TopK (Task b-i) | Temp=1.0 | “a picture of” | 0.156 (wrong)<br>true: 0.07 |
| TopK (Task b-ii) | Temp=0.7 | “a picture of” | 0.238 (wrong)<br>true: 0.13 |

| | | | |
|---|---|---|---|
| TopK (Task c) | Temp=1.0 | “the image introduces the viewer to” | 0.05 |
| TopK (Task c) | Temp=1.0 | “the viewer comes upon a strange view of” | 0.09 |
| TopK (Task c) | Temp=1.0 | “an image of” | 0.26 |

### Why does the result improve with lower temperature?

Having a higher temperature makes the output more exploratory and less accurate. In case of this training-setting this improves training but it is a hyperparameter and could be worse in a different scenario. In theory it makes sense to adjust the temperature over the duration of training (from exploratory to precise).

## 2.2 Image-Text Retrieval

The experiments were conducted under the same environment as the initial run (task a), including the number of epochs set to 5. The objective was to investigate how temperature affects training "from scratch." As a result, I discovered that a value between 0.3 and 0.0 for constant temperature seems to work best when training from scratch. However, a temperature of 0.2 could prove more effective for Image-To-Text tasks, while a lower temperature may be better suited for Text-To-Image tasks.

| Method | Hyperparameter Change | Result (R@1) Image to text | Result (R@1) Text to image |
|---|---|---|---|
| Initial (Task a) | | 0.43 | 0.41 |
| Finetune (Task b) | | 0.59 | 0.55 |
| Exp. 1 (Task c) | Temp = 0.5 | 0.32 | 0.33 |
| Exp. 2 (Task c) | Temp = 0.3 | 0.39 | 0.38 |
| Exp. 3 (Task c) | Temp=0.2 | 0.43 | 0.41 |

### How does the “fintuning” task (b) compare to the “initial training” (task a)?

The score achieved through fine-tuning is significantly higher than that of the initial training. This discrepancy can be easily explained by the use of a pre-trained network. Since the network was already trained on multiple images, fine-tuning it to the current images provides a significant advantage from the outset.