Kernel Methods for Machine Learning : Data Challenge Report

Franki NGUIMATSIA TIOFACK
franki.nguimatsiatiofack@ensae.fr
ENS Paris-Saclay (MVA 2023-2024)
Paris, France

Abstract

The purpose of this report is to present our work during the data challenge for the course titled "Kernel Methods for Machine Learning." The challenge focused on image classification. Our primary goal was to learn how to implement machine learning algorithms and adapt them to the structure of the data. We began by implementing and using the Kernel SVM algorithm and the Gaussian kernel to create two multi-class classification algorithms. However, due to the low accuracy obtained, we explored several methods for extracting relevant features and used several kernels. Using Grid-search and Cross-validation for hyperparameters tuning, we achieved our best score on the public kaggle test set by using features extracted by the SIFT algorithm and the Chi2 kernel (60.9%).

Keywords: Kernel, SVM, HOG, SIFT, Chi2 kernel, Gaussian kernel, Ensemble model, Image classification

1 Introduction

In this challenge, we worked with a set of 5000 color images, each with red, green, and blue channels, and dimensions of 32x32x3. These images are distributed across 10 classes: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. Each class contains 500 images, ensuring a balanced distribution. The provided labels correspond to these 5000 images. Additionally, an unlabeled test set of 2000 images was made available for evaluation.

The remainder of this report is structured as follows: we will first present our methodology step by step. Next, we'll discuss the performance of each implemented model. Finally, we'll briefly analyze errors made by our choosing model .

2 Methodology

2.1 Visualization

As recommended, we begin by visualizing some images to gain insights into their appearances. It becomes evident that these images have undergone preprocessing. Most of them appear blurred, and at this point, we are unable to determine the specific type of preprocessing applied. Furthermore, we cannot even affirm that all images have been transformed in the same manner.

2.2 Image Processing: features extraction

2.2.1 The Histogram of Oriented Gradients (HOG). HOG is a feature descriptor that focuses on capturing the structure and shape of objects within an image. Unlike other standard feature descriptors in computer vision, HOG computes features using both gradient magnitude and angle. It operates on localized cells, making it invariant to geometric and photometric transformations, except for object orientation. For more detailed information, you can refer to [1].

We modified the HOG algorithm from the skimage package. In the original version, the image is divided into 8x8 sub-blocks, which is better suited for images of dimensions 128x64. However, our case involves images of size 32x32. Since the skimage implementation did not yield good results (22.6% accuracy), we introduced a hyperparameter called div to control the sub-block division. Additionally, we removed the fourth step of the algorithm, which involves grouping 4 sub-blocks into a single block.

2.2.2 The Scale- Invariant Feature Transform (SIFT). The SIFT descriptor is robust and invariant to scale and rotation. It employs a keypoint detection algorithm and computes descriptors based on brightness gradient orientations. For further details, refer to [2].

Initially, we tried using the SIFT version in the skimage library, but it didn't work well, sometimes returning empty descriptor lists. To fix this, we used simplified a version of SIFT, which work as follow : instead of finding key points and their optimal scale, it created a grid over the image. Each grid point became a reference for a SIFT descriptor. It adjusted the local coordinate system around each point to ensure rotation invariance, then calculated gradient orientation and magnitude. Histograms are concatenated, standardized, and capped to reduce sensitivity to brightness changes, resulting in the final SIFT descriptor.

NB: We also tried using the kernel PCA algorithm from the second homework to reduce dimensionality and directly classify using these vectors, but due to extremely poor results (15% accuracy), we abandoned this approach.

2.3 Kernel SVM for binary classification

We initially attempted to adapt the Kernel SVM algorithm implemented in the second homework assignment for multi-class classification. However, it took too long to execute a model using the "one vs all" approach, with training still not complete after over 10 hours. With this implementation, obtaining results for the "one vs all" approach was impossible. Therefore, we also re-implemented the same algorithm, this time using the cvxopt library to solve the convex optimization problem associated with the algorithm, resulting in significantly faster execution.

2.4 Approaches for multi-class classification

We implemented the following approaches:

One vs. One: In this approach, for $K = 10$ classes, we train $\frac{K(K-1)}{2} = 45$ binary classifiers. Each classifier receives samples from a pair of classes from the initial training dataset and learns to distinguish between these two classes. During prediction, a voting scheme is applied: all 45 classifiers are used on an unknown sample, and the class that receives the highest number of "+1" predictions is predicted by the combined classifier.

One vs. All: This strategy involves training a single classifier per class, with samples from that class as positive examples and all other samples as negative examples. With this approach, the base classifiers produce a real score for their decision rather than a simple class label. To make decisions, all classifiers are applied to an unknown sample, and the label k for which the corresponding classifier reports the highest confidence score is predicted.

2.5 Training and Ensemble Method

We partitioned the set of 5000 images, along with their corresponding labels, into two distinct subsets: a training set (70%) and a local test set (30%). Utilizing the training set, we conducted a grid search combined with cross-validation to identify optimal values for the hyperparameters, primarily focusing on C for SVM, σ for the Gaussian Kernel, γ for the Chi2 Kernel and the div parameter, for each model we trained. Subsequently, we evaluated the generalization performance of our models using the local test set before employing them to predict labels for 2000 unlabeled images.

We implemented the Gaussian kernel, the Chi2 kernel, a kernel consisting of the sum of the two kernels (Gaussian and Chi2), and a kernel composed of the pointwise product of the two kernels. We utilized either SIFT or HOG features for both multi-class classification approaches described above, along with each of these kernels.

Furthermore, we developed an ensemble method in the following manner: applying the logic of the One vs All approach, we trained a single classifier per class using each of the kernel functions mentioned earlier. For each classifier, samples from the corresponding class were designated as positive examples, while all other samples were treated as negatives, alongside incorporating either HOG or SIFT features. This process resulted in a total of 60 classifiers. During prediction, the ensemble method generated predictions by using each stored classifier and summing their scores to determine the final class prediction for each input feature vector.

3 Experimental resultats overview

The results highlight the impact of feature extraction techniques and kernel functions on classification accuracy. For the One vs All approach (see Annex: table 1), using HOG features with the Gaussian kernel achieved an accuracy of 54.9% on the local test set and 55.2% on the public Kaggle test set. Conversely, the one using SIFT features with the Chi2 kernel led to an accuracy of 58.6% on the local test set and 59.4% on the public Kaggle test set. Furthermore, SIFT features with the Gaussian x Chi2 kernel achieved an accuracy of 60.4% on the local test set and 60.6% on the public Kaggle test set.

In the case of the One vs One approach (see Annex: Table 2), a model combining HOG features with the Gaussian kernel resulted in an accuracy of 54.3% on the local test set and 54.7% on the public Kaggle test set. Conversely, employing SIFT features with the Chi2 kernel yielded our highest accuracy, with 60.9% on the public Kaggle test set and 59.6% on the local test set . Finally, a model combining the Chi2 kernel and SIFT features achieved an accuracy of 58.6% on the local test set and 59.4% on the public Kaggle test set.

Regarding the ensemble combining Gaussian, Chi2, and Gaussian + Chi2 kernels with HOG and SIFT features, an accuracy of 59.7% on the local test set and 59.8% on the public Kaggle test set was achieved. While this demonstrates good generalization properties, it is noted that this accuracy is slightly lower compared to the model combining the Chi2 kernel and SIFT features. However, due to the time required to train this ensemble of models, we were unable to tune its hyperparameters, and we believe that by tuning these hyperparameters, better results could be achieved.

4 Error analysis

The selected model, which integrates SIFT features and Chi2 using the One vs One approach, exhibited a higher rate of incorrect predictions, exceeding 50%, particularly for the dog, deer, and bird classes on the local test set (refer to Annex: Figure 1). This maybe due to the visual similarity among the classes of dog and deer, complexity of their features, or a limitations of SIFT descriptors.

NB: By running the "START" notebook, you obtain the result of the model combining the Chi2 kernel and SIFT feature. However, the file contains the complete code required for all experiments in this work.

References

[1] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[2] E.N. Mortensen, Hongli Deng, and L. Shapiro. “A SIFT descriptor with global context”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 184–190 vol. 1. DOI: 10.1109/CVPR.2005.45.

Annex

Table 1. Results for the One vs All approach

Gaussian	Chi2	Gaussian + Chi2	Gaussian x Chi2	HOG	SIFT	Local accuracy	Kaggle accuracy
✓				✓		54.9%	55.2%
✓					✓	09.6%	11%
	✓			✓		59.7%	58.3%
	✓				✓	58.6%	59.4%
		✓			✓	58.3%	57.5%
			✓		✓	60.4% %	60.6%

Table 2. Results for the One vs One approach

Gaussian	Chi2	Gaussian + Chi2	Gaussian x Chi2	HOG	SIFT	Local accuracy	Kaggle accuracy
✓				✓		54.3%	54.7%
✓					✓	14.3%	10%
	✓			✓		55.2%	57.2%
	✓				✓	59.6%	60.9%
		✓			✓	59.8%	59.2%
			✓		✓	53.7%	53.1%

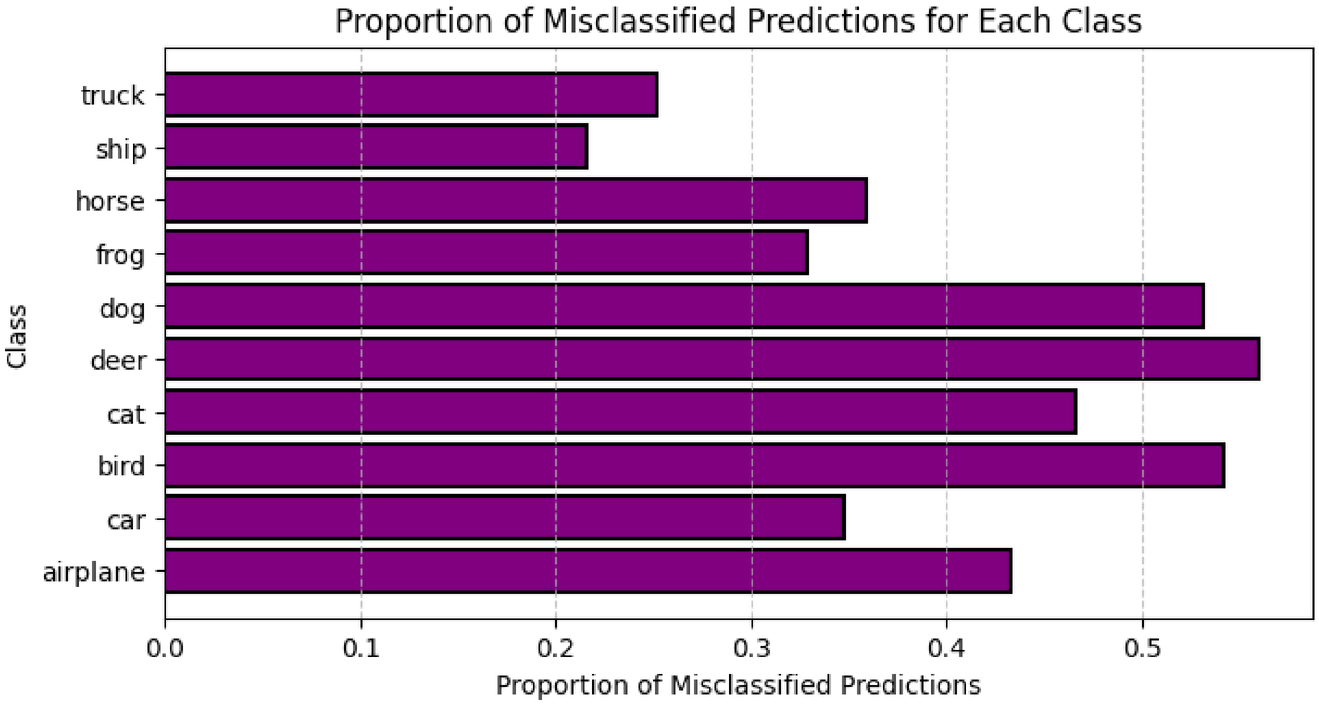

Figure 1. Proportion of Misclassified Predictions on local test set