MODELLING LOCAL DEEP CONVOLUTIONAL NEURAL NETWORK FEATURES TO IMPROVE FINE-GRAINED IMAGE CLASSIFICATION

ZongYuan Ge[†‡], *Christopher McCool*[‡], *Conrad Sanderson*[*◇], *Peter Corke*[†‡]

[†] Australian Centre for Robotic Vision, Brisbane, Australia
[‡] Queensland University of Technology (QUT), Brisbane, Australia
[*] University of Queensland, Brisbane, Australia
[◇] NICTA, Australia

ABSTRACT

We propose a local modelling approach using deep convolutional neural networks (CNNs) for fine-grained image classification. Recently, deep CNNs trained from large datasets have considerably improved the performance of object recognition. However, to date there has been limited work using these deep CNNs as local feature extractors. This partly stems from CNNs having internal representations which are high dimensional, thereby making such representations difficult to model using stochastic models. To overcome this issue, we propose to reduce the dimensionality of one of the internal fully connected layers, in conjunction with layer-restricted retraining to avoid retraining the entire network. The distribution of low-dimensional features obtained from the modified layer is then modelled using a Gaussian mixture model. Comparative experiments show that considerable performance improvements can be achieved on the challenging Fish and UEC FOOD-100 datasets.

Index Terms— fine-grained classification, deep convolutional neural networks, session variation modelling, Gaussian mixture models.

1. INTRODUCTION

Fine-grained image classification refers to the task of recognising the class or subcategory (for instance the particular fish species) under the same basic category such as bird or fish species [1, 17]. This is a challenging task for two reasons. First, some classes (species) from the same category, such as fish, can appear to be very similar in terms of appearance leading to low inter-class variation. Second, there is a high degree of variability in the instances of the same classes due to environmental and illumination variations leading to high intra-class variation. Fig. 1 shows examples of both issues.

An approach to tackling these two issues is to extract local region descriptors and to model them. Such an approach has previously been popular for recognition of faces [11, 16] and fish [1]. These approaches typically divide the image into patches (or blocks), with each patch considered to be an independent (and partial) observation of the object. Each patch is then represented by a feature vector and the distribution of all of these features vectors, from an image, is then modelled using a Gaussian mixture model (GMM). The feature vector to represent each patch has usually been obtained from a transform such as the 2D discrete cosine transform [16].

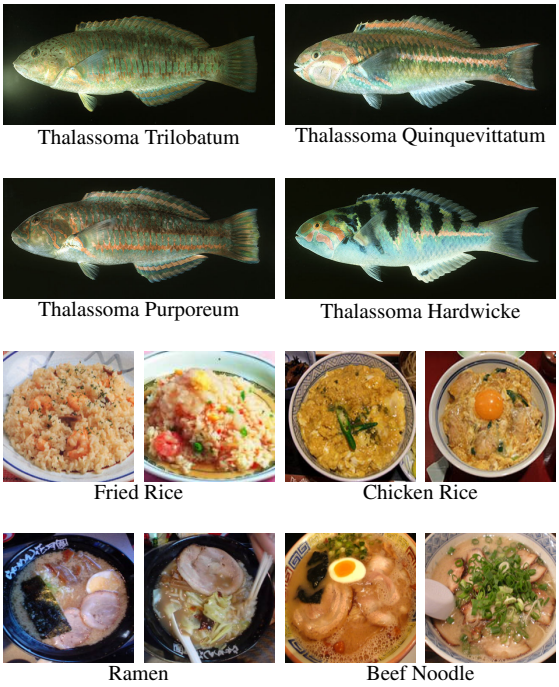

Fig. 1: First two rows show example images of four fish species, which have low inter-class variation: similar visual appearance despite being distinct species. (Images taken by J.E. Randall). The last two rows show images of four food dishes, with each dish type having high intra-class variation.

Recently, feature learning through the use of deep convolutional neural networks (CNNs) has led to considerable improvements for object recognition [10]. These deep CNN feature representations are trained on large datasets such as ImageNet [5] which has $1,000$ general object categories. It has been shown that these learnt features can be used to obtain impressive results for other recognition tasks when used as a global image representation [14]. However, to the best of our knowledge no work has examined how to use these learnt features as a local feature extractor for use with well known statistical modelling approaches such as GMMs.

To use these deep CNN features as a local feature extractor two issues need to be addressed. First, deep CNNs such

as [10] generally have an internal representation which is high dimensional, leading to the curse of dimensionality [3] for local modelling techniques such as GMMs. Second, we need to develop an efficient and effective method to retrain a deep CNN containing millions of weights using a relatively small set of images specific to a fine-grained class. In this paper we address both of these issues.

Inspired by recent work that has shown how to optimise deep CNN features for small datasets using fine-tuning [17], we propose a method to obtain a low-dimensional deep CNN representation that can be used as a local feature descriptor. Specifically, we propose to explicitly reduce the dimensionality of one of the internal fully connected layers, in conjunction with using layer-restricted retraining to avoid retraining the entire network. We demonstrate empirically that the proposed approach leads to considerable performance improvements for two fine-grained image classification tasks: fish recognition [1] and food recognition [12].

We continue the paper as follows. In Section 2 we briefly describe the image classification approach based on statistical modelling of local features and inter-session variability modelling. The approach is used as a base upon which we build on in Section 3, where we learn a low-dimensional deep CNN representation that can be used as local feature descriptor. Comparative experiments are given in Section 4, followed by the main findings and future directions in Section 5.

2. MODELLING LOCAL IMAGE FEATURES

Modelling the distribution of local features has been explored by several researchers [11, 16, 13]. In general, these methods divide the j-th image of the i-th class, $\boldsymbol{I}_{i,j}$, into N overlapping patches. Each patch is represented by an M-dimensional feature vector, of low dimensionality, to yield the set of N feature vectors $\boldsymbol{O}_{i,j} = [\boldsymbol{o}_{i,j,1}, \ldots, \boldsymbol{o}_{i,j,N}]$. The distribution of the vectors is then modelled using a GMM to obtain a prior model, referred to as a universal background model (UBM), that represents the basic category in question (eg. fish, food).

This UBM representation forms the basis which many feature modelling methods use. It can be used as a probabilistic bag-of-words representation [15] or a model can be derived for each class by performing mean-only relevance MAP adaptation [11]. Another extension is to perform inter-session variability (ISV) modelling [13] which learns those variations that can make one instance (image) of the same class look different to another image of the same class.

Irrespective of the specific method they all rely on a GMM which is known to perform poorly for high-dimensional data [4]. This is partly due to the curse of dimensionality where it becomes difficult to estimate a large number of parameters when there is limited data. To avoid this we will show how to learn a low-dimensional deep CNN representation, however, before proceeding to this we first describe the GMM feature modelling methods that we use in this work.

2.1. GMM Feature Modelling

We use two feature modelling approaches in this work, GMM mean-only MAP adaptation and its extension ISV. These two are chosen as they have been shown to provide consistently good performance [13].

GMM mean-only MAP adaptation takes the prior model (UBM) and adapts just the means using the enrollment data of the i-th class $\boldsymbol{O}_i$; all of the features for the J_i enrollment images. Using supervector notation [13], this is written as

$$\boldsymbol{s}_i = \boldsymbol{m} + \boldsymbol{D}\boldsymbol{z}_i, \tag{1}$$

where $\boldsymbol{s}_i$ is the mean supervector for the i-th class, $\boldsymbol{m}$ is the mean supervector of the UBM (the prior), $\boldsymbol{z}_i$ is a normally distributed latent variable, and $\boldsymbol{D}$ is a diagonal matrix that incorporates the relevance factor and the covariance matrix and ensures the result is equivalent to mean-only relevance MAP adaptation.

ISV is an extension of the GMM mean-only MAP model which learns a sub-space which models and suppresses session variation [13]. It includes a subspace $\boldsymbol{U}$ to cope with session variation and is written in supervector notation as

$$\boldsymbol{u}_{i,j} = \boldsymbol{m} + \boldsymbol{U}\mathbf{x}_{i,j} + \boldsymbol{D}\boldsymbol{z}_i, \tag{2}$$

where $\mathbf{x}_{i,j}$ is the latent session variable and is assumed to be normally distributed. Suppressing the session variation is done by jointly estimating the latent variables $\boldsymbol{z}_i$ and $[\mathbf{x}_{i,1}, \ldots \mathbf{x}_{i,J_i}]$ followed by discarding the latent session variables to give

$$\boldsymbol{s}_{ISV,i} = \boldsymbol{m} + \boldsymbol{D}\boldsymbol{z}_i, \tag{3}$$

For both of these methods, the log-likelihood ratio is used to determine if the t-th test image $\boldsymbol{I}_t$ was most likely produced by class i. This is efficiently calculated using the linear scoring approximation [7] which for GMM mean-only MAP is

$$h_{linear}\left(\boldsymbol{O}_t, \boldsymbol{s}_i\right) = \left(\boldsymbol{s}_i - \boldsymbol{m}\right)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{f}_{t|\boldsymbol{m}}, \tag{4}$$

and for ISV it is

$$h_{ISV}\left(\boldsymbol{O}_t, \boldsymbol{s}_i\right) = \left(\boldsymbol{s}_{ISV,i} - \boldsymbol{m}\right)^T \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{f}_{t|\boldsymbol{m}} - \boldsymbol{N}_t \boldsymbol{U} \mathbf{x}_{t|\boldsymbol{m}}\right),$$

where the diagonal matrix $\boldsymbol{\Sigma}$ is formed by concatenating the diagonals of the UBM covariance matrices, $\boldsymbol{f}_{t|\boldsymbol{m}}$ is the supervector of mean normalised first order statistics, and $\boldsymbol{N}_t$ contains the zeroth order statistics for the test sample in a block diagonal matrix [13].

3. PROPOSED METHOD

To extract features from local patches, we aim to learn a low-dimensional deep CNN representation which we refer to as a low-dimensional CNN feature vector (LDCNN). This is in contrast to the high dimensional representation (4096 dimensions) that is usually obtained from the fully connected layer (fc-6) of the pretrained deep CNN [10], the structure of this network can be seen in Fig. 2. Such high dimensional representations are difficult to be effectively modeled with a stochastic model such as a GMM, as such we aim to learn a

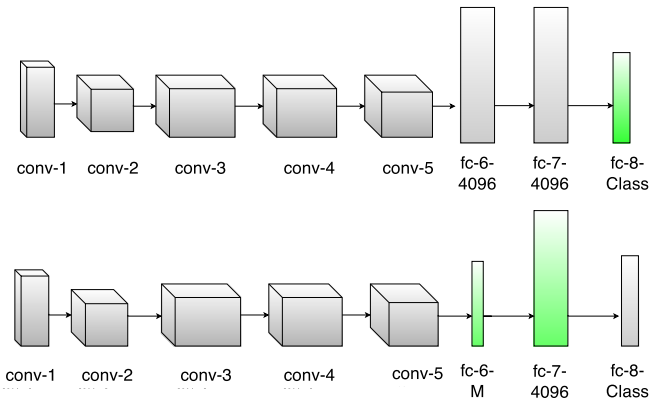

Fig. 2: Modifying and retraining the deep CNN through a 2 step procedure. For each step we have shaded in green the parts of the network that are changed and retrained. First step: the highlighted fc-8 layer is modified to have only as many outputs as the number of dataset specific classes. The layer is retrained, while all the other parameters remain fixed. Second step: the highlighted fc-6 layer is changed to map to only M outputs, followed by training the fc-6 layer in conjunction with the highlighted fc-7 layer, while keeping the remaining parameters fixed. The output of the fc-6 layer is used as a local feature extractor.

low-dimensional representation (LDCNN) whose dimensionality M is much less than 4096. To reduce the dimensionality while preventing the parameters from overfitting in the large CNN architecture, we propose a two step modification for the network.

In the first step, using the pretrained network of [10] as a starting point, we modify the final output layer (fc-8) to have outputs for the N_c training classes. The weights are randomly initialised[1] and retraining is then conducted such that only the fc-8 layer is updated using a learning rate of 0.01. This process equates to a multiclass linear regression, using the pretrained network as a feature extractor. It converges after a few thousand iterations.

In the second step we replace the two fully connected layers fc-6 and fc-7 and retrain only these two layers with the other layers fixed. We replace the original 4096 dimension fc-6 layer with a new M-dimensional fc-6 layer that is randomly initialised[1], where $M \ll 4096$. Features extracted from this layer are referred to as LDCNN. The fc-7 layer is also replaced and randomly initialised[1] as fc-6 and fc-7 are densely connected. However, when we retrain the network, fc-7 retains its original dimensionality of 4096. Retraining is then performed using back propagation and stochastic gradient descent to update only these two layers. The learning rate is initially set to 0.01 but this rate reduces by a factor of 10 for every 1000 iterations throughout training process. In this way, all pretrained convolutional layer filters from the original network [10] are retained.

[1] Random initialisation is performed by drawing from $\mathcal{N}\left(0, 0.01^2\right)$.

4. EXPERIMENTS

We evaluate our approach on two fine-grained image datasets: Fish [1] and UEC FOOD-100 [12]. For both datasets we present two baseline systems, both of which perform classification using an SVM and extract a single global CNN feature to represent each image. The first baseline extracts a single global feature vector using fc-6 of the pre-trained deep CNN [10] (4096 dimensions); we refer to this as **SVM-CNN**. The second baseline extracts a single global feature vector using the re-trained low-dimensional CNN feature (LDCNN) vector; we refer to this as **SVM-LDCNN**.

The local features modelling results (GMM), where the image is divided into N overlapping patches, use two feature extractors. These feature extractors obtain an M-dimensional feature vector from each of the N patches which is then modelled using a GMM. The first, **GMM-LDCNN**, uses the proposed low-dimensional CNN feature vector (LDCNN) to obtain the M-dimensional feature vector. The second, **GMM-PCA-CNN**, uses fc-6 pre-trained deep CNN [10] (4096 dimensions) and learns a transform using principal component analysis (PCA) [6] to reduce the dimensionality to M.

When we perform local feature modelling (GMM) a range of parameters are varied. The number of components evaluated for the GMM were $C = [128, 256, 512, 1024]$, the size of the ISV subspace was $N_U = [2, 4, 8, \ldots, 256]$, and the range of block sizes $B = [32, 64, 96, 128]$. For both datasets the images were resized to be 256×256. Caffe [8] was used to extract and retrain the CNN features and Bob [2] was used to learn the GMM and ISV models.

4.1. Fine-Grained Fish Classification

We use the Fish image dataset from [1] which consists of $3,960$ images collected from 468 species. This dataset contains images captured in different conditions, defined as "controlled", "out-of-the-water" and "in-situ". The "controlled" images consist of fish specimens with controlled background and illumination. The "in-situ" images are underwater images of fish in their natural habitat and the "out-of-the-water" images consist of fish specimens taken out of the water with a varying background.

Following the defined protocols, the dataset is split into three sets: a training set (*train*) to learn/derive UBM GMM models; a development set (*dev*) to determine the optimal parameters and decision threshold for our models and an evaluation set (*eval*) to measure the final system performance. There are two protocols: protocol 1a evaluates the system performance when high quality ("controlled") data is used to enrol classes and protocol 1b evaluates the system performance when low quality ("in-situ") data is used to enrol classes. For both protocols, the same test imagery (a mix of "controlled", "in-situ" and "out-of-the-water" images) is used. The local modelling approach used for these experiments was the ISV extension of the GMM approach as this provided a considerable boost for the initial experiments; we refer to this as **GMM-LDCNN**.

Table 1: Results on the Fish image dataset [1]. The two baseline approaches, SVM-CNN and SVM-LDCNN, are presented along with the state-of-the-art local modelling approach from [1] (Local GMM). GMM-PCA-CNN uses PCA reduced features from fc-6 of the pre-trained CNN [10]. The proposed GMM-LDCNN method uses LDCNN features in conjunction with GMMs. GMM-LDCNN-xy extends LDCNN features by adding the spatial location of each block.

System	Protocol 1a		Protocol 1b	
	Dev	Eval	Dev	Eval
SVM-CNN	40.9	45.8	41.9	45.7
SVM-LDCNN	39.2	44.2	40.3	43.5
Local GMM [1]	43.1	49.3	40.8	46.7
GMM-PCA-CNN	45.7	51.5	44.0	47.2
GMM-LDCNN	51.8	55.5	**46.4**	49.5
GMM-LDCNN-xy	**53.8**	**57.0**	46.2	**53.3**

It has been shown in [1] that incorporating spatial information can be advantageous, and as such we further propose to extend the GMM-LDCNN approach by adding the spatial location (x, y) to each local feature vector prior to modelling; we refer to this method as GMM-LDCNN-xy.

The results in Table 1 show that in contrast to global features, local modelling provides notable improvements: the two baseline systems (SVM-CNN and SVM-LDCNN) which use global features perform worse than the previous state-of-the-art local ISV modelling approach (Local GMM). Furthermore, our local low-dimensional GMM-LDCNN approach[2] outperforms local modelling of PCA-CNN features (GMM-PCA-CNN), with an average relative performance improvement of 6.4%. The extended form of the proposed approach (GMM-LDCNN-xy) provides further improvements and obtains state-of-the-art results, with an average relative performance improvement of 14.9% over Local GMM [1]. This demonstrates the effectiveness of local modelling over global features, and highlights the potential to use feature learning techniques such as CNNs to learn effective local representations.

4.2. Results on Food Dataset

We use the UEC FOOD-100 dataset which contains 100 Japanese food categories with more than 100 images for each category. Some images contain multiple classes and a bounding box is provided for each class. Examples are shown in Fig. 1. Features are extracted from the bounding box only, so detection/localisation is not considered in this paper.

We use half of the images from each class for training and the other half for testing[3]. The training images are used for retraining the CNN and to learn the UBM model. The dimensionality for fc-6 is set to $M = 256$ based on initial experiments. Initial experiments also indicated that the ISV extension to local modelling and including spatial (x, y) information in each feature vector did not provide performance improvements. As such, they were not used on this dataset. We believe that ISV did not lead to increased performance as this is a closed-set problem[4] with a high number of enrollment images, resulting in less effective learning of a representation for session variation independent of the class. The spatial information did not help as the images are not accurately registered, consequently modelling the location of parts (such as the eggs in Fig. 1) is not useful.

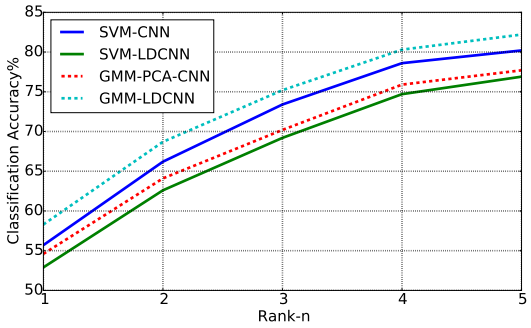

Fig. 3: Rank-n classification accuracy on the UEC FOOD-100 dataset [12].

The results, presented in Fig. 3, show that performing local modelling using the LDCNN features (GMM-LDCNN) provides the best performance[5]. The results in Fig. 3 are presented in terms of rank-n classification accuracy, where rank-n refers to if the class of interest is in the n best matches. In terms of rank-1 accuracy (identification accuracy), local modelling of the LDCNN features (GMM-LDCNN) has an accuracy of 58.3%, which provides a considerable relative performance improvement of 9.4% compared to the SVM-LDCNN approach (using LDCNN to extract a global feature) which has an accuracy of 52.9%. The GMM-LDCNN approach also outperforms the SVM-CNN approach which is similar to the best single feature system presented in [9] (referred to as DCNN in their work) and has a rank-1 accuracy of 55.7%.

5. CONCLUSION

In this paper we have explored the benefits of using deep convolutional neural networks (CNNs) to extract local features which are then modelled using a GMM. Our two-step retraining procedure provides an effective way to perform dimensionality reduction and provides considerably better performance than a simple linear model such as PCA. Comparative experiments show that considerable performance improvements can be achieved on the challenging Fish and UEC FOOD-100 datasets.

Future work will examine other ways to retrain the deep CNN. For instance, an issue not examined in this work is the possibility of extracting thousands of local patches from each image and using these samples to retrain the entire network.

[2]Optimal parameters for protocol 1a were $C = 1024$, $B = 128$, and $N_U = 128$, while for protocol 1b $C = 512$, $B = 96$, and $N_U = 128$.

[3]We developed these protocols as insufficient details were provided to reproduce the experiments in [9]; our protocol files will be publicly available.

[4]By closed set we mean that while the data differs between the training and testing sets, the classes in both sets are the same.

[5]The optimal parameters were $C = 512$ and $B = 32$.

6. ACKNOWLEDGMENTS

The Australian Centre for Robotic Vision is supported by the Australian Research Council via the Centre of Excellence program. NICTA is funded by the Australian Government through the Department of Communications, as well as the Australian Research Council through the ICT Centre of Excellence program.

7. REFERENCES

[1] K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. Fookes, P. Corke, D. Tjondronegoro, and S. Sridharan. Local inter-session variability modelling for object classification. *WACV*, 2014.

[2] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel. Bob: a free signal processing and machine learning toolbox for researchers. In *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*. ACM Press, Oct. 2012.

[3] C. Bishop. *Pattern Recognition and Machine Learning*, pages 33–38. Springer, 2006.

[4] C. Bouveyron, S. Girard, and C. Schmid. High dimensional data clustering. Technical report, LMC-IMAG, Université J. Fourier, Grenoble, 2006.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[6] K. Fukunaga. *Introduction to Statistical Pattern Recognition*, pages 399–417. Elsevier, second edition, 1990.

[7] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny. Comparison of scoring methods used in speaker recognition with joint factor analysis. In *ICASSP 2009*, pages 4057–4060.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[9] Y. Kawano and K. Yanai. Food image recognition with deep convolutional features. In *Proc. of ACM UbiComp Workshop on Cooking and Eating Activities (CEA)*, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[11] S. Lucey and T. Chen. A GMM parts based face representation for improved verification through relevance adaptation. In *CVPR 2004*, volume 2, pages 855–861.

[12] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, 2012.

[13] C. McCool, R. Wallace, M. McLaren, L. E. Shafey, and S. Marcel. Session variability modelling for face authentication. *IET Biometrics*, 2:117–129(12), September 2013.

[14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CVPR Workshop on Deep Vision*, 2014.

[15] C. Sanderson and B. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. *Lecture Notes in Computer Science (LNCS), Vol. 5558*, pages 199–208, 2009.

[16] C. Sanderson and K. K. Paliwal. Fast features for face authentication under illumination direction changes. *Pattern Recognition Letters*, 24(14):2409–2419, 2003.

[17] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, pages 834–849. 2014.