GRENOBLE UNIVERSITY

REPORT

Data Challenge : Kernel Methods

Students :
SOUFIANE KARIMI
AUGUSTINA AMAKOR

Professor :
Julien MAIRAL

10 février 2022

Table des matières

1 Methodology

The purpose of the challenge is to classify images which are not properly represented. Thus, as a preprocessing step, we have firstly augmented the data and then extracted features by implementing a SIFT algorithm. That's allows to have less and better dense features to be fed in classification algorithms.
To classify the different classes, the approach taken was to test multiple classifiers (Ridge Regression, SVM) associated with diverse kernels (linear, gaussian, quadtratic, RBF, chi-square, Laplacian) and to tune its corresponding parameters.
Finally, SVM with RBF kernel performs the best.

2 Data Check and Visualization

As a first step, we check that the data is properly distributed over the classes. Indeed,in the training set, each class is associated with 500 samples.
Then, we visualize certain images to get an idea of their appearance.We remark then that most of the images appear blurred.

3 Data Augmentation

We firstly extend the size of the training set in order to help the algorithm better generalize using data augmentation techniques carrying some flipping vertically and horizontally operations.
we choose this techniques because we noticed that this transformed images are very rare in both training and test set. However, we haven't taken into account rotations and modifying brightness, since SIFT is invariant on rotations, and brightness aspect is already taken into consideration in SIFT.

4 Feature Extraction : SIFT

SIFT is an algorithm used in computer vision to compare two images.If two images are similar, they will have similar SIFT descriptors, otherwise the descriptors will be different. The implementation we have used for this project focused on the construction of these descriptors by creating a grid on the image .
Around each element(Key Point) of the grid, we starts by modifying the local coordinate system to guarantee invariance to the rotation, using a rotation of angle equal to the orientation of the key point, but of opposite direction. At each point, the orientation and amplitude of the gradient are calculated. Then, the retrieved histograms are concatenated and standardized. In order to reduce the sensitivity of the descriptor to changes in brightness, the values are capped and the histogram is again normalized, finally providing the SIFT descriptor of the key point.

5 Algorithms

We test algorithms with a set of diverse kernels, namely linear, gaussian, quadtratic, RBF, chi-square, and laplacian kernels and we realize that SIFT with RBF kernel performs the best with the augmented data.
We also use kernels instead of feature vectors because the computation is much easier with kernels, we then replace the dot product term with kernel, hence we don't have to use feature vector at all.

5.0.1 Ridge Regression

To solve the Equation associated to KRR, we use scipy.linalg module.
And to convert reel predictions to integers, we have taken absolute value of the prediction.

5.0.2 Logistic Regression

The way we used to solve KLR is to iteratively solve a Weighted Kernel Ridge Regression problem until convergence by fixing a tolerance variable as constraint.

5.0.3 SVM

We implement a one vs all SVM approach, indeed, we train our model using 10 classifiers.
In order to solve the convex optimization, we use QP solver (from the cvxopt package) which stores the entire kernel matrix.

6 Parameters tuning

In order to tune the parameters, we used 6- fold cross validation.
The parameters we take into consideration are SIFT parameters(Size of patches, number of angles per patch), parameter of L2 regularisation and kernels...

7 Results

We obtained our final model using data augmentation(flipping), extracting features using SIFT algorithm before, and then using a multi-class SVM (one-versus-all) algorithm associated with Rbf kernel to distinguish the 10 different classes.
We used 10-cross validation to locally test our algorithms before submitting, finally, the model score 66% on kaggle leaderboard.