Revisiting Deep Fisher Vectors: Using Fisher Information to Improve Object Classification

Sarah Ahmed
ssarahahmedd@gmail.com
Tayyaba Azim
ta7g21@soton.ac.uk
Joseph Early
j.a.early@soton.ac.uk
Sarvapali D. Ramchurn
sdr1@soton.ac.uk

The UKRI Trustworthy Autonomous Systems (TAS) Hub,
University of Southampton,
Southampton, UK.

Abstract

Although deep learning models have become the gold standard in achieving outstanding results on a large variety of computer vision and machine learning tasks, the use of kernel methods has still not gone out of trend because of its potential to beat deep learning performances at a number of occasions. Given the potential of kernel techniques, prior works have also proposed the use of hybrid approaches combining deep learning with kernel learning to complement their respective strengths and weaknesses. This work develops this idea further by introducing an improved version of Fisher kernels derived from the deep Boltzmann machines (DBM). Our improved deep Fisher kernel (IDFK) utilises an approximation of the Fisher information matrix to derive improved Fisher vectors. We show IDFK can be utilised to retain a high degree of class separability, making it appropriate for classification and retrieval tasks. The efficacy of the proposed approach is evaluated on three benchmark data sets: MNIST, USPS and Alphanumeric, showing an improvement in classification performance over existing kernel approaches, and comparable performance to deep learning methods, but with much reduced computational costs. Using explainable AI methods, we also demonstrate why our IDFK leads to better classification performance.

1 Introduction

Over the last two decades, advances in machine learning have mostly utilised deep models with improvements in the learning algorithms and architectures to beat the state of the art performance in computer vision and machine learning. The supremacy of deep models was first challenged by Jaakola et al. [23] who proposed the Fisher kernel to encode higher order statistics from data that continued to work very well for large scale learning problems [8, 52, 53, 54, 55, 56]. Recently, the generalization performance of over parameterized deep learning models became a subject of intense study which lead to building a new connection

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

between ANN training and kernel methods for learning by analyzing deep architectures of infinite network [13, 24, 30]. Both the paradigms have their unique advantages and limitations, leading to the development of hybrid approaches that combine the benefits of deep models with kernel methods [5, 7, 20, 34, 46, 47]. Our work focuses on enhancing a particular hybrid approach that utilises Fisher kernels derived from the deep Boltzmann machines (DBMs) [5] to improve the discrimination power of the Fisher score space in its compact form for kernel extraction. Our approach shows that the discrimination power of such Fisher vectors could be enhanced by using an approximation of Fisher information matrix for image classification and retrieval tasks for limited budget applications, where incorporating ultra-deep models such as GANs could be challenging even in their pretrained form due to their high memory footprint. Our approach is summarised in Figure 1, and the contributions of this paper are as follows:

1. We demonstrate novel theoretical support for deriving an improved Fisher kernel from a compact DBM using the Fisher information matrix (FIM). To the best of our knowledge, the Fisher information matrix (FIM) has not been deployed with this genre of Fisher kernels before.
2. We empirically show that using an approximated FIM improves the discrimination power of deep Fisher score space on three benchmark data sets: MNIST, USPS and Alphanumeric.
3. We interpret the model trained on our improved deep Fisher features using global SHAP values [31], and also discuss the faster convergence rates and improved computational costs of our approach.

The remainder of this paper is organised as follows: Section 2 discusses the related work, Section 3 describes our proposed approach. Section 4 details our experiments and results, Section 5 discusses our research findings, and Section 6 concludes the paper with pointers to our future research plan.

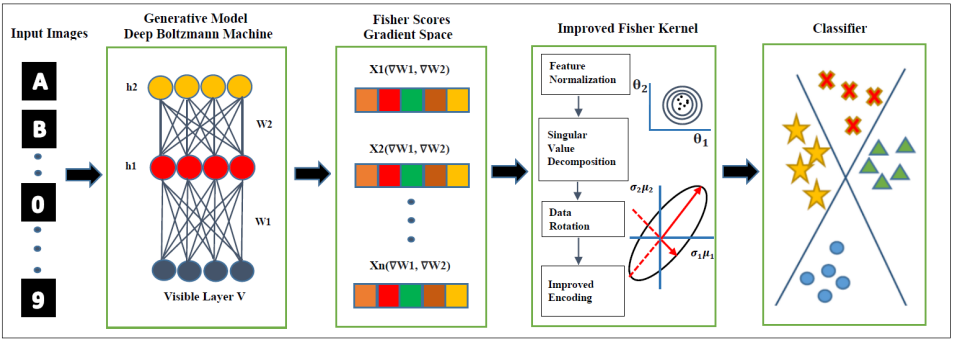

Figure 1: An overview of our proposed framework that bridges the gap between the two popular paradigms of kernel learning and deep learning methods for object classification.

2 Background and Related Work

Kernel methods are an attractive solution to image classification and retrieval due to their strong mathematical foundation and ability to solve complex tasks. Fisher kernels, originally proposed by Jaakkola and Haussler [22], have shown to be effective in a variety of

applications related to speech, text, and images. The performance gain of the technique on object classification task was first highlighted by Holub et al. [19] by combining the probabilistic constellation model with Fisher kernels. Following them, Perronin and Dance [33] applied the Fisher kernel framework to a visual vocabulary of bag of words features modelled via Gaussian mixture model. Since then, the idea has been successfully applied for classification on many large scale object recognition data sets such as CalTech-256, PASCAL VOC 2007, PASCAL VOC 2008 and ImageNet LSVRC 2012 using Gaussian mixture models [10, 35, 41, 43]. Although Fisher vectors (also known as Fisher scores) derived from Fisher kernels are suitable for image categorisation, they are known to be dense and high dimensional, leading to large image signatures and high memory costs. In image retrieval, this challenge is overcome by using compression and binarisation techniques applied prior to image search to speed up the retrieval process without significant loss of performance [1, 2, 42]. Furthermore, the Fisher kernel approaches have been largely overshadowed with emerging deep neural models with higher depth, consistently outperforming the existing kernel methods. As such, prior work has drawn parallels between deep learning and kernel learning leading to the development of hybrid approaches [4, 5, 12, 16, 20, 34, 46], which utilise the best parts of both paradigms. This work also focuses on one such hybrid approach that combines the mathematical rigor of kernel learning method with the structural richness of deep Boltzmann machine using Fisher kernels whose discrimination power has been improved through an approximation approach discussed in the next section. Recently, a hybrid approach close to our research was proposed by Zhang et al. [57], who discussed the possibility of drawing Fisher kernel from neural networks, in specific LeNet, multi-layer perceptron (MLP), generative adversarial network (GAN) and variational auto-encoders (VAEs) by deploying singular value decomposition for low rank approximation of Fisher vectors based on power iterations methods. Our research, in comparison, deploys a different deep learning model, i.e. DBM, in its compact form for kernel extraction. The difference just not resides in the use of a different deep learning architecture but also in its scale (i.e. a 2 layered compact model with few hidden units), thus showing the potential of the proposed approach for limited budget applications, where incorporating ultra-deep models such as GANs could be challenging even in their pre-trained form due to their high memory footprint. Moreover, Zhang et al. [57] have used a linear classifier on top of the feature embeddings, whereas we have used SVM and k-NN as a classifier on the top of the improved Fisher score embeddings.

3 Methodology

This section explains our proposed approach of computing an empirical approximation of Fisher information matrix, embedded into a deep Fisher kernel derived from a very compact deep Boltzmann machine (DBM). To the best of our knowledge, the Fisher information matrix has not been deployed with this genre of Fisher kernels before. We show that our approach can achieve near to the state of the art performance while only using a very compact neural deep model for drawing a gradient manifold capable of giving discriminative feature space.

3.1 Deep Boltzmann Machines

Deep Boltzmann machines (DBMs) [40] are generative models of symmetrically connected binary stochastic units. They consists of a set of visible units $\mathbf{v} \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{D}}$ for observing data, and a sequence of layers of hidden units $\mathbf{h^1} \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{L_1}}$, $\mathbf{h^2} \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{L_2}} \ldots, \mathbf{h^n} \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{L_n}}$ that

detect interesting features in the observations fed to the visible layer during training. There are connections only between the hidden units in adjacent layers, as well as between the visible units and the hidden units in the first hidden layer. In this work, we use a DBM with two hidden layers, i.e. L = 2. In this case, the energy of the state $\{\mathbf{v},\mathbf{h}\}$ is defined as:

$$\mathrm{E}(\mathbf{v},\mathbf{h};\theta) = \mathbf{v^T W^1 h^1} + \mathbf{h^1 W^2 h^2}, \tag{1}$$

where $\mathbf{h} = \{\mathbf{h^1},\mathbf{h^2}\}$ is the set of hidden units in each respective layer, and θ= $\{\mathbf{W^1},\mathbf{W^2}\}$ are the model parameters, representing visible-to-hidden and hidden-to-hidden symmetric connections. The probability that the model assigns to a visible vector $\mathbf{v}$ is given as:

$$p(\mathbf{v};\theta) = \frac{p^*(\mathbf{v};\theta)}{Z(\theta)} = \frac{1}{Z(\theta)}\sum_{\mathbf{h}}\exp(-\mathrm{E}(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2;\theta)), \tag{2}$$

where p^* denotes the un-normalized probability and $Z(\theta)$ is the partition function. To utilise this model with Fisher kernel, we must first compute the Fisher vectors $\phi_{\mathbf{x}}$ (also known as Fisher scores) by computing the gradients of the log likelihood of this generative model with respect to its parameters $\theta = \{\mathbf{W}^1,\mathbf{W}^2\}$:

$$\begin{aligned}
\phi_{\mathbf{v}} = \nabla_\theta \log p(\mathbf{v}_n|\theta) &= \left[\mathbf{S_{[n]}}\middle|\mathbf{Q_{[n]}}\right], \text{where} \\
\mathbf{S_{[n]}} = \nabla_{\mathbf{W}^1} \log p(\mathbf{v}_n|\mathbf{W}^1) &= \langle \mathbf{v h^{1T}}\rangle_{P_{data}} - \langle \mathbf{v h^{1T}}\rangle_{P_{model}}, \\
\mathbf{Q_{[n]}} = \nabla_{\mathbf{W}^2} \log p(\mathbf{v}_n|\mathbf{W}^2) &= \langle \mathbf{h^1 h^{2T}}\rangle_{P_{data}} - \langle \mathbf{h^1 h^{2T}}\rangle_{P_{model}}.
\end{aligned} \tag{3}$$

The angle brackets, $\langle . \rangle$ in the above equations refer to the expected value over a certain distribution specified by the subscript it follows: P_{data} refers to the probability distribution $p(\mathbf{h}|\mathbf{v})$ which is tractable, whereas P_{model} refers to the probability distribution, $p(\mathbf{v},\mathbf{h})$, which cannot be calculated analytically when the model has a non-trivial number of hidden units. In practice, contrastive divergence is used to train such a system [17].

3.2 Fisher Kernel

The Fisher kernel [22] provides a generic framework for deriving a kernel from a generative probability model, $P(\mathbf{x}|\theta)$. The Fisher kernel function is defined as as:

$$K(\mathbf{x_i},\mathbf{x_j}) = \phi_{\mathbf{x_i}}^{\mathbf{T}} \mathbf{F}^{-1} \phi_{\mathbf{x_j}}, \tag{4}$$

where $\phi_{\mathbf{x}}$ are Fisher vectors (as derived above for DBMs), and $\mathbf{F}$ is the Fisher information matrix. It represents the covariance matrix of Fisher scores:

$$\mathbf{F} = \mathrm{E}_{p(\mathbf{v}|\theta)}[\phi_{\mathbf{v}}\phi_{\mathbf{v}}^T] = -E\left[\left(\frac{\partial^2 L(\theta|\mathbf{v})}{\partial\theta\partial\theta^T}\right)\right] = -E\left[H\left(\theta|\mathbf{v}\right)\right], \tag{5}$$

where L(θ;$\mathbf{v}$) defines the log-likelihood of the probability density function and H is the Hessian. The Fisher kernel uses the kernel trick to map the data points $\mathbf{x}$ to higher dimensional Fisher scores $\phi_{\mathbf{x}}$. We can then measure the similarity of examples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the Fisher score space. As such, the Fisher kernel can be utilised in discriminative classifiers such as logistic regression, decision trees, k-nearest neighbours, and support vector machines.

Unfortunately, utilising $\mathbf{F}$ exactly as given in Equation 4 is computationally infeasible due to the size of the Fisher vectors (and corresponding Fisher information matrix). Therefore, common approximations of $\mathbf{F}$ include: (1) using an identity matrix [25], (2) using diagonal empirical approximation that results in whitening of the signal (i.e. each dimension

will have zero-mean and unit-variance) [35], or (3) using an analytical approximation [48]. Note all these heuristics have been tested for Fisher kernels derived from Gaussian mixture models previously because of their significance for large scale object classification problem. In the next section, we show that the use of FIM is not immaterial for deep Fisher kernels derived from the DBMs and introducing it can bring benefits for the classification performance and convergence speed of the algorithm.

3.3 Improved Deep Fisher Kernel

Generally, a simple Fisher kernel without Fisher information matrix provides a good substitute empirically with reduced computational costs [1, 6, 7, 22]. While computing $\mathbf{F}$ is expensive, in this work we only use an approximation of it in our proposed kernel to significantly reduce the computational costs of downstream tasks such as classification (as opposed to using complete $\mathbf{F}$ in the kernel). We propose singular value decomposition (SVD) as a means of approximating the Fisher information matrix. This is only achievable after computing $\mathbf{F}$ exactly, but it allows us to create a more informative approximation of $\mathbf{F}$ leading to semantically rich Fisher vectors $\phi_{\mathbf{x}}$. The use of our approach also makes our derived Fisher vectors invariant to re-parametrisation of the probabilistic model thus improving the downstream classification performance (as we show in Section 4.3).

In our approach, the improved deep Fisher kernel (IDFK), takes the Fisher vectors derived from a DBM (Equation 3), and first normalises them using min-max and L2 normalisation schemes. We used min-max normalisation to prevent features with large numerical values from dominating other features in distance-based objective functions and then applied L2 normalisation to introduce non-sparsity in features in-order to leverage more rotation invariant features. The covariance matrix (Fisher information matrix $\mathbf{F}$) of these normalised Fisher scores $\phi_{\mathbf{x}}$ is then computed, and since it is symmetric and positive semi-definitive in nature, we factorise it using the SVD approach:

$$\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}} = \mathbf{u_1}\sigma_1\mathbf{v}_1^{\mathbf{T}} + \cdots + \mathbf{u_r}\sigma_r\mathbf{v}_r^{\mathbf{T}}. \tag{6}$$

Note that computing the SVD of $\mathbf{F}$ involves calculating the eigen values and eigen vectors of $\mathbf{F}\mathbf{F}^{\mathbf{T}}$ or $\mathbf{F}^{\mathbf{T}}\mathbf{F}$. The matrix $\mathbf{S}$ is a diagonal matrix with non-zero real valued entries which are singular and are arranged in descending order to denote their order of significance, while the matrix $\mathbf{U}$ is an orthogonal rotation matrix composed of eigen vectors of the covariance matrix which represent the dominant direction of the distribution. In order to derive our empirical approximation, we first rotate the features:

$$\mathbf{X}_{rot} = \mathbf{U}^{\mathbf{T}} \times \phi_{\mathbf{x}} \tag{7}$$

This reduces the multi-collinearity of the features, making them more suitable for downstream tasks such as classification. Next, diagonal elements of the $\mathbf{S}$ matrix which bound the parameter variances are used to compute improved deep Fisher vectors as the reciprocal of the rotated deep Fisher score and the square root of diagonal elements of the matrix $\mathbf{S}$:

$$\phi_{\mathbf{x}}^{improved} = \mathbf{X}_{rot} \times \mathbf{S}^{\frac{-\mathbf{1}}{\mathbf{2}}}. \tag{8}$$

Our improved deep Fisher kernel (IDFK) is thus defined as:

$$K(\mathbf{x_i}, \mathbf{x_j}) = (\phi_{\mathbf{x_i}}^{\text{improved}})^{\mathbf{T}}(\phi_{\mathbf{x_j}}^{\text{improved}}). \tag{9}$$

Note that an exact $\mathbf{F}$ is not used in Equation 9. Instead, we have used an empirical approximation of the Fisher information matrix for the diagonal terms only. In previous works that have used Fisher kernels derived from a DBM, $\mathbf{F}$ has been dropped altogether [4, 5]. This is the first work that shows the utility of introducing Fisher information matrix in deep Fisher kernel derived from DBM to induce more discrimination for the classification task.
We would here emphasize that using the IDFK on its own is insufficient — the pre-processing steps outlined above (normalisation of the vectors) must be followed in order to get competitive results with the linear classifier in SVM (see Section 4.3). Re-scaling of Fisher vectors also improves gradient based learning and optimisation as shown previously in the literature and Section 5.2 below.

4 Empirical Evaluation

To establish the efficacy of our approach, we performed several experiments and compared it to the existing approaches using Accuracy metric. In this section, we detail the data sets (Section 4.1), models (Section 4.2), and results (Section 4.3) of our experiments.

4.1 Data Sets

We have used three benchmark character recognition data sets: MNIST [29] USPS [21], and Alphanumeric[1] to evaluate the performance of the proposed approach. MNIST contains 28×28 gray scale handwritten digits ranging from 0 to 9. The data set has 60,000 training and 10,000 test images. USPS contains 16×16 hand written digits from 0 to 9. Although it is smaller than MNIST (7291 training examples and 2007 test examples), it is considered quite challenging (a human error rate of 2.5%), and the test set is more difficult than the training set. Alphanumeric consists of 20×16 binary images characters 0-9 and A-Z, with a total of 1404 samples (39 images per character). As no definitive train/test split is given for this data set, we used stratified k-fold cross validation (k=40) to generate our splits. In all cases, the images in these data sets are binarised and flattened before being passed to the visible layer of the DBM. This ensures a one-dimensional binary input as required.

4.2 Models and Classifiers

We examine the performance of three hybrid models utilising SVMs: Linear SVM with Improved DFK from DBM (our method), Linear SVM with DFK from DBM [5], and Linear SVM with FK from RBM [6]. We also train three further models using the same kernel approaches but replace the SVM with a k-nearest neighbour (k-NN) classifier, where k=1. When using these hybrid models, we first train the respective DBM or RBM model, and then use same model to generate the Fisher vectors. The architectures for the DBM and RBM models are given in Table 1. We also compare our approach to several further approaches: Deep Belief Networks (DBNs) [18], ClassRBM [28], and a suite of deep learning methods [11, 15, 26, 27, 36, 49, 58].

4.3 Results

Table 2 gives the comparison of our technique against other hybrid methods. Across all three data sets, and for both the SVM and k-NN approaches, Fisher vectors derived from IDFK give better classification results than using DFK and FK. As the generative models used are consistent between the models, the improvement in performance is due solely to the use of

[1] https://cs.nyu.edu/~roweis/data.html

Table 1: Architectures for the DBM and RBM models used in the hybrid approaches. Note that IDFK with DBM and DFK with DBM use the same model (both in terms of architecture and learned weights,i.e:$|\phi_{\mathbf{v}}|$=$|(\mathbf{v}\times\mathbf{h^1})+(\mathbf{h^1}\times\mathbf{h^2})|$, however embedding FIM improves the discrimination power of deep Fisher scores.

Method	**MNIST**	**USPS**	**Alphanumeric**
IDFK with DBM (Ours) [Hidden Units per Layer, $\|\phi_{\mathbf{v}}^{improved}\|$]	[(20, 40), 16480]	[(40, 80), 13440]	[(40, 80), 16000]
DFK with DBM [6] [Hidden Units per Layer, $\|\phi_{\mathbf{v}}\|$]	[(20, 40), 16480]	[(40, 80), 13440]	[(40, 80), 16000]
RBM with FK [6] [Hidden Units, $\|\phi_{\mathbf{v}}\|$ =$\|(\mathbf{v}\times\mathbf{h})\|$]	[10, 7840]	[10, 2560]	[10, 3200]

IDFK. We investigate why IDFK leads to better performance in Section 5.1, and provide area under the receiver operator characteristic curve results in the Supplementary Material.

Table 2: Performance comparison of our improved deep Fisher kernel with DBM using linear SVM and k-NN classifiers. The average accuracy is reported with standard deviation over multiple repetitions.

Method	**MNIST(%)**	**USPS(%)**	**Alphanumeric(%)**
Linear SVM IDFK with DBM (Ours)	$\mathbf{99.20\pm0.10}$	$\mathbf{99.10\pm0.02}$	$\mathbf{79.00\pm0.05}$
Linear SVM DFK with DBM [6]	98.20 ± 0.10	94.86 ± 0.02	70.56 ± 0.01
Linear SVM FK with RBM [6]	91.20 ± 0.10	86.92 ± 0.28	70.50 ± 2.73
k-NN IDFK with DBM (Ours)	$\mathbf{96.47\pm0.01}$	$\mathbf{97.71\pm0.01}$	$\mathbf{77.6\pm0.07}$
k-NN DFK with DBM [6]	85.61 ± 0.01	91.78 ± 0.01	71.52 ± 0.09
k-NN FK with RBM [6]	90.95 ± 0.03	78.02 ± 1.67	59.11 ± 3.84

Furthermore, using the accuracy metric, we have compared the performance of our proposed approach with existing deep learning approaches . Table 3 shows that our approach outperforms or is at least comparable to these methods. However, as we discuss in Section 5.3, our method has a smaller computational cost than these existing deep learning approaches.

Table 3: Performance comparison of our proposed approach (i.e. improved deep Fisher kernel with DBM) with other popular deep learning frameworks using the accuracy metric.

Method	**MNIST(%)**	**USPS(%)**	**Alphanumeric(%)**
Linear SVM IDFK DBM (Ours)	99.20 ± 0.10	99.10 ± 0.02	79.00 ± 0.05
Deep Belief Network [28]	98.75	93.10 ± 0.42	87.57 ± 1.60
ClassRBM [28]	94.51 ± 0.03	91.51 ± 0.72	71.52 ± 0.81
AlexNet [29]	96.60	97.30	**94.00**
CNN [28]	99.18	98.20	73.00
CKELM [28]	96.80	96.20	—
CNN-SVM Hybrid [29]	98.00	99.20	—
Resnet50 + nLDA [28]	99.10	99.20	—
Q-learning Deep Belief Network [28]	**99.50**	98.60	—
CNN with Two-State Q-Learning [28]	99.00	**99.70**	—

5 Discussion

In this section, we further discuss our findings, and investigate why our proposed IDFK outperforms existing hybrid approaches. We begin by examining the feature explainability

of the derived Fisher vectors (Section 5.1), then look into how IDFK affects convergence of downstream SVM training (Section 5.2), and finally discuss the computational cost of our approach (Section 5.3).

5.1 Feature Explainability of Fisher Vectors

In Section 4.3, we demonstrated that the IDFK proposed in this work significantly improves performance in comparison to DFK and FK. As the other parts of the methods (generative model and classifier) are kept consistent, the improvement in performance is due solely to the use of IDFK. In order to understand how the IDFK leads to better performance, we utilised t-distributed stochastic neighbor embedding (t-SNE) to project the high-dimensional Fisher vectors into 2D space. This allows us to compare the discriminant features of the Fisher vectors produced by IDFK with those from DFK. Figure 2 shows that IDFK leads to Fisher vectors with clearer class separation. This improved separability leads to easier downstream classification, e.g. using SVMs or k-NN classifier as in this work.

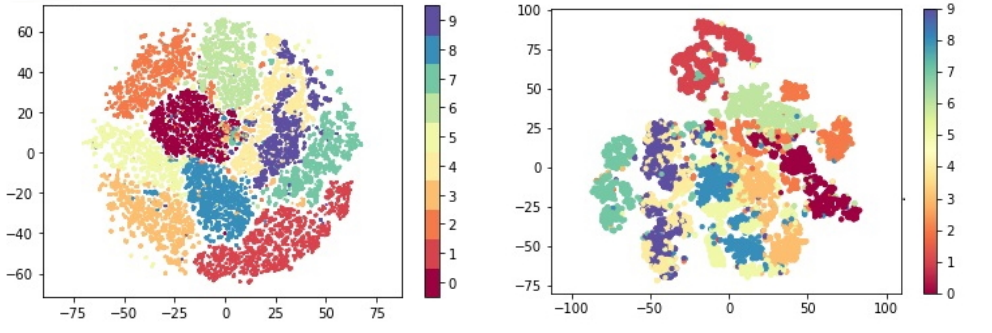

Figure 2: Comparison of the derived Fisher Vectors from IDFK (left) and DFK (right) for the MNIST dataset. We provide similar plots for the USPS and Alphanumeric datasets in the Supplementary Material.

While Figure 2 shows the separability of the MNIST classes using IDFK, it does not provide an analysis of the classifier that is applied to the derived Fisher vectors. To further investigate why IDFK leads to better performance, we extend our analysis to include the classifier as well. To explore how the derived features are used in classification, we can utilise SHapley Additive exPlanations (SHAP; [31]) to produce a global explanation of how the derived Fisher features relate to the class predictions. In Figure 3, we give an overview of the global feature importance for IDFK kNN on MNIST derived using SHAP. For each class, we identify the most supporting (largest positive SHAP value) and most refuting (largest negative SHAP value) features. We observe that there are Fisher vector features that provide strong support and strong rejection for each class. This further reinforces that improved separability is achieved, as each class is strongly supported as well as refuted.

5.2 Impact of Fisher Information Matrix on SGD Convergence

In this section, we analyse the performance of stochastic gradient descent (SGD) learning algorithm for estimating the parameters of linear SVM on improved deep Fisher vectors computed from the closed form approximation of Fisher information matrix. Numerous works

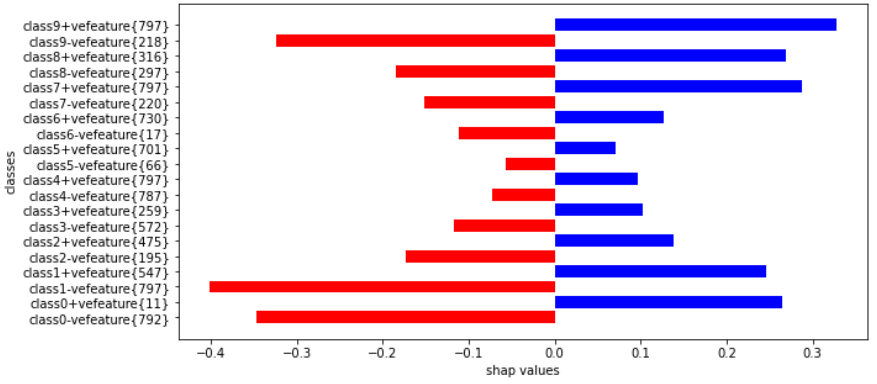

Figure 3: Top supporting and refuting features for each MNIST class for the IDFK with k-NN classifier. These were derived using global SHAP explanations [31]. We provide a similar plot for the USPS datasets in the Supplementary Material.

have advocated the optimisation of SGD and training deep learning models with stochastic gradient descent (SGD) approach, that not only estimates the parameters of the model to scale well on large data sets but also has a good convergence and generalization error at a minimal computational cost with respect to training time [37, 38, 45, 51]. However, in both theory and practice, they suffer from numerical instability and statistical inefficiency as estimators of the true parameter value. In order to tackle these issues, most works have focused on replacing the Hessian matrix with Fisher information matrix that is guaranteed to be positive definite and makes the SGD procedure more stable by affecting its learning speed [32, 39, 50]. It is observed that efficient computation of Fisher information matrix (FIM) helps to achieve the optimal Cramér-Rao bound under strong convexity and is considered an optimal unbiased estimator of the true parameter value for estimating the objective function shape.

Building a connection between generalisation error and stability of SGD procedure, we have also demonstrated the impact of embedding Fisher information matrix on SGD learning via performance curves on train data with respect to the number of epochs deployed by SGD learning algorithm. The convergence plots on the two data sets, MNIST and USPS are shown in Figure 4. These figures show that the proposed Fisher information matrix (embedded in IDFK) not only leads to higher accuracy, but also faster convergence. This implies the improved separability achieved by using IDFK allows the decision boundaries between classes to be learnt more easily, and also boosts generalisation (i.e., test set classification performance).

5.3 Computational Cost of Improved Deep Fisher Kernel (IDFK)

The time complexity of training DBM depends on the exact maximum likelihood learning of both the data dependent expectations $\langle . \rangle_{P_{data}}$ and the model's expectations $\langle . \rangle_{P_{model}}$, and is exponential in the number of hidden and visible units [40]. Since we have derived Fisher kernel from small DBM architectures, the training time is minimal, especially in comparison to larger deep learning models. The time complexity of computing the IDFK is dependent on two factors: finding the diagonal covariance matrix using SVD ($m \times n$ matrix takes $O(mn^2)$ [14]) and formation of kernel trick ($O(n)$ [44]). As discussed in Section 5.2, our approach also leads to faster convergence for SVM's SGD training, which further reduces the compu-

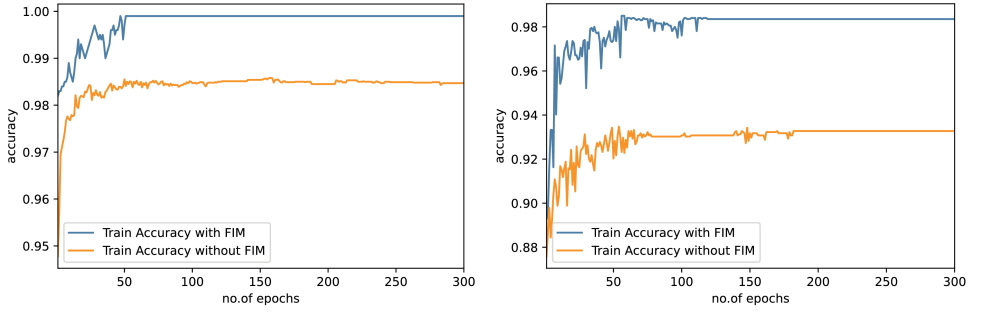

Figure 4: Convergence of SGD for training the SVM classifier on MNIST (left) and USPS (right) data sets . The train accuracy of the models with FIM (i.e. IDFK with DBM) and without FIM (i.e. DFK with DBM) is shown on the y-axis.

tational cost as fewer epochs are required to reach a high level of classification performance.

6 Conclusion

This work enhances the use of Fisher kernels drawn from the deep Boltzmann machine for visual object classification task. We show how our improved deep Fisher kernel (IDFK) could be derived by approximating the Fisher information matrix. The IDFK leads to improved performance when compared to the existing hybrid approaches [5, 6] on three benchmark data sets. Despite using a very compact deep model, the approach demonstrates a comparable performance to various deep learning approaches, but with a smaller computational footprint. Furthermore, through the use of explainable AI techniques, we show that the use of IDFK leads to better separability of the derived Fisher vectors, and also faster convergence during downstream training of SVM classifiers. In future, the approach could be further improved by embedding sparsity into the Fisher information matrix. This improvement would reduce the memory footprint of the proposed Fisher vectors enabling it to scale to larger object classification and retrieval tasks.

7 Acknowledgement

This work was funded by AXA Research Fund and the UKRI Trustworthy Autonomous Systems Hub (EP/V00784X/1). We would also like to thank the University of Southampton and the Alan Turing Institute for their support.

References

[1] Sarah Ahmed and Tayyaba Azim. Compression Techniques for Deep Fisher Vectors. In *ICPRAM*, pages 217–224, 2017.

[2] Sarah Ahmed and Tayyaba Azim. Condensing Deep Fisher Vectors: To Choose or to Compress? In *International Conference on Pattern Recognition Applications and Methods*, pages 80–98. Springer, 2017.

[3] Sarah Ahmed and Tayyaba Azim. Diversified Fisher Kernel: Encoding Discrimination in Fisher Features to Compete Deep Neural Models for Visual Classification Task. *IET Computer Vision*, 14(8):658–664, 2020.

[4] Mert Al, Thee Chanyaswad, and Sun-Yuan Kung. Multi-Kernel, Deep Neural Network and Hybrid Models for Privacy Preserving Machine Learning. IEEE Press, 2018. doi: 10.1109/ICASSP.2018.8462336. URL https://doi.org/10.1109/ICASSP.2018.8462336.

[5] T Azim. Fisher Kernels Match Deep Models. *Electronics Letters*, 53(6):397–399, 2017.

[6] Tayyaba Azim. *Visual Scene Recognition with Biologically Relevant Generative Models*. PhD thesis, University of Southampton, 2014.

[7] Tayyaba Azim and Mahesan Niranjan. Inducing Discrimination in Biologically Inspired Models of Visual Scene Recognition. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2013.

[8] Yunyin Cao, Jian Zhang, and Jun Yu. Image Retrieval via Gated Multiscale NetVLAD for Social Media Applications. *IEEE MultiMedia*, 27(4):69–78, 2020.

[9] Lin Chen and Sheng Xu. Deep Neural Tangent Kernel and Laplace Kernel Have the Same RKHS. *arXiv preprint arXiv:2009.10683*, 2020.

[10] Gabriela Csurka and Florent Perronnin. Fisher Vectors: Beyond Bag-of-Visual-Words Image Representations. In *International Conference on Computer Vision, Imaging and Computer Graphics*, pages 28–42. Springer, 2010.

[11] Shifei Ding, Lili Guo, and Yanlu Hou. Extreme Learning Machine with Kernel Model Based on Deep Learning. *Neural Computing and Applications*, 28(8):1975–1984, 2017.

[12] Amnon Geifman, Abhay Yadav, Yoni Kasten, Meirav Galun, David Jacobs, and Basri Ronen. On the Similarity Between the Laplace and Neural Tangent Kernels. *Advances in Neural Information Processing Systems*, 33:1451–1461, 2020.

[13] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do Neural Networks Outperform Kernel Methods? *Advances in Neural Information Processing Systems*, 33:14820–14830, 2020.

[14] Gene Golub and Charles F Van Loan. Matrix Computations. xxi, 2013.

[15] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. Reinforcement learning Based Handwritten Digit Recognition with Two-State Q-Learning. *arXiv preprint arXiv:2007.01193*, 2020.

[16] Tamir Hazan and Tommi Jaakkola. Steps Toward Deep Kernel Methods from Infinite Neural Networks. *arXiv preprint arXiv:1508.05133*, 2015.

[17] Geoffrey Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[18] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006.

[19] Alex Holub, Max Welling, and Pietro Perona. Combining Generative Models and Fisher Kernels for Object Recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 136–143. IEEE, 2005.

[20] Po-Sen Huang, Haim Avron, Tara Sainath, Vikas Sindhwani, and Bhuvana Ramabhadran. Kernel Methods Match Deep Neural Networks on TIMIT. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 205–209. IEEE, 2014.

[21] Jonathan J. Hull. A Database for Handwritten Text Recognition Research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[22] Tommi Jaakkola and David Haussler. Exploiting Generative Models in Discriminative Classifiers. In *Advances in Neural Information Processing Systems*, pages 487–493, 1999.

[23] Tommi S Jaakkola, Mark Diekhans, David Haussler, et al. Using the Fisher Kernel Method to Detect Remote Protein Homologies. In *ISMB*, volume 99, pages 149–158, 1999.

[24] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *Advances in neural information processing systems*, 31, 2018.

[25] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2011.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[27] Hüseyin Kusetogullari, Amir Yavariabdi, Abbas Cheddad, Håkan Grahn, and Hall Johan. Ardis: A Swedish Historical Handwritten Digit Dataset. *Neural Computing & Applications (Print)*, 32(21):16505–16518, 2020.

[28] Hugo Larochelle and Yoshua Bengio. Classification Using Discriminative Restricted Boltzmann Machines. In *Proceedings of the 25th International Conference on Machine Learning*, pages 536–543, 2008.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[30] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. *arXiv preprint arXiv:1711.00165*, 2017.

[31] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[32] James Martens. New Insights and Perspectives on the Natural Gradient Method. *arXiv preprint arXiv:1412.1193*, 2014.

[33] Florent Perronnin and Christopher Dance. Fisher Kernels on Visual vocabularies for Image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[34] Florent Perronnin and Diane Larlus. Fisher Vectors Meet Neural Networks: A Hybrid Classification Architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3743–3752, 2015.

[35] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision*, pages 143–156. Springer, 2010.

[36] Junfei Qiao, Gongming Wang, Wenjing Li, and Min Chen. An Adaptive Deep Q-learning Strategy for Handwritten Digit Recognition. *Neural Networks*, 107:61–71, 2018.

[37] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[38] Lorenzo Rosasco, Silvia Villa, and Bng Công Vũ. Convergence of Stochastic Proximal Gradient Algorithm. *Applied Mathematics & Optimization*, pages 1–27, 2019.

[39] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical Analysis of the Hessian of Over-parametrized Neural Networks. *arXiv preprint arXiv:1706.04454*, 2017.

[40] Ruslan Salakhutdinov and Geoffrey Hinton. Deep Boltzmann Machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.

[41] J. Sanchez and F. Perronnin. High-Dimensional Signature Compression for Large-Scale Image Classification. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, page 1665–1672, USA, 2011. IEEE Computer Society. ISBN 9781457703942. doi: 10.1109/CVPR.2011.5995504. URL https://doi.org/10.1109/CVPR.2011.5995504.

[42] Jorge Sanchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Compressed Fisher Vectors for Large-scale Image Classification. *Rapport de recherche RR-8209, INRIA*, 2013.

[43] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[44] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[45] Ohad Shamir and Tong Zhang. Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

[46] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Fisher Networks for Large-Scale Image Classification. In *Advances in Neural Information Processing Systems*, pages 163–171, 2013.

[47] Vladyslav Sydorov, Mayu Sakurada, and Christoph Lampert. Deep Fisher Kernels-end to end Learning of the Fisher Kernel GMM Parameters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1402–1409, 2014.

[48] Masayuki Tanaka, Akihiko Torii, and Masatoshi Okutomi. Fisher Vector Based on Full-covariance Gaussian Mixture Model. *Information and Media Technologies*, 8(4): 1041–1045, 2013.

[49] Yichuan Tang. Deep Learning Using Linear Support Vector Machines. *arXiv preprint arXiv:1306.0239*, 2013.

[50] Panos Toulis, Dustin Tran, and Edo Airoldi. Towards Stability and Optimality in Stochastic Gradient Descent. In *Artificial Intelligence and Statistics*, pages 1290–1298, 2016.

[51] Lin Xiao and Tong Zhang. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[52] Jun Yu, Yong Rui, and Bo Chen. Exploiting Click Constraints and Multi-view Features for Image Re-ranking. *IEEE Transactions on Multimedia*, 16(1):159–168, 2013.

[53] Jun Yu, Yong Rui, and Dacheng Tao. Click Prediction for Web Image Reranking Using Multimodal Sparse Coding. *IEEE Transactions on Image Processing*, 23(5): 2019–2032, 2014.

[54] Jun Yu, Dacheng Tao, Meng Wang, and Yong Rui. Learning to Rank Using User Clicks and Visual Features for Image Retrieval. *IEEE Transactions on Cybernetics*, 45 (4):767–779, 2014.

[55] Jun Yu, Min Tan, Hongyuan Zhang, Dacheng Tao, and Yong Rui. Hierarchical Deep Click Feature Prediction for Fine-grained Image Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[56] Jun Yu, Chaoyang Zhu, Jian Zhang, Qingming Huang, and Dacheng Tao. Spatial Pyramid-enhanced NetVLAD with Weighted Triplet loss for Place Recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2):661–674, 2019.

[57] Ruixiang Zhang, Shuangfei Zhai, Etai Littwin, and Josh Susskind. Learning Representation from Neural Fisher Kernel with Low-rank Approximation. *arXiv preprint arXiv:2202.01944*, 2022.

[58] Fa Zhu, Junbin Gao, Jian Yang, and Ning Ye. Neighborhood Linear Discriminant Analysis. *Pattern Recognition*, 123:108422, 2022.