HegazyVati Kaggle Challenge Team

Machine Learning with Kernel Methods

Inès Vati*
MVA, ENS Paris-Saclay, Cachan, France
ines.vati@eleves.enpc.fr

Mahmoud Hegazy†
MVA, ENS Paris-Saclay, Cachan, France
mahmoud.hegazy@ip-paris.fr

1 Introduction

This small report explains the approach employed by our team during the data challenge[3] as part of the MVA course "Machine Learning with Kernel Methods". The challenge is an image multi-classification task with pre-processed images. After implementing simple multi-class kernel SVMs, which performed poorly, our team opted to implement Convolution Kernel Networks (CKN) [2, 3]. The code is available at https://github.com/InesVATI/data-challenge-kernel-methods.

In this report, in Section 2, we first report the results achieved with simple classifiers. Subsequently, Section 3 provides a brief summary of the CKN architecture and our implementation details.

Methods	Accuracies
RBF SVM	0.147
Cosine Similarity KFD	0.146
CKN	0.625

Table 1: Validation accuracies of the main methods we investigated

2 Simple Classifier Baselines

We first explore kernel multi-class SVM with a Gaussian kernel. SVM applied on the flattened image did not perform well. Then, we used Kernel Fisher's Discriminant (KFD) as explained in [1, 4] using cosine similarity kernel. The accuracy obtained on validation data are reported in Table 1.

3 Convolution Kernel Network (CKN)

We then implemented convolutional kernel networks [2, 3] to derive a new image representation. Convolutional Kernel Network is an approximation of the multilayer convolutional kernel which is defined in Definition 3 in [3]. It consists of a sequence of spatial convolutions with learned filters, denoted as Z_k for a layer k, pointwise non-linearities, and pooling operations, as illustrated in Figure 1.

First, the projection step allows to represent local neighbourhoods in a RKHS $\mathcal{H}_k$ by projecting onto a finite-dimensional subspace of $\mathcal{H}_k$. For all pixel $p_{k-1} \in \Omega_{k-1}$[4], $M_k(p_{k-1}) = \psi(p_{k-1})$ is the representation in the corresponding RKHS of the patch of shape $e_k \times e_k$ centered in p_{k-1}. This

*Engineering student from École des Ponts ParisTech, Champs-sur-Marne, France

†PhD-track IP-Paris https://mahegz.github.io

[3] https://www.kaggle.com/competitions/data-challenge-kernel-methods-2023-2024-extension/overview

[4] Ω_{k-1} is the set of coordinate of the input image of layer k

step is encoded by **Idea 1** in [2]. Subsequently, in order to gain invariance to small shifts, a linear pooling described in **Idea 2** in [2] is applied to M_k to obtained $I_k \in \Omega_k$, the output of layer k.

In the unsupervised setting, the parameters Z_k are the centroids of a database built from random patches from training images. They are obtained by spherical K-means algorithm.

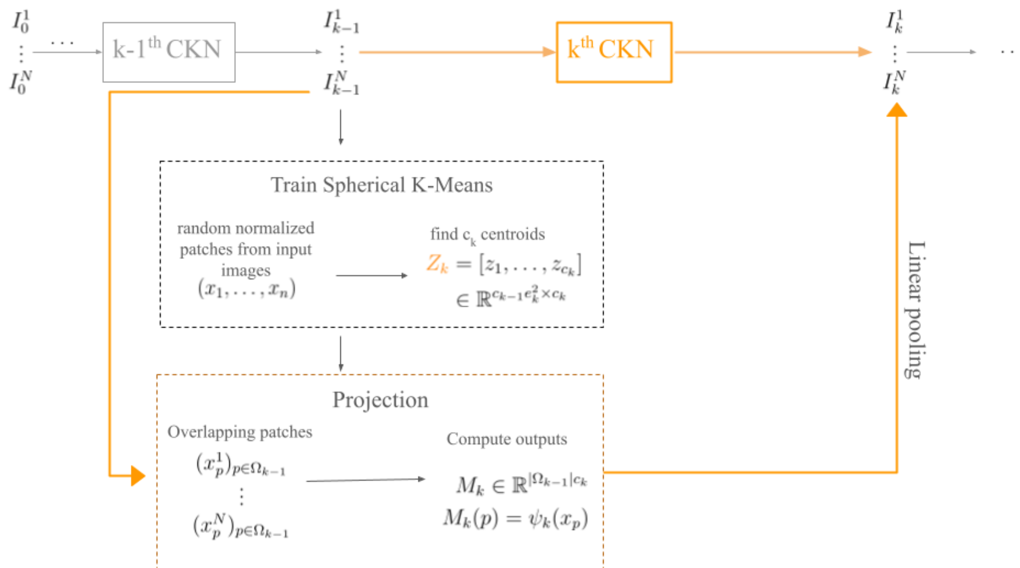

Figure 1: Illustration of the unsupervised training of a CKN model. Z_k are the learnt parameters of the kth layer. c_k is the output channel size of the layer, it corresponds to the number of centroids (z_i). $e_k \times e_k$ is the shape of the patch encoding the receptive field of the layer.

We used 3 layers with a patch size of 3 for the first layer and of 2 for the two last layers. The output channel size are respectively 64, 128 and 256. For computation time sake, we only trained the model on 2,000 random training images. In addition, we used a batch size of 500 images to prevent memory issues. It is enough to learn the model's parameters as the spherical K-means is trained on random patches so that the overall number of training data is sufficient.

The trained model is then applied on all 5,000 training images to obtain the corresponding features. The feature matrix is then z-scored and a Multi-class SVM is trained with a Gaussian kernel whose variance is 256. The parameter C is set to $C = 1$.

References

[1] Xijun Liang et al. *Kernel-based Algorithms for Image Classification: A Review*. Nov. 8, 2023. DOI: 10.21203/rs.3.rs-3576956/v1.

[2] Julien Mairal. *End-to-End Kernel Learning with Supervised Convolutional Kernel Networks*. Oct. 25, 2016. arXiv: 1605.06265[cs,stat]. URL: http://arxiv.org/abs/1605.06265 (visited on 02/26/2024).

[3] Julien Mairal et al. "Convolutional Kernel Networks". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/hash/81ca0262c82e712e50c580c032d99b60-Abstract.html (visited on 02/23/2024).

[4] S. Mika et al. "Fisher discriminant analysis with kernels". In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*. Neural Networks for Signal Processing IX: 1999 IEEE Signal Processing Society Workshop. Madison, WI, USA: IEEE, 1999, pp. 41–48. ISBN: 978-0-7803-5673-3. DOI: 10.1109/NNSP.1999.788121. URL: http://ieeexplore.ieee.org/document/788121/ (visited on 02/10/2024).