Image Classification

Image classification: assign label to a input image

Challenges:

- Viewpoint variation
- Scale variation
- Deformation
- Occlusion: when only a part of an object is visible.
- Illumination conditions
- Background clutter: when the objects are blended o the background
- Intra-class variation

A good classification must be invariant to these aspects.

Data-driven approach: a range of image examples are given to a computer for make it learn.

Pipeline:

- Input: N images labeled with one of K classes (training set)
- Learning: train a classifier or learn a model
- Evaluation: test classifiers with images outside the training set and compare to their true labels (ground truth) to analyse performance

Nearest Neighbor Classifier:

- L1 distance: calculates the differences between the same location pixels of two images, then sum it all up. $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$
- L2 distance: $d_2(I_1, I_2) = \sqrt{\left(\sum_p |I_1^p - I_2^p|\right)^2}$
- L1 vs. L2: L2 distances prefers many medium disagreements to one big one.

k-Nearest Neighbor Classifier: instead of finding the single closest image in the training set, find the top k closest images and have them voe on the label of the test image.

Validation sets for Hyperparameter tuning

- Hyperparameter: which norm to use, which k to pick…
- Do not use the test set for the purpose of tweaking hyperparameter, because it is possible to adjust the model to work well on the test set, but in the real world the performance is low. It is called overfit.
- Validation set: a “fake test-set” to tune the hyper-parameters
- Cross-validation: estimate of how well a value of k works by iterating over a different validation sets and averaging the performance across these.

Pros and cons of NN:

- Advantage: no time to train
- Disadvantage: computational cost at the test time.