Image classification using kernel methods

Hippolyte Pilchen, Lucas Gascon, Team name Hip&Gasc*

1 Abstract

During this challenge, we implemented several methods to classify images in 10 classes. Features extractors are computing using both handmade code and implemented libraries. Then, kernel methods such as kernel PCA and kernel SVM are used to finally perform classification. We achieve a test score of 0.592 on the public leaderboard ranking 2^{nd} at the first set deadline.

2 Introduction

This report is related to an image classification challenge in which we were tasked with developing our own kernel methods and feature extraction techniques from the ground up. Initially, we implemented all the methods we discuss in this report on our own using only NumPy. However, as the challenge progressed, we received new instructions and pre-existing libraries for the feature extraction process were authorized. This switch to using libraries, which turned out to be more efficient, enhanced our already good performance in the competition's rankings. However, we followed the instructions and did not use any autograd methods using scikit-image and CVXOPT libraries only[1]. All the code and necessary resources can be found here[2].

Kernel methods for image classification

To classify an image with a Support Vector Machine (SVM), we must initially extract characteristics from the image. These characteristics could include pixel color values, edge detection, or the textures within the image. After these features are identified, they serve as inputs for the SVM algorithm. The SVM operates by identifying the hyperplane that best divides the various classes within the feature space. A key benefit of employing kernel SVMs is that some kernel can map the input images in a feature space where classes can be easily separated.

The dataset

The studied dataset consists in 5000 train images and 2000 in the test set. Raw images have three channels (RGB format) and can easily be visualized by normalizing all values to fit in $[0, 1]$, with a *sigmoid* function or using *min* and *max* functions. There are 10 possible labels per image. Labels in the train dataset are balanced which means that they are 500 images of each label. The low resolution of the images makes classification difficult even for the human eye. Thus, it seems necessary to extract features from each image in order to classify them more easily.

3 Local features and descriptors

Our first feature is the raw image. Indeed, an image in itself might contain interesting features. We experimented that sometimes adding the raw image to the constructed feature vector could enhance the results of our SVM. However, we designed feature extractors to ease SVM classification tasks.

Local Binary Pattern (LBP)

Local Binary Pattern (LBP) is a texture descriptor, that operates by comparing each pixel in an image to its neighbors, creating a binary code that captures local texture information. This process results in a feature vector derived from the histogram of these binary patterns, effectively summarizing the texture of the image. We apply this algorithm to the images converted in gray scale.

Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG) [Dalal and Triggs 2005] is a feature descriptor that quantifies occurrences of gradient direction within specific sections of an image. Primarily concentrating on an object's structure or shape, the HOG descriptor creates histograms for image regions based on the gradient's magnitude and direction. We apply it on several patches of the image in order to have local features. If HOG is the only feature used in our SVM we concatenate all these "local" vectors.

Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) [Lowe 2004] is an algorithm in computer vision to detect and describe local features in images. The algorithm was introduced by David Lowe in 1999 and has become a standard for feature detection and matching. We can either derive several vectors for one image (one by detected keypoint) or the concatenation of them all. The process of SIFT feature detection and description involves several key steps, which are:

1. **Scale-space Peak Selection:** This stage involves pinpointing prospective feature sites across various scales by constructing a scale-space pyramid and detecting local maxima and minima in the Difference of Gaussians (DoG).
2. **Keypoint Localization:** This involves enhancing the accuracy of keypoint positions and scales, while also discarding keypoints that have low contrast or resemble edges.
3. **Orientation Assignment:** Each keypoint is assigned one or more orientations based on the gradients of the local image, which guarantees the algorithm's invariance to rotation.
4. **Keypoint Descriptor:** This step involves encapsulating each keypoint within a high-dimensional vector, which is formulated from the orientations and magnitudes of local gradients.

4 Global features

After experimenting these local descriptors, we thought of implementing global features which gather previously extracted local descriptors in order to generate a feature vector for the entire image. We tried the two following algorithms with SIFT and HOG local descriptors. We also tried to concatenate to these global descriptors features calculated with algorithms described above.

Bag of visual words

The Bag of Visual Words (BoVW) model represents images using distinctive keypoints and their descriptors to form a visual vocabulary, akin to words in text. These keypoints are robust to image changes like rotation or scaling. By clustering descriptors from various images, 'visual words' are created, and each image is depicted as a histogram of these words.

Fisher vectors

The Fisher Vector [Sánchez et al. 2013] method for image classification enhances the Bag of Words model by incorporating statistical information about the distribution of local features in images,

*MVA, Applied Mathematics Department, ENS Paris-Saclay
forename.lastname@polytechnique.edu

[1] https://cvxopt.org — https://scikit-image.org

[2] https://github.com/lucasgascon/Kaggle-KM

typically using a Gaussian Mixture Model (GMM). The process involves extracting local descriptors from images, using these to train a Gaussian Mixture Model, and then computing Fisher Vectors by evaluating the deviation of each image's feature distribution from the Gaussian Mixture Model. These vectors are then normalized. Fisher Vector captures detailed texture and pattern information, making it effective for distinguishing subtle differences in images, albeit with higher computational demands. We experimented that with our implementation, this method suffers from a high variance and is computationally challenging due to the Expectation-Maximization algorithm to fit the GMM on our data.

5 Classification with SVMs

At the beginning of this challenge, we drew our inspiration from [Charpiat et al. 2015] where the authors present kernel methods to classify medical images. In the same way, we developed a pipeline which enables to first extract the chosen local and global features, concatenate them, apply a kernel Principal Components Analysis in order to reduce dimension of the feature space and finally perform a classification with a kernel SVM. To improve our training set, we performed data augmentation. Since most of the local features are rotation invariant, we used flipping as augmentation in order to also increase the number of different extracted features.

5.1 Kernels

When working with kernel methods in machine learning, such as Support Vector Machines (SVMs), the choice of the kernel function is crucial for the model's performance. The following kernels have been tested for both kernel PCA and kernel SVM. Here is a list of the various kernels we have implemented and used in our work:

1. **Linear Kernel**: The simplest kernel given by $K(x, y) = x \cdot y$. Useful for linearly separable data.
2. **Polynomial Kernel**: Models interactions between features up to a specified degree p, defined as $K(x, y) = (1 + x \cdot y)^p$. The degree p greatly influences the model's complexity.
3. **Gaussian (RBF) Kernel**: Popular for non-linear data, given by $K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$. The parameter σ controls the overfitting.
4. **Sigmoid Kernel**: Transforms the feature space similar to a neural network's sigmoid function, defined as $K(x, y) = \tanh(\gamma x \cdot y + r)$.
5. **Laplacian Kernel**: Similar to the Gaussian kernel but uses the L1-norm, defined as $K(x, y) = \exp\left(-\frac{\|x-y\|_1}{\sigma^2}\right)$. It can be more robust to outliers.
6. **Chi-Squared Kernel**: Especially useful in computer vision for histogram-based features, defined as $K(x, y) = \exp\left(-\sigma \sum \frac{(x_i-y_i)^2}{x_i+y_i+\epsilon}\right)$, with $\epsilon > 0$. This non-linear kernel captures complex patterns in the feature space.

Each kernel has its strengths and is suited to different types of data and problems. The choice of kernel and its parameters often depends on the specific dataset and requires empirical tuning. For each good-performing kernel, we performed a cross-validation in order to select the best hyperparameters which classify the images the best.

5.2 Strategies

Since kernel SVM is designed to perform binary classification, we had to develop a strategy to extend this algorithm to solve a 10 classes classification task. We learned more about different possible strategies through this PhD thesis [Passerini]. The first tested method is 'onevsall' which consists in training 10 kernel SVMs to classify between one label and the rest of the labels. Then, each trained model perform a classification on the test data and labels with most positive vote are then gathered and one is sampled randomly as final predicted label. However, several labels could often be equally picked and random picking occurs. So, we implemented a 'onevsone' strategy which consists in training one model per pair of labels in order to compare them all and then to select the most predicted label by a vote. In practice, this method is a bit longer even though the training data for each model is crucially reduced ($\leq 20\%$ of the full dataset) and better performs than the other strategy.

6 Results and conclusion

At first, with our own implementations of these algorithms we were able to mere exceed the baseline score in the ranking. After we have been authorized to use pre-implemented versions of the feature extractors, we tested a very large number of combinations of these features and hyperparameters. Through these experiments we noticed:

- In this case, reducing dimension of features with kernel PCA does not improve results.
- SIFT features are best classified with a linear kernel.
- HOG descriptors extract the most insightful features and are best classified with RBF kernel.

With Fisher vector based on HOG descriptors we were able to reach a good score on the leaderboard (0.48). However, our best score **0.592** has been reached by using HOG feature descriptors on raw and flipped images. The final submission file is an average (ensembling) between results from classification with a Laplacian kernel with $\sigma \in \{3.4, 3.5, 3.6, 3.7\}$ and C (the regularization parameter) set to 5. If several labels are equally predicted, there are chosen at random according to a probability related to the inverse frequency of the label occurrence for the already predicted images. Indeed, from our experiments and because of the train set distribution we assumed that the test set is also composed of 10% of each label. We finally reached 0.592 of accuracy on the public leaderboard

Finally, through this challenge, we realized how important it is to choose the kernel according to the data to be classified, or more precisely, the descriptors extracted from this data. Moreover, the choice of hyperparameters is also crucial in order to select the best possible kernel.

References

CHARPIAT, G., HOFMANN, M., AND SCHÖLKOPF, B. 2015. Kernel Methods in Medical Imaging. In *Handbook of Biomedical Imaging*, N. Paragios, J. Duncan, and N. Ayache, Eds. Springer US, Boston, MA, 63–81.

DALAL, N., AND TRIGGS, B. 2005. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, San Diego, CA, USA, vol. 1, 886–893.

LOWE, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*, 2 (Nov.), 91–110.

PASSERINI, A. Kernel Methods, Multiclass Classification and Applications to Computational Molecular Biology.

SÁNCHEZ, J., PERRONNIN, F., MENSINK, T., AND VERBEEK, J. 2013. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision 105*, 3 (Dec.), 222–245.