Kernel Methods for Machine Learning - Data Challenge

Les dénoyauteurs - 8th position (best accuracy 61.3%)

Chaïmaa Kadaoui (MVA) **Luca Grementieri (MVA)** **Pierre-Alain Langlois (MVA)**

1 Kernel SVM

Given a train data X of color images and labels $y \in \{0, \dots, 9\}$, the goal of the challenge is to predict the labels. For our whole pipeline, we chose to implement a Kernel SVM with intercept as seen in the course. The dual formulation of Kernel SVM is:

$$\min_{\alpha} \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad \text{subject to the constraints} \quad \sum_j y_j \alpha_j = 0 \text{ and } 0 \leq \alpha_j \leq C.$$

K is the chosen kernel function. The score function for one image x is then: $f(x) = \sum_i \alpha_i \cdot y_i \cdot K(x_i, x) + b$.

We fit one SVM for each class versus the others. Afterwards we estimate the posterior probability of each class using the method in [1]. To do so, we suppose that the posterior probability can be expressed as: $p(class = 1|f) = \frac{1}{1+\exp(Af+B)}$. To find A and B, we take for each image i,, $t_i = \frac{y_i+1}{2}$ and we solve: $\min_{A,B} - \sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i)$ where $p_i = p(class = 1|f(x_i))$. At the end, the predicted class is the one with the maximum posterior probability.

2 Hierarchical kernel descriptors

Our best submission is based on the method explained in Bo et al. [2], whose main idea is to approximate infinite-dimensional points of the RKHS associated to the Gaussian kernel with a finite representation in order to create a feature vector. Features vectors can be used as new data to iterate the same procedure many times, in this way we can obtain a hierarchy of descriptors that summarizes features from different scales. In particular, since the provided images are very small the hierarchy is composed by two layers: patch-level descriptors and image-level descriptors.

2.1 Patch-level kernel descriptors

Patch-level kernel descriptors are implemented according to Bo et al. [3]. In order to compare patches we define three match kernels based on the gradient, the intensity value and the local pattern of every pixel in the patch. For every pixel in a patch we indicate z its position normalized to [0,1], $m(z), \theta(z)$ the magnitude and the direction of the gradient, $c(z)$ the intensity value, $s(z)$ the standard deviation of intensity values in the 8-neighborhood of the pixel and $b(z)$ a boolean 8-dimensional vector such that $b(z, i) = 1$ if $c(z_i) > c(z)$, 0 otherwise (using i as an index of the 8-neighborhood of the pixel). Let P, Q be two patches, $k_a(x, x') = \exp(-\gamma_a ||x - x'||^2)$ the Gaussian kernel indexed by a and ϵ_g, ϵ_s small constants, then the used patch-level match kernels are

$$K_{grad}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} \widetilde{m}(z)\widetilde{m}(z') k_o(\widetilde{\theta}(z), \widetilde{\theta}(z')) k_p(z, z')$$

$$K_{col}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} k_c(c(z), c(z')) k_p(z, z')$$

$$K_{shape}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} \widetilde{s}(z)\widetilde{s}(z') k_b(b(z), b(z')) k_p(z, z'),$$

where $\widetilde{m}(z) = \frac{m(z)}{\sqrt{\sum_{z \in P} m(z)^2 + \epsilon_g}}$, $\widetilde{\theta}(z) = [\sin(\theta(z)) \cos(\theta(z))]$ and $\widetilde{s}(z) = \frac{s(z)}{\sqrt{\sum_{s \in P} s(z)^2 + \epsilon_s}}$.

Every match kernel can be expressed as a scalar product, for example

$$K_{grad}(P, Q) = \langle \sum_{z \in P} \widetilde{m}(z) \phi_o(\widetilde{\theta}(z)) \otimes \phi_p(z), \sum_{z' \in Q} \widetilde{m}(z') \phi_o(\widetilde{\theta}(z')) \otimes \phi_p(z') \rangle = \langle F_{grad}(P), F_{grad}(Q) \rangle$$

The vectors $\phi_o(\widetilde{\theta}(z)))$ and $\phi_p(z)$ are infinite-dimensional but we approximate them with their projection on the subspace of the RKHS generated by points chosen over a regular grid of the support region. In order to obtain a (reduced) orthogonal basis of this subspace we use KPCA and the final kernel descriptor is the vector of the coefficients obtained projecting every point to this reduced basis.

For example, in the case of K_{grad}, if $\{x_i\}_{i=1}^{d_o}$ is a family of normalized gradient vectors and $\{y_j\}_{j=1}^{d_p}$ is a family of normalized positions, then the selected subspace is generated by $\{\phi_o(x_1) \otimes \phi_p(y_1), \dots, \phi_o(x_{d_o}) \otimes \phi_p(y_{d_p})\}$. We extract T kernel principal components from this set of generators obtaining the family of vectors $\{\alpha^t\}_{t=1}^T$ and so the t-th kernel principal component is $PC^t = \sum_{i=1}^{d_o}\sum_{j=1}^{d_p}\alpha_{ij}^t\phi_o(x_i)\phi_p(y_j)$. Finally, the t-th component of the T-dimensional gradient kernel descriptor is

$$\overline{F}_{grad}^t(P) = \langle F_{grad}(P), PC^t\rangle = \sum_{z\in P}\widetilde{m}(z)\sum_{i=1}^{d_o}\sum_{j=1}^{d_p}\alpha_{ij}^t k_o(\widetilde{\theta}(z), x_i)k_p(z, y_j).$$

We have chosen the set of generator for k_o, k_c and k_p from 24, 5 and 5×5 uniform grids respectively and used the recommended hyperparameters $\gamma_o = 5$, $\gamma_p = 3$, $\gamma_c = 4$, $\gamma_b = 2$, $\epsilon_g = 0.8$, $\epsilon_s = 0.2$, $T_{grad} = 200$, $T_{col} = 50$, $T_{shape} = 200$. From every image we have selected 8×8 patches over a regular grid with a spacing of 2 pixels, i.e. 169 patches per image.

2.2 Image-level kernel descriptors

To obtain the kernel descriptor of an image we iterate the same procedure. For every patch P in an image we indicate C_P its position normalized to [0,1], W_P the average of the descriptor coefficient over the patch (i.e. the average of $m(z)$, 1 or $s(z)$ over the patch P) and F_P the gradient, color or shape kernel descriptors. Let $\mathcal{I}, \mathcal{J}$ be two images and ϵ_h a small constant, then for every type of kernel descriptors we can define the corresponding image-level match kernel as

$$K_h(\mathcal{I},\mathcal{J}) = \sum_{P\in\mathcal{I}}\sum_{P'\in\mathcal{J}}\widetilde{W}_P\widetilde{W}_{P'}k_h(F_P, F_{P'})k_C(C_P, C_{P'}) \quad \text{with} \quad \widetilde{W}_P = \frac{W_P}{\sqrt{\sum_{P\in\mathcal{I}}W_P)^2 + \epsilon_h}}.$$

Here the feature space is too big to be sampled with an uniform grid so we set the subspace generators as the centroids obtained with k-means clustering over kernel descriptors (1000 clusters). In order to perform the KPCA on this enormous generator set we look for the highest T eigenvectors of $\mathbf{K}^c = \mathbf{K}_F^c \otimes \mathbf{K}_C^c$, called $\{\beta^t\}_{t=1}^T$, among the Kronecker product of the eigenvectors of $\mathbf{K}_F^c$ and $\mathbf{K}_C^c$. This procedure is much more efficient than the direct eigenvector computation on the big matrix $\mathbf{K}^c$.

Let $\{X_i\}_i^{d_F}$ be the family of centroids of the kernel descriptors and $\{Y_j\}_j^{d_C}$ the family of normalized patch positions, then the t-th component of the T_h-dimensional hierarchical kernel descriptor is

$$\overline{F}_h^t(\mathcal{I}) = \langle F_h(\mathcal{I}), PC^t\rangle = \sum_{P\in\mathcal{I}}\widetilde{W}(P)\sum_{i=1}^{d_F}\sum_{j=1}^{d_C}\beta_{ij}^t k_h(F_P, X_i)k_C(C_P, Y_j).$$

The generator vectors for k_C are picked from a 5×5 uniform grid and we used the hyperparameters $\gamma_h = 1$, $\epsilon_h = 0.5$ for every kernel. By cross-validation we have selected the values $T_G = 800$, $T_C = 20$, $T_S = 800$ and we have performed our predictions using a linear SVM on the features vectors obtained by the concatenation of image-level gradient, intensity and shape image-level kernel descriptors.

3 Implemented methods

During the project, we have implemented other methods that were finally not used in the final proposal. Regarding the SVM, we finally used the one vs all method as explained above. However, we also tested the one vs one method in which we need to train one SVM for each couple of class (45 SVMs in our case). The final decision is then taken by a max voting. Unfortunately, we found experimentally that this method would not give better result during cross-validation.

The first features that we tried to extract were the histograms of gradients (hog). More precisely, we first tried to divide the picture into a grid and to compute a feature which is the concatenation of the obtained hogs. The grid allows here to preserve spatial information, but it is an arbitrary division that does not focus on important point. We tried to select more relevant points thanks to the Harris cornerness measure and the adaptative non-maxima suppression but this would not improve the results.

We put also some effort into implementing the bag of visual words algorithm. This method consists in performing clustering in the set of all histograms of gradients (hog) acquired in the dataset thanks to the k-means algorithm (k is here a parameter). Then, we train an SVM-based model on the histograms of the hog that belong to each cluster for each image. After extensive testing and cross-validation, we found that this method did not improve the basic method presented above.

We also started experimentations with the scattering networks, but this part remained unfinished since we obtained good results with the hierarchical kernel method.

References

[1] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. 1999.

[2] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object recognition with hierarchical kernel descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011.

[3] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *Advances in neural information processing systems*, 2010.