**Exercise for MA-INF 2213 Computer Vision SS21**
**25.06.2021**
**Submission on 06.07.2021**

1. **Bag-of-Words model** *(5 Points)*
   Implement a Bag-of-Words (BoW) model [1] following the subsequent steps:
   (a) Extract SIFT keypoints and features [3] for all images.
   (b) Build a BoW dictionary of 54 words by grouping the features of the previous step (only the *training set*) using K-Means.
   (c) Extract for each image the histogram that corresponds to the BoW dictionary of the previous step.
   (d) *Hint: Read/Write your BoW directory from/to disc to save some time.*

   A dataset[1] of 265 images containing 53 object classes is provided in the directory `./images`. Each image contains only one object instance and can be thus labeled with its name. The dataset contains 5 images per class. The first image should be used just for testing (*testing set*), while only the last 4 images should be used for training (*training set*).

   Since feature extraction is not the main focus of the exercise, the following (or similar) `OpenCV` structures and methods can be used for this step (Exercise 1): `cv2.BOWKMeansTrainer`, `cv2.FlannBasedMatcher`, `cv2.BOWImgDescriptorExtractor`, `cv2.xfeatures2d.SIFT_create`.

   In case the library `cv2.xfeatures2d` is not found, reinstall python-opencv via
   `conda install -c conda-forge opencv`
   if you use Anaconda. If you use pip, install opencv via
   `pip install opencv-python==4.5.1.48`
   `pip install OPENCV-CONTRIB-PYTHON=4.5.1.48`

2. **k-Nearest-Neighbor Classification** *(5 Points)*
   Classify the *testing set* images of the dataset using the SIFT BoW model of the previous step and a *k-Nearest-Neighbor* classifier with a Euclidean distance metric. Compute and print the percentage of false assignments for $k = 1$ and $k = 3$.

3. **Metric Learning** *(10 Points)*
   Implement a classifier for the provided dataset following the approach of [2].
   (a) Based on similarity[2] define all the similar ($y_{ij} = 1$) and dissimilar ($y_{ij} = 0$) pairs of the training set.
   (b) Compute the covariance matrices of the above mentioned sets (Equations 12, 13 of [2]):

$$\Sigma_{y_{ij}=1} = \frac{1}{N_{y_{ij}=1}} \sum_{y_{ij}=1} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{1}$$

$$\Sigma_{y_{ij}=0} = \frac{1}{N_{y_{ij}=0}} \sum_{y_{ij}=0} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{2}$$

   where $(i, j)$ define a pair of images, $\mathbf{x}$ is a feature vector and N denotes the cardinality of the specified set (*similar* and *dissimilar* image pairs).

[1] A modified version of the dataset `http://www.vision.ee.ethz.ch/datasets_extra/Obj_DB.tar.gz`
[2] A similar pair is a pair of images of the same object from a different viewpoint.

(c) Learn a squared Mahalanobis distance metric using (non-numbered equation that follows Equation 16 in [2]):

$$\hat{\mathbf{M}} = \Sigma^{-1}_{y_{ij}=1} - \Sigma^{-1}_{y_{ij}=0} \tag{3}$$

(d) Verify that the learned metric is positive semi-definite by checking if all eigenvalues are positive. In the opposite case recover a positive semi-definite metric $\mathbf{M}$ by re-projecting $\hat{\mathbf{M}}$ onto the cone of positive semi-definite matrices by:

- Replacing all negative eigenvalues with a small positive value (0.001).
- Using the formula

$$M = P \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} P^{-1} \tag{4}$$

where $\mathbf{M}$ is the positive semi-definite matrix to be reconstructed, $P$ is a matrix whose columns are the eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ and $\lambda_1, \dots, \lambda_n$ are the corresponding eigenvalues.

**Hint:** You can use the method `np.linalg.eig` of `numpy` for eigenanalysis.

(e) Classify the *testing set* images of the dataset using the SIFT BoW model of the first step and a *k-Nearest-Neighbor* classifier with the learned (squared) Mahalanobis distance metric (Equation 16 of [2]):

$$d^2_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T M (\mathbf{x}_i - \mathbf{x}_j) \tag{5}$$

Compute and print the percentage of false assignments for $k = 1$ and $k = 3$.

You are not allowed to use any library apart from the ones provided in the template.

# References


[1] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in C.Vision, ECCV*, 2004.

[2] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.

[3] D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.