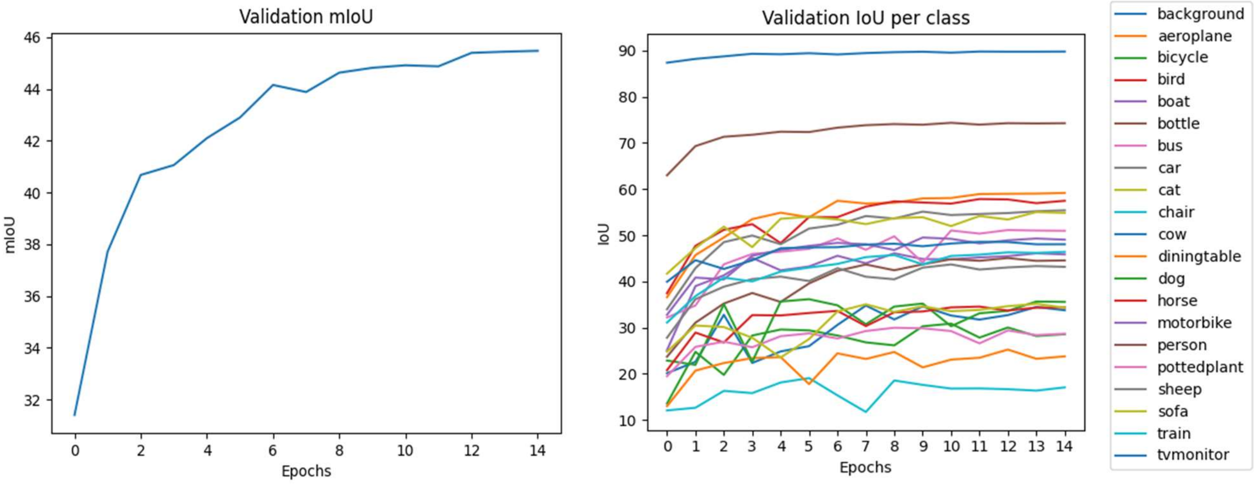
Deep Learning Lab Course

Assignment 2: Segmentation, Image Captioning and Image-Text Retrieval

Soham Basu – 5576954 May 16, 2023

1. Segmentation

1.1 Linear Head



(a) (b)

Figure 1: (a) Validation mIoU per epoch, (b) Validation IoU per class per epoch

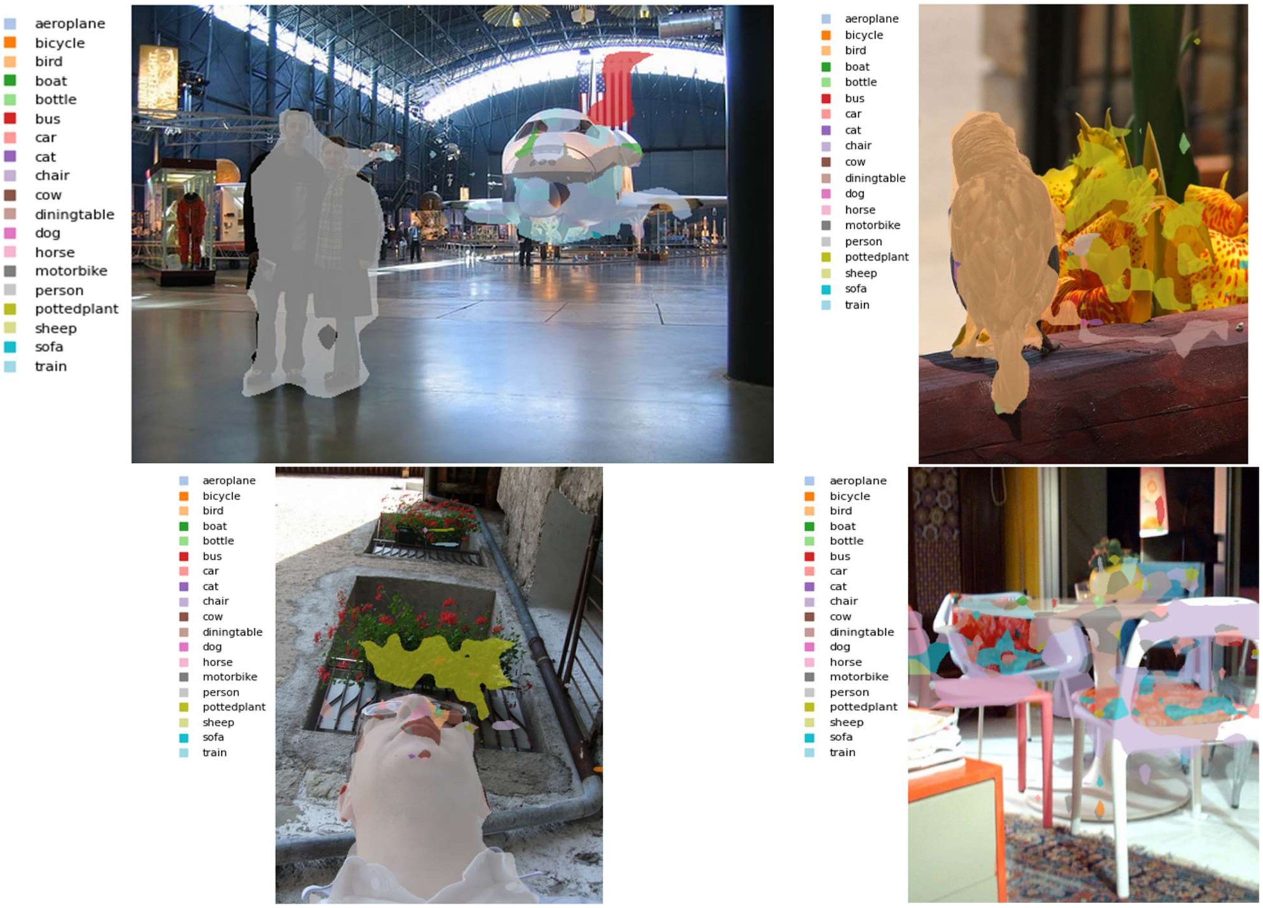
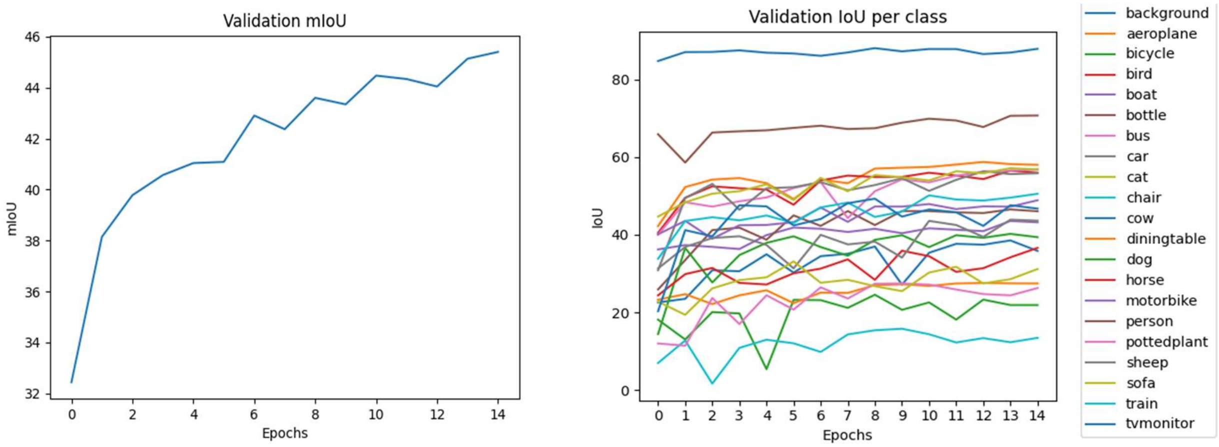


Figure 2: Segmented images from the Validation set

1.2 Convolutional Head



(a) (b)

Figure 3: (a) Validation mIoU per epoch, (b) Validation IoU per class per epoch

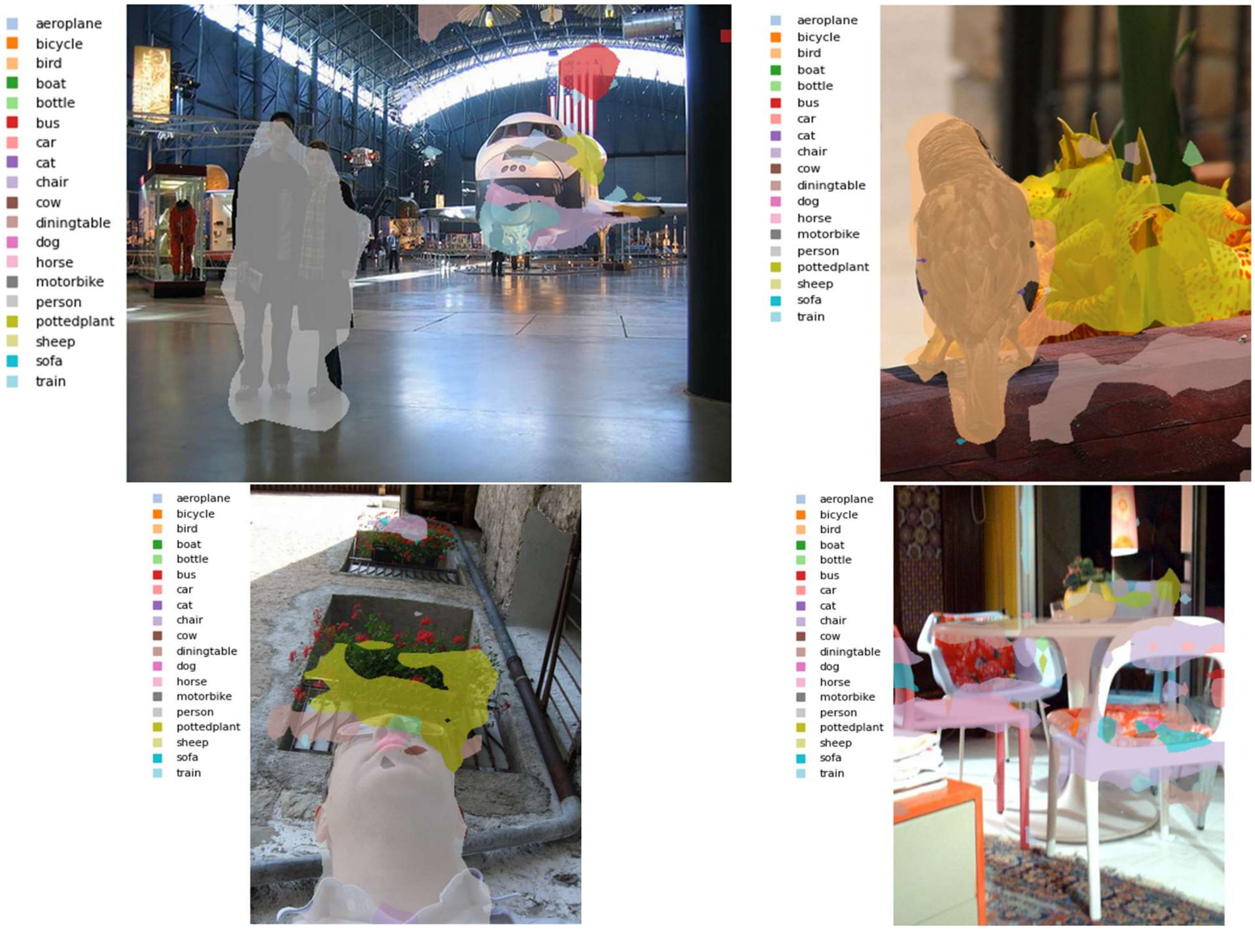
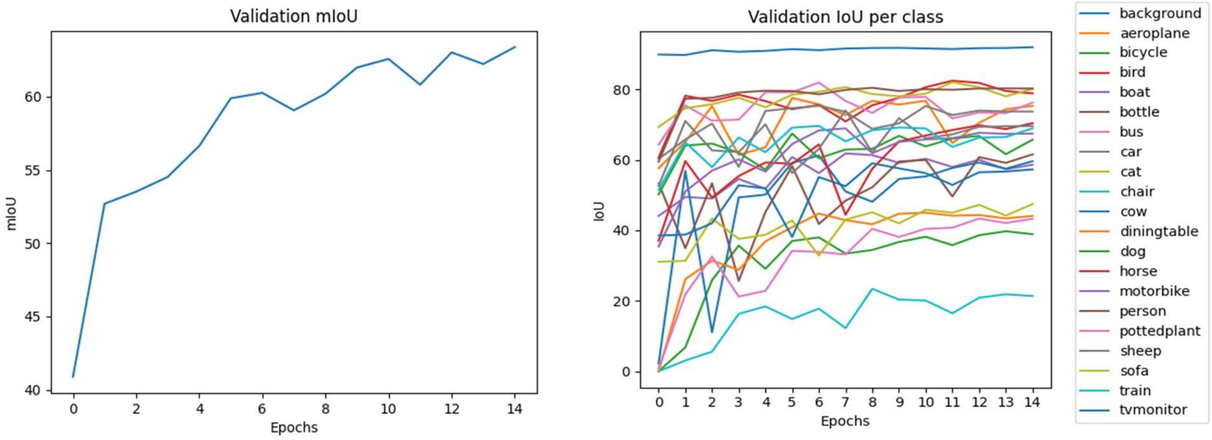


Figure 4: Segmented images from the Validation set

TODO Answer: The Convolutional head implemented consists of a single Conv2D Layer with kernel size 3, stride 1 and no padding or dilation. The head performs ever so slightly worse (0.08%) than the Linear head. Perhaps the 3x3 kernel is unable to capture enough contextual information compared to the Linear head that uses a 1x1 kernel. This is further proven by the fact that a Conv2D layer with 5x5 kernel performs even worse.

1.3 Transformer Head



(a) (b)

Figure 5: (a) Validation mIoU per epoch, (b) Validation IoU per class per epoch

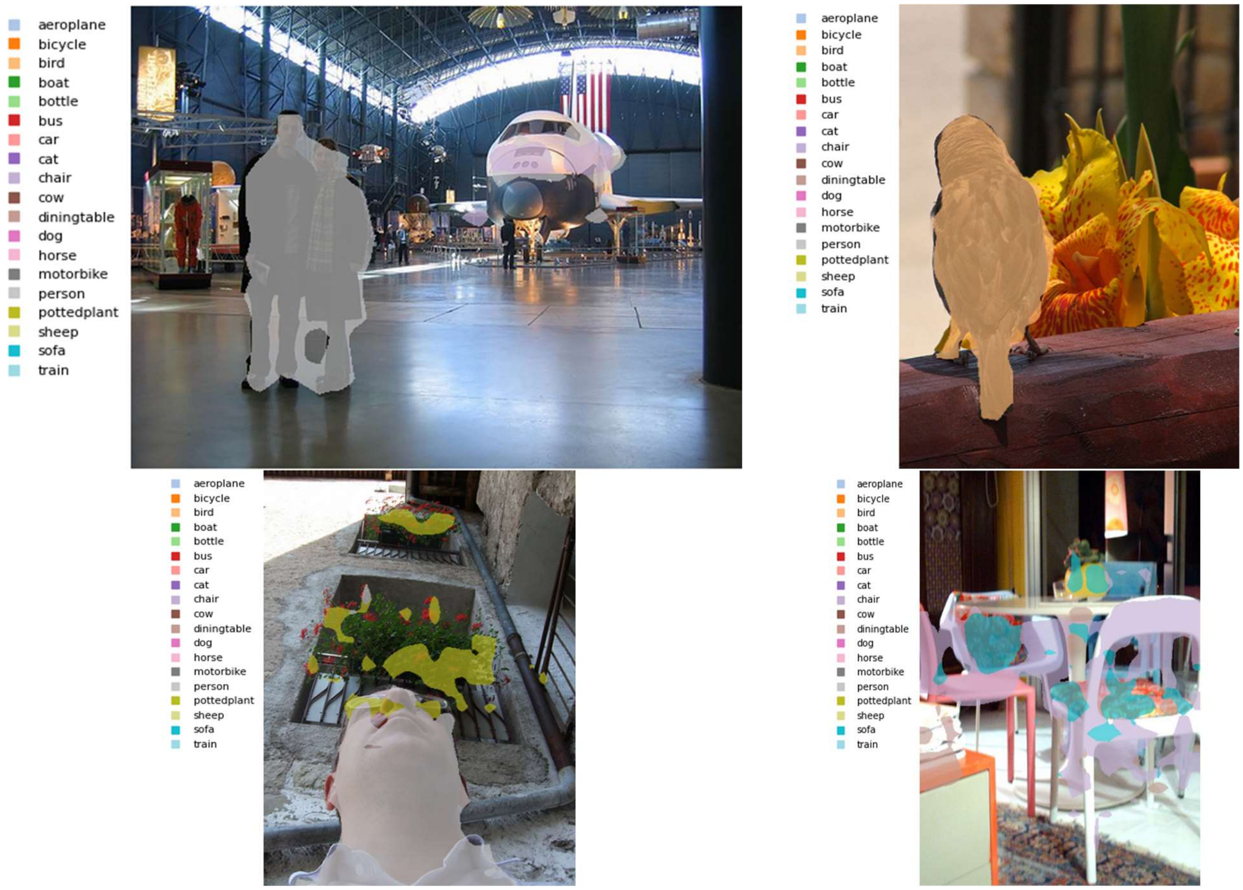
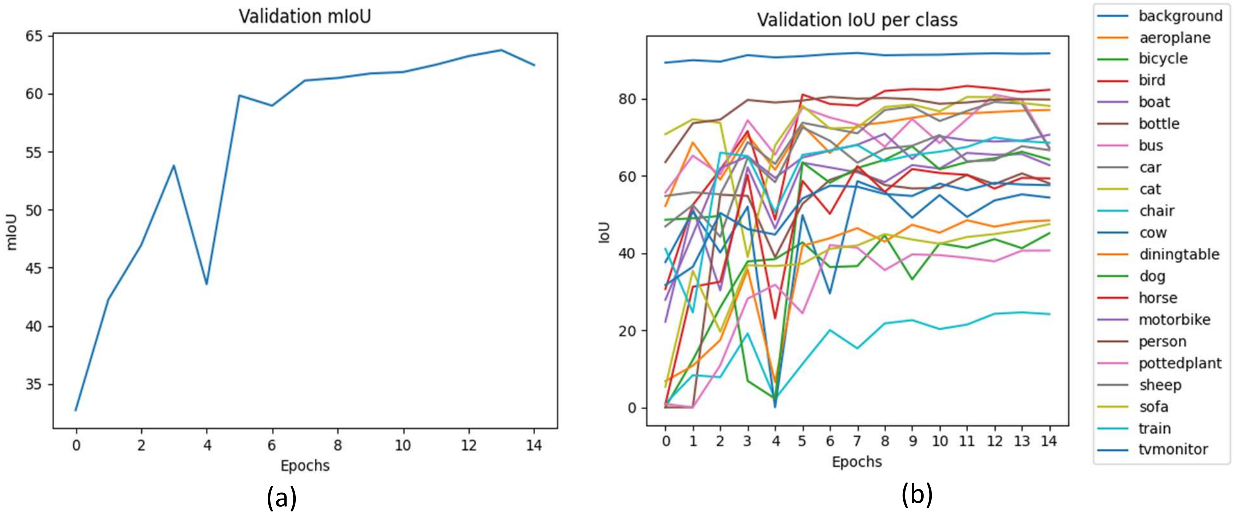


Figure 6: Segmented images from the Validation set

1.3.1 Transformer Head (with shared Q-K)



(a) (b)

Figure 7: (a) Validation mIoU per epoch, (b) Validation IoU per class per epoch

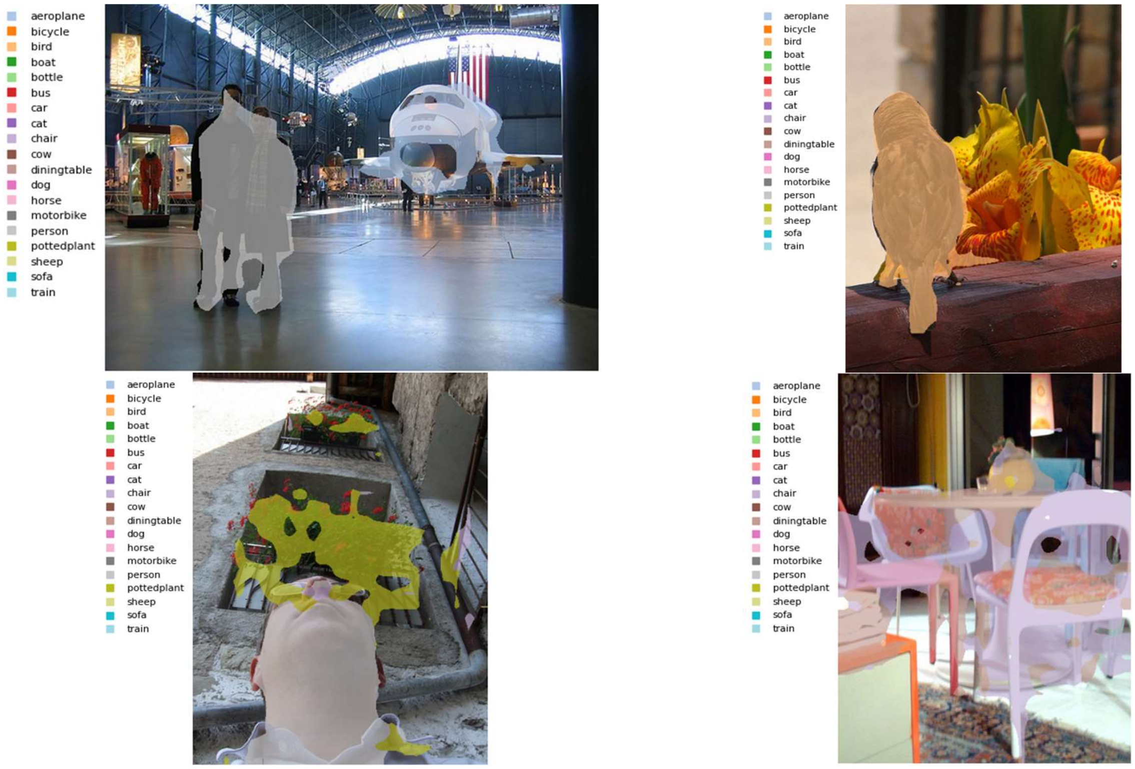


Figure 8: Segmented images from the Validation set

TODO Answers: The transformer head using the same projection layer for keys and queries performs worse than the actual Transformer head. This might be expected because the reason for keys and queries having separate projection layers is to enable the learning of different representations from the input embeddings to better express the spatial relationships and context information in the input. Having the same projection layer doesn't allow the decoder to learn distinct representations and evidently reduces performance.

1.4 Results and Conclusion

Table 1: Image Segmentation Results using different heads

Head	Validation mIoU
Linear	45.48

Convolutional	45.40
Transformer	63.33
Transformer with shared Q-K	62.44

From the experiments, it is quite clear that the Transformer head achieves the best mIoU score of **63.33**. This is expected since Transformers use the Attention Mechanism, which, in a Computer Vision context, helps capture long range spatial dependencies and contexts between pixels far better than CNNs that use local receptive fields. By attending to the relevant parts of the image, transformers can learn better representations of the input feature maps.

2. Image-Text

2.1 Image Captioning

2.1.1 TODOs and Experiments

Table 2: Results of the TODOs with given hyperparameter settings

Type	K	Temperature (T)	prompt	BLEU Score (%)
Greedy search	-	-	“a picture of”	27.9
Top-K sampling	50	1.0	“a picture of”	6.8
Top-K sampling	50	0.7	“a picture of”	11.7

Table 3: Experimental Results with new hyperparameters (* indicates highest BLEU Score)

Type	K	Temperature (T)	prompt	BLEU Score (%)
Top-K sampling	50	0.7	"this image shows"	10.2
Top-K sampling	50	0.1	“a picture of”	26.29
Top-K sampling	30	0.7	“a picture of”	12.09
Top-K sampling	10	0.1	“a picture of”	26.77
Top-K sampling	5	0.1	“a picture of”	28.14*
Top-K sampling	5	0.05	“a picture of”	27.99
Top-K sampling	1	0.1	“a picture of”	27.95
Greedy search	-	-	“this picture shows”	27.59

2.1.2 Conclusion

TODO Answer: Temperature seems to have a significant effect on the BLEU score. Lowering the temperature improves the score quite a lot. This is probably because the temperature is used to change the sharpness of the peaks in the probability distribution where Top-K sampling is applied. Lowering the temperature increases the sharpness, thereby creating peaks in the distribution and only words with very high probabilities are chosen. This improves the BLEU score. The best BLEU score obtained was **28.14%** (Table 3).

2.2 Image-Text Retrieval

2.2.1 TODOs and Experiments

Table 4: Results of the TODO tasks with given hyperparameter settings

Mode	Learning Rate	Weight Decay	Temperature	Epochs	R@1 (%)
Evaluating with the provided checkpoint	-	-	-	-	53.89
Training from scratch	1e-3	1e-3	0.1	5	43.96
Finetuning checkpoint	1e-5	0	0.1	3	58.94

TODO Answer: After Finetuning the BLIP retrieval head, an R@1 score of 58.94% is obtained. This score achieved here is quite higher compared to when training the model from scratch. This is probably because the BLIP checkpoint the head was loaded from was already pretrained on a large dataset with 129M images. Evaluation results already show an R@1 of 53.89% (Table 4). Thus, finetuning this checkpoint with the right hyperparameters would only get us better results.

Table 5: Experimental Results with new Hyperparameters (* indicates highest R@1)

Mode	Learning Rate	Weight Decay	Temperature	Epochs	R@1 (%)
Training from scratch	1e-3	1e-3	0.1	8	46.79
Training from scratch	1e-2	1e-3	0.1	8	42.17
Training from scratch	1e-4	1e-3	0.1	8	36.71
Training from scratch	3e-3	1e-4	0.1	5	42.03
Training from scratch	1e-3	5e-4	0.1	8	47.83
Training from scratch	1e-3	5e-4	0.5	8	32.02
Training from scratch	3e-4	5e-2	0.1	8	44.03
Training from scratch	1e-3	5e-2	0.1	8	48.31*
Training from scratch	1e-3	0	0.1	8	46.31

2.2.2 Conclusion

It's quite easy to improve over the 43% image-to-text R@1 obtained with the default hyperparameters. Training from scratch for 8 epochs with randomly initialized weights, learning rate set to 1e-3, weight decay of 5e-2 and temperature 0.1 gives us the best R@1 of **48.31%**. However, improving over the 54% R@1 of the pretrained checkpoint doesn't seem to be possible after training for only a few epochs and on a much smaller dataset.