Autonomous Vision Group
Prof. Dr.-Ing. Andreas Geiger

Eberhard Karls
Universität
Tübingen

Computer Vision

Exercise 6 – Self-Supervision and Diverse Topics

6.1 Pen and Paper

Self-Supervised Optical Flow and Depth

a) You are given the following sequence of 5×7 images, where part of the background is occluded by a thin structure (represented by the column of 10s). The sequence has 4 frames, denoted by t=1 to t=3.

1	2	3	10	5	6	7
1	2	3	10	5	6	7
1	2	3	10	5	6	7
1	2	3	10	5	6	7
1	2	3	10	5	6	7

(a) t=1

1	2	3	10	5	6	7
1	2	3	10	5	6	7
1	2	3	10	5	6	7
1	2	3	10	5	6	7
1	2	3	10	5	6	7

(b) t=2

1	2	10	4	5	6	7
1	2	10	4	5	6	7
1	2	10	4	5	6	7
1	2	10	4	5	6	7
1	2	10	4	5	6	7

(c) t=3

2	10	4	5	6	7	8
2	10	4	5	6	7	8
2	10	4	5	6	7	8
2	10	4	5	6	7	8
2	10	4	5	6	7	8

(d) t=4

Try to annotate the 5×7 optical flow labels in the x and y directions for the three frame transitions in the video sequence (t=1 to t=2, t=2 to t=3 and t=3 to t=4). You can assume zero padding at the image boundaries where necessary. Record the total time you spent labeling these images with a stopwatch. Assuming you can annotate a real video with the same speed, approximate the time required to label the optical flow for a 10 second video sequence of frame-rate 30 fps and resolution 1024×1024 pixels. Based on your findings, do you think it is feasible to manually annotate optical flow datasets for training deep networks?

b) Do you think it is harder to perform self-supervised optical flow and depth estimation for images of a vehicle with well-polished, shiny paint or one with a dull painted surface? Explain your reasoning.

c) In the stereo lecture, we discussed the benefits of left-right consistency checks. Similarly, it is common in optical flow networks to perform bi-directional training, where the network estimates both the forward and backward optical flow. Does a forward-backward consistency loss for optical flow have the same purpose as a left-right consistency loss in stereo models? If not, discuss the differences.

d) Self-supervised monocular depth estimation recovers the disparity map for the input image (where the units are in pixels). Is it possible to convert this into an actual depth map with physical units (meters)? If so, how would this be done?

e) The training objective (loss function) does not change for self-supervised depth prediction when using stereo images rather than nearby frames in a video sequence. However, stereo data requires a specialized camera setup to collect. Given this, mention two key advantages of stereo-based training.

Pretext Tasks

a) A key problem in designing pretext tasks for self-supervised learning is the tendency of the network to prioritize trivial shortcuts over meaningful learning. Considering the approach discussed in the lecture of learning by solving jigsaw puzzles, what are the three main shortcuts exploited by the network, and the solutions proposed to overcome these?

b) Consider the following two pretext tasks: (i) predicting if an image has been horizontally flipped, (ii) predicting if an image has been vertically flipped. When trained on a dataset of images from the internet, which of the two pretext tasks would you expect to lead to better self-supervised representations, and why?

c) If the same two pretext tasks from the previous question were applied to satellite images (eg., `https://earthview.withgoogle.com/`), which of the two would you now expect to lead to better representation learning?

Contrastive Learning

a) An important design choice in nearly all contrastive learning methods is the score function or similarity metric. Given two features $\mathbf{f}_1$ and $\mathbf{f}_2$, write down the mathematical equation for the following 3 commonly used score functions: (i) L_1 (Manhattan) distance, (ii) L_2 (Euclidean) distance, and (iii) cosine similarity. Rank the three functions based on how susceptible you think they are to outliers when used in high-dimensional feature spaces.

Remark: *An outlier here refers to an example for which the features lie far away from the mean of the data distribution.*

b) The InfoNCE loss for 1 reference (x), 1 positive (x^+) and N-1 negative (x_j^-) examples is given as follows:

$$\mathcal{L} = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

Consider an alternate loss function where there are no negative examples, such that the denominator is removed from the InfoNCE loss as follows:

$$\mathcal{L} = -\mathbb{E}_X \left[s(f(x), f(x^+)) \right]$$

What is the problem with this loss function in practice, which leads to the need for negative examples during training?

Input Optimization

a) Explain the difference between white-box and black-box adversarial attacks. Which of the two types of attacks is harder to achieve in practice?

b) Consider the following 2×2 image:

$$\begin{bmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{bmatrix}$$

We apply two element-wise feature extractors, F_1 and F_2, to obtain the following feature channels:

$$\begin{bmatrix} F_1(\mathbf{a}) & F_1(\mathbf{b}) \\ F_1(\mathbf{c}) & F_1(\mathbf{d}) \end{bmatrix}, \begin{bmatrix} F_2(\mathbf{a}) & F_2(\mathbf{b}) \\ F_2(\mathbf{c}) & F_2(\mathbf{d}) \end{bmatrix}$$

In style transfer applications, the style of an image is expressed as a Gram Matrix:

$$G_{ij} = \sum_{\mathbf{p} \in \Omega} F_i(\mathbf{p}) F_j(\mathbf{p})$$

Express the 2×2 Gram Matrix G for these two feature channels in terms of F_1, F_2, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$.

Hint: *Flatten the image into a 4-dimensional vector and represent the feature map as a 2×4 matrix before computing the Gram Matrix.*

6.2 Coding Exercises

The coding exercises use the jupyter notebook `code/self_supervision.ipynb`. You will implement a k-Nearest Neighbor classifier for CIFAR-10 using self-supervised representations from the *Barlow Twins* approach discussed in the lecture. As in the previous exercises, the notebook is self-contained. The dataset will be downloaded to your working environment as you execute the functions in the notebook.

When you see helper functions in the notebook, you don't need to do anything– they are already implemented. The functions you need to implement are indicated as "Exercise Function". For this exercise, there are **four exercise functions** in total. When you are done, you can explore the approach in more detail if you like, by re-training the model with different augmentations and hyper-parameters.